



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	Loose coupling in heterogeneous event-based systems via approximate semantic matching and dynamic enrichment
Author(s)	Hasan, Souleiman
Publication Date	2016-01-27
Item record	<a href="http://hdl.handle.net/10379/5511">http://hdl.handle.net/10379/5511</a>

Downloaded 2024-04-25T11:28:28Z

Some rights reserved. For more information, please see the item record link above.





NATIONAL UNIVERSITY OF IRELAND, GALWAY

DOCTORAL THESIS

---

**Loose Coupling in Heterogeneous  
Event-Based Systems via Approximate  
Semantic Matching and Dynamic  
Enrichment**

---

*Author*

Souleiman HASAN

*Supervisor*

Dr. Edward CURRY

*Examiners*

Prof. Gordon Blair

Dr. John Breslin

*A thesis submitted in fulfilment of the requirements  
for the degree of Doctor of Philosophy*

Insight Centre for Data Analytics  
College of Engineering and Informatics

January 2016



# Declaration of Authorship

I, Souleiman HASAN, declare that this thesis titled, ‘Loose Coupling in Heterogeneous Event-Based Systems via Approximate Semantic Matching and Dynamic Enrichment’ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

(Souleiman Hasan)

(January 2016)

*“I do not know what I may appear to the world, but to myself I seem to have been only like a boy playing on the seashore, and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me...”*

Isaac Newton

NATIONAL UNIVERSITY OF IRELAND, GALWAY

## *Abstract*

Insight Centre for Data Analytics  
College of Engineering and Informatics

Doctor of Philosophy

### **Loose Coupling in Heterogeneous Event-Based Systems via Approximate Semantic Matching and Dynamic Enrichment**

by Souleiman HASAN

There has been a significant change in the data landscape with the emergence of the Internet of Things (IoT). Tens of billions of devices are expected to connect to the Internet in the coming years within smart buildings, smart grids, smart cities, and cyber-physical systems. A basic requirement to realize the IoT is an infrastructure of sensing and communication solutions. Middleware systems, such as event processing, are also required to abstract the application developers from the underlying technologies.

Large-scale event processing environments are open, distributed, and heterogeneous in semantics and contexts. Interoperability is a key requirement and currently addressed by top-down granular agreements represented by ontologies and taxonomies for semantics. Such approaches are non-scalable, and achieving such agreements may be unfeasible under the characteristics of current and future event environments such as the IoT. This thesis analyses this problem using a decoupling versus coupling trade-off framework.

Event producers and consumers do not know each other and are decoupled in space, time, and synchronization to enable scalable deployments. They have boundaries that they have to cross in order to communicate with other systems. Such boundaries are syntactic, semantic, and pragmatic. Events are boundary objects that convey meanings signified by symbols. They must effectively cross the three levels of boundaries to establish interoperability and communication between event agents.

The current event processing paradigm is focused on crossing lower syntactic boundaries. Thus, human agents are needed in the loop to cross semantic and pragmatic boundaries through explicit agreements on event types, properties, values, and contexts, introducing coupling into these systems. Coupling limits the paradigm and contradicts the fundamental basis of decoupling for scalability. A trade-off can be concluded between decoupling for scalability and coupling for interoperability.

Space, time, and synchronization decoupling dimensions of event systems contribute to event transfer. I define two new types of problematic coupling dimensions: the semantic coupling and the pragmatic coupling. They correspond to granular and labour-intensive agreements on event semantics and contexts by humans involved in developing and using the event system. Such agreements may not be feasible in large-scale environments such as the IoT. Current approaches to semantic and context interoperability in event processing are coupled on one or more of these two dimensions, limiting scalability.

This thesis concerns two research questions of how semantic and pragmatic coupling can be loosened effectively and efficiently. I propose an approach based on four elements: subsymbolic semantics, free tagging, dynamic native enrichment, and approximation. A statistical vector-space model of semantics is built from a textual corpus that reflects the mutual understanding of event producers and consumers. Subscriptions are consumers' expressions to match events of interest. Free tags, called themes, are added to events and subscriptions to improve their meanings. Subscriptions are enhanced with indications of context to dynamically enrich events. Terms in events and subscriptions are decoded into their subsymbolic vector representations that are then matched using an approximate probabilistic matcher, resulting in scored relevance of events to subscriptions.

The hypotheses underlying the proposed approach are empirically validated within synthetic and real-world scenarios from the smart cities and energy management domains. A loose semantic coupling can be achieved with coarse-grained agreements on statistical semantics, with 100 approximate subscriptions compensating for 74,000 exact subscriptions otherwise needed. The approximate matcher achieves a magnitude of 1,000 events/sec of throughput, and an effectiveness of over than 95%  $F_1$ Measure. Using thematic tagging, a lightweight amount of tags is needed: around 2–7 for events and 2–15 for subscriptions. It delivers a magnitude of 800 events/sec in the worst case and 85%  $F_1$ Measure as opposed to 62% worst-case for non-thematic processing.

Loose pragmatic coupling is achieved with 4 high-level clauses in the subscriptions to guide the dynamic enricher. They specify the source, the retrieval method, the context search strategy, and the fusion method of events with context. Enrichment is instantiated with spreading activation in Linked Data graphs. It is tested with 24,000 events, with live DBpedia, a structured version of Wikipedia, as a contextual source. It reaches an efficiency and effectiveness of 7 times more than other instantiations of the enricher.

The research discussed in this thesis has been deployed in working systems for energy and water management where it has had an impact on real world applications. The model has also been developed into the concept of thingsonomies, an architecture for the Internet of Things that can tackle variety and allows IoT systems to evolve into large-scale, heterogeneous, and loosely coupled environments.

*Dedicated to*

*Mum, Dad, and Sonya*



# *Acknowledgements*

I would like to thank my advisor Dr. Edward Curry who has been an excellent guidance throughout my PhD years. The time he dedicated and the advice he has always been willing to give towards the completion of this work are priceless. Not all pieces of advice could be expressed; some are in fact put in practice, and that is how they are conveyed. Setting a great role model for me as a professional scientist, Dr. Curry helped me understand the true beauty of science.

My thanks also go to Dr. Sean O’Riain for his support in the early years of my PhD. My great appreciation is also for my graduate research committee members: the research director Prof. Stefan Decker, Prof. Manfred Hauswirth, Dr. Adegboyega Ojo, and Dr. Sami Bhiri. Their great input at various stages of the PhD has been a valuable help keeping this work on the track towards completion. I am also very grateful to my examiners Prof. Gordon Blair and Dr. John Breslin for their time to evaluate this work and their precious feedback and comments, and thanks to Dr. Paul Buitelaar for chairing the process.

I would like to thank Dr. André Freitas for his inspirational insights, and for his generous provision of the distributional semantics index used in this work. I also thank Mauricio Banduk, Yongrui Qin, and Kalpa Gunaratna who contributed to the development of COLLIDER demos. Thanks go also to Dr. Brian Davis, Wassim Derguech, Gerard Conway, Niall O’Brolchain, Suad Darra, Umair ul Hassan, and Housam Ziad for their proofreading comments towards the preparation of the final version of this thesis.

I would like to forward my acknowledgement to the funding agencies who supported this work at different stages: Science Foundation Ireland, Enterprise Ireland, and the European Commission. I would also like to thank every member of my institute, the Insight Centre at NUI Galway, and formerly DERI. Fellow researchers, students, staff, and friends, all made the place an exceptional environment for pursuing the PhD venture. The great value of discussions, feedback, and support on every occasion have been priceless.

I thank Dr. Ammar Kheirbek for his long-lasting inspiration to pursue science. Warm thanks go to my family: Mum, Dad, brothers and sister who have always been close, and my final thanks are to my beloved wife, Sonya Abbas, whose existence beside me at every moment made the journey a beautiful and enjoyable one.



# Contents

<b>Declaration of Authorship</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>Contents</b>	<b>xi</b>
<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xix</b>
<b>Abbreviations</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 General Introduction . . . . .	1
1.2 Motivation and Problem Overview . . . . .	2
1.3 The Event Processing Computational Paradigms . . . . .	3
1.4 Problem Description . . . . .	6
1.5 Core Requirements and Research Questions . . . . .	8
1.6 Existing Approaches . . . . .	9
1.7 Proposed Approach . . . . .	11
1.8 Hypotheses . . . . .	14
1.9 Research Methodology . . . . .	15
1.10 Contributions . . . . .	15
1.11 Thesis Outline . . . . .	17
1.12 Associated Publications . . . . .	18
<b>2 Problem Analysis: Crossing Boundaries in Open Distributed Systems</b>	<b>21</b>
2.1 Introduction . . . . .	21
2.2 Motivational Scenarios . . . . .	22
2.2.1 Scenario 1: Heterogeneous Energy Events . . . . .	23
2.2.2 Scenario 2: Incomplete Energy Events . . . . .	23
2.3 Challenges . . . . .	24
2.4 Significant Trends in the Data Landscape . . . . .	25

---

2.4.1	Internet of Things	25
2.4.2	Big Data	26
2.4.3	Common Characteristics	27
2.5	The Event Processing Paradigm	30
2.5.1	Evolution Towards Event Processing	31
2.5.2	The Information Flow Processing Domain	39
2.6	Terminology and Definitions	42
2.6.1	Event	42
2.6.2	Producers and Consumers	43
2.6.3	Subscriptions and Rules	44
2.6.4	Event Processing Engine	45
2.7	Traits of Large-Scale Event Processing	45
2.7.1	Distribution	45
2.7.2	Heterogeneity	46
2.7.3	Openness	47
2.8	The Principle of Decoupling	48
2.9	A Theory for Event Exchange	50
2.9.1	The Model of Communication	50
2.9.2	Event Exchange as Crossing System Boundaries	51
2.9.3	Discussion	53
2.10	Limitations of the Current Event Processing Paradigm	54
2.11	Requirements, Questions, and Scope	57
2.12	Chapter Summary	60
<b>3</b>	<b>Related Work</b>	<b>61</b>
3.1	Introduction	61
3.2	Requirements	62
3.3	Categories of Related Work	67
3.4	Content-Based Event Processing	68
3.4.1	Carzaniga et al. (SIENA)	68
3.4.2	Eugster et al.	71
3.4.3	Fiege et al. (Rebeca)	72
3.5	Concept-Based Event Processing	74
3.5.1	Petrovic et al. (S-ToPSS)	74
3.5.2	Wang et al. (OPS)	76
3.5.3	Zeng and Lei	77
3.5.4	Blair et al. (CONNECT)	79
3.6	Approximate Event Processing	81
3.6.1	Zhang and Ye (FOMatch)	81
3.6.2	Liu and Jacobsen (A-TOPSS)	82
3.6.3	Drosou et al. (PrefSIENA)	83
3.6.4	Wasserkrug et al.	85
3.7	Dedicated Event Enrichers	86
3.7.1	Schilling et al. (DHEP)	86
3.7.2	Hohpe and Woolf	88
3.8	Query-Based Event Fusion	89
3.8.1	Arasu et al. (CQL)	89

3.8.2	Teymourian et al. . . . .	90
3.8.3	Le-Phuoc et al. (CQELS) . . . . .	92
3.8.4	Anicic et al. (EP-SPARQL) . . . . .	93
3.9	Semantic and Context Event Transformation . . . . .	95
3.9.1	Freudenreich et al. (ACTrESS) . . . . .	95
3.9.2	Cilia et al. (CREAM) . . . . .	96
3.10	Discussion and Gap Analysis . . . . .	98
3.10.1	Gap Analysis at the Requirements Level . . . . .	98
3.10.2	Gap Analysis at the Features Level . . . . .	101
3.11	Other Relevant Approaches . . . . .	103
3.11.1	Schema Matching . . . . .	104
3.11.2	Approximate Query Answering Over Databases . . . . .	105
3.12	Chapter Summary . . . . .	106
<b>4</b>	<b>Approximate Semantic Event Matching and Dynamic Enrichment</b>	<b>107</b>
4.1	Introduction . . . . .	107
4.2	Main Models . . . . .	108
4.2.1	The Approximate Semantic Event Matching Model . . . . .	108
4.2.2	The Thematic Event Matching Model . . . . .	109
4.2.3	The Dynamic Native Event Enrichment Model . . . . .	110
4.3	Main Elements . . . . .	111
4.3.1	Subsymbolic Distributional Event Semantics . . . . .	111
4.3.2	Free Event Tagging . . . . .	112
4.3.3	Dynamic Native Event Enrichment . . . . .	113
4.3.4	Approximation . . . . .	115
4.3.5	Elements within the Event Flow Functional Model . . . . .	116
4.4	Scope . . . . .	119
4.5	Chapter Summary . . . . .	120
<b>5</b>	<b>The Approximate Semantic Event Matching Model</b>	<b>121</b>
5.1	Introduction . . . . .	121
5.2	Overview . . . . .	123
5.3	Distributional Semantics as a Loosely Coupled Event Semantic Model . . . . .	123
5.4	Subsymbolic Distributional Event Semantics . . . . .	124
5.4.1	Semiotic Systems for Symbols and Meanings . . . . .	124
5.4.2	Symbolic Representation of Meaning . . . . .	126
5.4.3	Conceptual Representation of Meaning . . . . .	130
5.4.4	Subconceptual Representation of Meaning . . . . .	138
5.4.5	How Subsymbolic Distributional Event Semantics Meets the Requirements . . . . .	139
5.5	Approximation . . . . .	139
5.5.1	Approximate Computing Versus Time . . . . .	140
5.5.2	Approximate Computing Versus Full Integration . . . . .	141
5.5.3	Limitations of Approximation . . . . .	143
5.5.4	How Approximation Meets the Requirements . . . . .	143
5.6	Event Flow Model . . . . .	144
5.7	Event Model . . . . .	145

5.8	Language Model	146
5.9	Matching	147
5.9.1	First-Line Matchers and Similarity Matrices	149
5.9.2	Global Aggregator and the Combined Similarity Matrix	150
5.9.3	Top-1 Matcher	151
5.9.4	Top- $k$ Matcher	152
5.9.5	Matcher Extensibility	155
5.10	Evaluation	156
5.10.1	Evaluation Metrics	156
5.10.2	Methodology for Effectiveness Evaluation	158
5.10.3	Methodology for Efficiency Evaluation	162
5.10.4	Results	162
5.11	Chapter Summary	166
<b>6</b>	<b>The Thematic Event Matching Model</b>	<b>169</b>
6.1	Introduction	169
6.2	The Thematic Model	170
6.3	Free Event Tagging	172
6.3.1	The Web and Social Tagging	172
6.3.2	Metadata Generation and Fixed Taxonomies	173
6.3.3	Folksonomies	174
6.3.4	Limitations of Free Event Tagging	175
6.3.5	How Free Event Tagging Meets the Requirements	176
6.4	Model Instantiation	177
6.4.1	Distributional Semantics	178
6.4.2	Themes	179
6.4.3	Thematic Event Model	179
6.4.4	Thematic Language Model	180
6.4.5	Thematic Matching Model	181
6.5	Parametric Vector Space Model	182
6.5.1	Distributional Vector Space Model	182
6.5.2	Thematic Projection	183
6.5.3	Distance and Semantic Relatedness	184
6.6	Evaluation	185
6.6.1	Evaluation Metrics	186
6.6.2	Methodology	187
6.6.3	Results	191
6.7	Chapter Summary	195
<b>7</b>	<b>The Dynamic Native Event Enrichment Model</b>	<b>197</b>
7.1	Introduction	197
7.2	Overview of the Dynamic Native Event Enrichment Model	198
7.3	Dynamic Native Event Enrichment	199
7.3.1	Information Incompleteness	200
7.3.2	Dimensions of Incompleteness	201
7.3.3	Unified and Native Event Enrichment	204
7.3.4	Late Dynamic Event Enrichment	206

---

7.3.5	Limitations of Dynamic Native Event Enrichment . . . . .	206
7.3.6	How Dynamic Native Event Enrichment Meets The Requirements	207
7.4	Approximate Computing Versus Incomplete Information . . . . .	207
7.5	Elements of Enrichment . . . . .	208
7.5.1	Determination of the Enrichment Source . . . . .	208
7.5.2	Retrieval of Information Items from the Enrichment Source . . . . .	209
7.5.3	Finding Complementary Information in the Enrichment Source . . . . .	210
7.5.4	Fusion of Complementary Information with the Event . . . . .	210
7.6	Event and Enrichment Flow Model . . . . .	211
7.7	Formal Model . . . . .	214
7.8	A Linked Data Instantiation . . . . .	218
7.8.1	Event Model . . . . .	218
7.8.2	Enrichment Source Model . . . . .	220
7.8.3	Matching Element Model . . . . .	220
7.8.4	Enrichment Element Model . . . . .	221
7.8.5	Native Enricher . . . . .	222
7.9	Evaluation . . . . .	224
7.9.1	Event Set and Enrichment Source . . . . .	225
7.9.2	Unified Subscriptions Set . . . . .	226
7.9.3	Minimal Successfully Enriched Events Construction . . . . .	227
7.9.4	Evaluation Metrics . . . . .	228
7.9.5	Results . . . . .	228
7.10	Chapter Summary . . . . .	230
<b>8</b>	<b>Prototype and Use Cases</b>	<b>231</b>
8.1	Introduction . . . . .	231
8.2	Internal Architecture . . . . .	232
8.2.1	Input and Output Adapters . . . . .	233
8.2.2	Language . . . . .	233
8.2.3	Enricher . . . . .	234
8.2.4	Single Event Matcher . . . . .	234
8.2.5	Pattern Matcher . . . . .	234
8.2.6	Event Player and Evaluator . . . . .	235
8.3	Thingsonomies for the Internet of Things . . . . .	235
8.4	Self-Configurable Energy Management Systems Use Case . . . . .	239
8.4.1	Event Processing Requirements in the Use Case . . . . .	240
8.4.2	COLLIDER Implementation for Energy Management . . . . .	242
8.4.3	Building an Energy Domain Corpus for Semantic Relatedness . . . . .	243
8.4.4	Validation within Self-Configurable Energy Management Systems . . . . .	245
8.5	Water Management Use Case . . . . .	245
8.5.1	Event Processing Requirements in the Use Case . . . . .	247
8.5.2	Linked Water Dataspace . . . . .	248
8.6	Reflections on COLLIDER in Use . . . . .	251
8.7	Chapter Summary . . . . .	252
<b>9</b>	<b>Conclusions and Future Work</b>	<b>255</b>
9.1	Thesis Summary . . . . .	255

---

9.2 Thesis Conclusions . . . . .	257
9.3 Contributions . . . . .	260
9.4 Limitations . . . . .	263
9.5 Future Work . . . . .	264

<b>Bibliography</b>	<b>267</b>
---------------------	------------

# List of Figures

1.1	The event processing functional model . . . . .	5
1.2	Decoupling dimensions . . . . .	6
1.3	Trade-off between decoupling and event exchange across boundaries . . . . .	7
1.4	Dimensions of de/coupling . . . . .	8
1.5	The proposed event processing model . . . . .	13
2.1	The IFP functional model . . . . .	40
2.2	Event semantic heterogeneity . . . . .	47
2.3	Dimensions of decoupling . . . . .	48
2.4	Schematic diagram of Shannon-Weaver general communication system . . . . .	50
2.5	Boundaries in knowledge exchange . . . . .	52
2.6	Trade-off between decoupling and knowledge exchange across boundaries . . . . .	56
2.7	Dimensions of de/coupling . . . . .	57
4.1	The main models of the proposed approach . . . . .	108
4.2	The main elements within the proposed event processing approach . . . . .	117
5.1	A sign of two parts: a signifier and a signified . . . . .	125
5.2	Event flow model . . . . .	145
5.3	The approximate semantic event matcher model . . . . .	149
5.4	Top- $k$ by an evolving frontier algorithm . . . . .	152
5.5	Methodology for effectiveness evaluation . . . . .	160
5.6	Top- $k$ time vs. $k$ . . . . .	163
5.7	Top- $k$ time vs. $n$ . . . . .	163
5.8	Top- $k$ time vs. $m$ . . . . .	163
5.9	Optimized matcher . . . . .	163
5.10	Effectiveness vs. % of $\sim$ . . . . .	164
5.11	Efficiency vs. % of $\sim$ . . . . .	164
5.12	Effectiveness vs. the number of approximate subscriptions . . . . .	165
5.13	Efficiency vs. the number of approximate subscriptions . . . . .	165
6.1	Thematic event processing . . . . .	171
6.2	Thematic event matching model . . . . .	177
6.3	Parametric distributional vector space for thematic event processing . . . . .	183
6.4	Evaluation methodology for thematic event processing . . . . .	188
6.5	Effectiveness of thematic matcher . . . . .	191
6.6	Effectiveness sample error . . . . .	192
6.7	Throughput of thematic matcher . . . . .	193
6.8	Throughput sample error . . . . .	194

---

7.1	Unified and native enrichment model . . . . .	213
7.2	The universe $U$ , the event $e$ , the enrichment source $ES$ , the world $W$ , the enrichment view $HVS$ , and a matching view $MVS$ . . . . .	215
7.3	An example Linked Data event . . . . .	219
7.4	An example Linked Data enrichment source . . . . .	220
7.5	The base path-shaped graph used to generate the matching elements of the subscriptions . . . . .	226
7.6	The combined $F_5Score$ achieved by the enrichment approaches for each subscription . . . . .	229
8.1	Internal architecture for the COLLIDER system . . . . .	232
8.2	An example thingsonomy for tagging a device's events . . . . .	235
8.3	Architecture for loosely coupled semantic normalization for Internet of Things software . . . . .	236
8.4	The role of COLLIDER in the Self-Configurable Energy Management Systems Use Case . . . . .	239
8.5	An energy domain-specific scenario . . . . .	241
8.6	COLLIDER and semantic relatedness within the SENSE project . . . . .	242
8.7	COLLIDER simulation engine . . . . .	243
8.8	Building an energy domain-specific corpus and semantic relatedness for energy management . . . . .	244
8.9	Waternomics overall architecture . . . . .	246
8.10	Waternomics Linked Water Dataspace components view . . . . .	248
8.11	Waternomics Linked Dataspace architecture . . . . .	250

# List of Tables

3.1	Requirements Dimensions as Defined by Previous Work . . . . .	66
3.2	Requirements as Addressed by Previous Work . . . . .	99
3.3	Features as Addressed by Previous Work . . . . .	102
4.1	Elements and Models of the Proposed Approach with Respect to Requirements and Research Questions . . . . .	118
5.1	Semantic Models and Requirements . . . . .	140
5.2	Approximation and Requirements . . . . .	144
5.3	Base Concepts for Effectiveness Evaluation . . . . .	157
5.4	Sensor Capabilities . . . . .	159
5.5	Approximate versus Exact Model . . . . .	166
6.1	Example Social Tagging Websites . . . . .	173
6.2	Free Event Tagging and Requirements . . . . .	176
6.3	Sensor Capabilities . . . . .	187
6.4	Thematic Model versus Exact Model . . . . .	189
7.1	Dynamic Native Event Enrichment and Requirements . . . . .	207
7.2	Matching Elements of the Unified Subscriptions Set . . . . .	227



# Abbreviations

<b>6LoWPAN</b>	<b>IPv6 over Low Power Wireless Personal Area Networks</b>
<b>AI</b>	<b>Artificial Intelligence</b>
<b>AOP</b>	<b>Aspect-Oriented Programming</b>
<b>BAM</b>	<b>Business Activity Monitoring</b>
<b>BMS</b>	<b>Building Management System</b>
<b>CCPP</b>	<b>Composite Capabilities/ Preference Profiles</b>
<b>CEP</b>	<b>Complex Event Processing</b>
<b>CIM</b>	<b>Common Information Model</b>
<b>CoAP</b>	<b>Constrained Application Protocol</b>
<b>CORBA</b>	<b>Common Object Request Broker Architecture</b>
<b>CSR</b>	<b>Corporate Social Responsibility</b>
<b>CWA</b>	<b>Closed World Assumption</b>
<b>DACE</b>	<b>Distributed Asynchronous Computing Environment</b>
<b>DAHP</b>	<b>Database-Active Human-Passive</b>
<b>DERI</b>	<b>Digital Enterprise Research Institute</b>
<b>EAI</b>	<b>Enterprise Application Integration</b>
<b>ECA</b>	<b>Event Condition Action</b>
<b>EDBC</b>	<b>Event-Driven Backward Chaining</b>
<b>EI</b>	<b>Enterprise Integration</b>
<b>EPL</b>	<b>Event Processing Language</b>
<b>EPTS</b>	<b>Event Processing Technical Society</b>
<b>ESA</b>	<b>Explicit Semantic Analysis</b>
<b>FDCML</b>	<b>Field Device Configuration Markup Language</b>
<b>GPS</b>	<b>Global Positioning System</b>
<b>GSDML</b>	<b>Generic Station Description Markup Language</b>

---

<b>HADP</b>	<b>H</b> uman- <b>A</b> ctive <b>D</b> atabase- <b>P</b> assive
<b>HR</b>	<b>H</b> uman <b>R</b> esources
<b>IEC</b>	<b>I</b> nternational <b>E</b> lectrotechnical <b>C</b> ommission
<b>IETF</b>	<b>I</b> nternet <b>E</b> ngineering <b>T</b> ask <b>F</b> orce
<b>IFP</b>	<b>I</b> nformation <b>F</b> low <b>P</b> rocessing
<b>IMDB</b>	<b>I</b> nternet <b>M</b> ovie <b>D</b> ata <b>B</b> ase
<b>IoT</b>	<b>I</b> nternet <b>o</b> f <b>T</b> hings
<b>IP</b>	<b>I</b> nternet <b>P</b> rotocol
<b>IR</b>	<b>I</b> nformation <b>R</b> etrieval
<b>ITU</b>	<b>I</b> nternational <b>T</b> elecommunication <b>U</b> nion
<b>J2EE</b>	<b>J</b> ava <b>P</b> latform <b>E</b> nterprise <b>E</b> dition
<b>JMS</b>	<b>J</b> ava <b>M</b> essaging <b>S</b> ervice
<b>JSON-LD</b>	<b>J</b> ava <b>S</b> cript <b>O</b> bject <b>N</b> otation for <b>L</b> inked <b>D</b> ata
<b>LEI</b>	<b>L</b> inked <b>E</b> nergy <b>I</b> ntelligence
<b>LSA</b>	<b>L</b> atent <b>S</b> emantic <b>A</b> nalysis
<b>LWD</b>	<b>L</b> inked <b>W</b> ater <b>D</b> ataspace
<b>MoM</b>	<b>M</b> essage-oriented <b>M</b> iddleware
<b>OECD</b>	<b>O</b> rganization for <b>E</b> conomic <b>C</b> ooperation and <b>D</b> evelopment
<b>OMG</b>	<b>O</b> bject <b>M</b> anagement <b>G</b> roup
<b>OWA</b>	<b>O</b> pen <b>W</b> orld <b>A</b> ssumption
<b>OWL</b>	<b>W</b> eb <b>O</b> ntology <b>L</b> anguage
<b>PVSM</b>	<b>P</b> arametric <b>V</b> ector <b>S</b> pace <b>M</b> odel
<b>RDF</b>	<b>R</b> esource <b>D</b> escription <b>F</b> ramework
<b>RDFS</b>	<b>R</b> DF <b>S</b> chema
<b>RFID</b>	<b>R</b> adio <b>F</b> requency <b>I</b> dentification
<b>RPC</b>	<b>R</b> emote <b>P</b> rocedure <b>C</b> all
<b>SA</b>	<b>S</b> preading <b>A</b> ctivation
<b>SOA</b>	<b>S</b> ervice- <b>O</b> riented <b>A</b> rchitectures
<b>SPARQL</b>	<b>S</b> PARQL <b>P</b> rotocol <b>A</b> nd <b>R</b> DF <b>Q</b> uery <b>L</b> anguage
<b>SQWRL</b>	<b>S</b> emantic <b>Q</b> uery <b>E</b> nanced <b>W</b> eb <b>R</b> ule <b>L</b> anguage
<b>SVD</b>	<b>S</b> ingular <b>V</b> alue <b>D</b> ecomposition
<b>TF/IDF</b>	<b>T</b> erm <b>F</b> requency / <b>I</b> nverse <b>D</b> ocument <b>F</b> requency
<b>UI</b>	<b>U</b> ser <b>I</b> nterface

<b>UMLS</b>	<b>Unified Medical Language System</b>
<b>UNA</b>	<b>Unique Name Assumption</b>
<b>URI</b>	<b>Uniform Resource Identifier</b>
<b>VANET</b>	<b>Vehicular Ad-hoc Network</b>
<b>VSM</b>	<b>Vector Space Model</b>
<b>WSD</b>	<b>Word Sense Disambiguation</b>
<b>WWW</b>	<b>World Wide Web</b>



# Chapter 1

## Introduction

“The beginning is the most important part of the work.”

— Plato

### 1.1 General Introduction

In the recent years, there has been a tremendous increase in information sources and volume. The Organization for Economic Co-operation and Development (OECD) estimates that there will be about 50 billion devices connected to the Internet by 2020 [1]. This leads to challenges for information processing solutions as the volume, velocity, and variety of data increase. Smart cities [2], smart grids [3], smart buildings [4], and cyber-physical systems [5] are examples of active research topics. A technology enabler for such areas is represented by the Internet of Things [6].

A basic requirement to realize the IoT is an infrastructure of communication solutions and standards such as the Constrained Application Protocol (CoAP) by the Internet Engineering Task Force (IETF) [6]. Sensing technologies such as Radio-frequency Identification (RFID) form the basic infrastructure for IoT to map the world of things into the world of computationally processable information. There is a need for middleware that can abstract the application developers from the underlying technologies, and that is crucial to the adoption and evolution of IoT applications [6]. Among the technologies that contribute to this layer are Service-Oriented Architectures (SOA) [6] and event processing systems.

While significant efforts in the area of IoT come from communication and networking communities, there has been a growing realization that the challenges of the IoT will be more prevalent at the data level [7] including data collection, management, and analytics. At this level, IoT can be linked with the area of Big Data, signified by the three main dimensions of volume, velocity, and variety.

The trends of IoT and Big Data signify a considerable shift in the characteristics of information production, communication, and consumption. Such a shift is characterized by a set of aspects: an increasing number of sources, a growing heterogeneity, an increasing number of users, a decentralized organization of users, information incompleteness, and a timeliness requirement.

Given this shift in the data landscape, there has been an evolution in the information processing paradigms required to meet these new challenges. Thus, the event processing paradigm has been motivated by a plethora of distributed applications that require on-the-fly and low-latency processing of information items. The event processing paradigm has evolved from the works of several communities including: active databases, reactive middleware, event-based software engineering, message-oriented middleware, and data stream management systems.

## 1.2 Motivation and Problem Overview

Assume the scenario of energy consumption events generated from sensors in a smart building. Event producers and consumers can use different terms to describe their events and information needs such as *'energy consumption'*, *'energy usage'*, and *'power consumption'* to refer to the same thing. Consumers may also expect contextual information in events, which are not complete, such as the *'room'* or the *'floor'* where the event was originated.

To address these challenges, traditional event processing systems depend on explicit agreements on semantics and contexts (or pragmatics) between producers and consumers. Semantic agreements are manifested in the form of explicit semantic models such as taxonomies or ontologies. Pragmatic agreements are manifested by a distinction between events and background data, with data join or enrichment logic which are implemented in dedicated enrichers. Thus, this is tackled as an interoperability problem on the

levels of semantics and pragmatics. Nonetheless, large-scale event processing environments are open, distributed, and heterogeneous in semantics and contexts. Agreements may not be feasible in large-scale environments such as the IoT.

A fundamental principle of the event-based interaction paradigm is the use of the event to decouple producers and consumers. Event producers and consumers do not know each other and are decoupled in space, time, and synchronization to enable scalable deployments. Events are not a mere exchange of symbols, but they are boundary objects that convey meanings signified by symbols. I define two new types of problematic coupling dimensions: the semantic coupling and the pragmatic coupling. They correspond to granular and labour-intensive agreements on event semantics and contexts by humans involved in developing and using the event system.

Current approaches to semantic and context interoperability in event processing are coupled on one or more of these two dimensions. Human agents are needed in the loop to cross semantic and pragmatic boundaries through explicit agreements on event types, properties, values, and contexts, introducing coupling into these systems. This coupling limits the paradigm and contradicts the fundamental basis of decoupling for scalability.

Thus, decoupling is required to enable scalability. Coupling on the other hand, such as agreements, is necessary to enable effective event-based communication between producers and consumers. The problem tackled in this thesis is *loosening semantic and pragmatic coupling between event producers and consumers to enable scalable deployments of event processing systems in open, distributed, and heterogeneous environments and allow effective event-based communication at the same time.*

### 1.3 The Event Processing Computational Paradigms

The event processing paradigm has evolved through the work of several communities as suggested by Cugola and Margara [8], Hinze et al. [9], Etzion [10, p. 16–21], Eugster et al. [11], and Mühl et al. [12, p. 7–8]. Elements of the paradigm can be found in the following paradigms:

- *Active Databases* started to appear during the late twentieth century [13] to move active logic from the application layer into the database layer to avoid its redundancy among distributed applications which affects maintenance.
- *Reactive Middleware* are asynchronous and decoupled extensions that have been added to the existing middleware systems like the Java Platform Enterprise Edition (J2EE) [14] and the Real-time CORBA Event Service [15]. This is to accommodate the case where data and services are not tied to static network nodes as assumed in the Remote Procedure Call (RPC) paradigm [16].
- *Event-based Software Engineering* adopts an *implicit method invocation* model [17] which is widely used now in enterprise application integration [18], aspect-oriented programming [19], and graphical user interfaces [20].
- *Message-Oriented Middleware* motivated by the Internet becoming a platform for dynamic distributed applications such that a decoupled interaction scheme has become crucial to the development of large-scale applications. Eugster et al. [11] give decoupling a high importance concerning scalability, recognizing three dimensions of decoupling: *space*, *time*, and *synchronization* which are concerned with addresses, activity time, and blocking respectively. Thus, the publish/subscribe paradigm evolved to overcome this issue [11].
- *Data Stream Management Systems* emerged as active databases do not scale under high rates of database updates or a large number of rules [8]. Streams form the basic concepts in DSMS as opposed to tables in conventional databases. DSMSs adopt an interaction paradigm based on *continuous queries* where users register a set of queries with the DSMS and data items are homogeneous in a stream, they do not typically have temporal or causal semantics, and languages are typically of a transformation nature.
- *Complex Event Processing* associate a specific semantics to their data items: they represent *events*. I follow the Event Processing Technical Society (EPTS)'s glossary [21] definition of an event as: "An object that represents, encodes, or records an event, generally for the purpose of computer processing." A Complex Event Processing (CEP) engine emphasizes the matching of event *patterns* specifically with ordering conditions such as temporal sequencing and causal relationships.

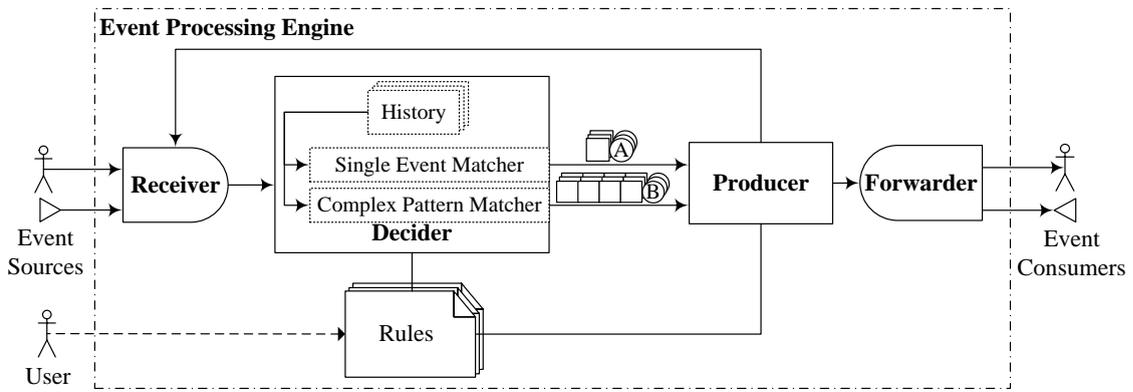
FIGURE 1.1: The event processing functional model <sup>1</sup>

Figure 1.1 presents an elaboration of Cugola and Margara’s functional model of event processing [8]. It shows the main functionalities of an event engine. *Event Sources*, which can be human, software, or hardware agents, create events. Events get received by the *Receiver*, which sends the events to the *Decider*. The *Decider* is responsible for the detection of conditions or patterns, which hold in single events or a set of events, according to the condition parts of the *Rules* registered in the engine. When a condition is detected in the *Decider*, the participating events that caused the detection are propagated along with the condition to the *Producer*, refer to *A* and *B* in Figure 1.1.

The *Producer* generates an event as dictated by the action parts of the *Rules* whose conditions or patterns are detected with possibly binding of placeholders with actual values from events. The generated events may feed back to the *Receiver* and/or propagate to the *Forwarder* which sends them to the external event *Consumers*, which may be human agents, software applications such as user interfaces, or hardware agents. The *Decider* keeps in its working memory a *History* of events, which may be eligible to trigger a detection. The *Single Event Matcher* detects only single events that match some filtering conditions while the *Complex Pattern Matcher* detects a pattern of events such as the sequence of two or more events that have passed single event matching.

Large-scale event processing systems feature three main technical traits: distribution, heterogeneity, and openness. A fundamental principle of the event-based interaction paradigm is the use of the event to decouple producers and consumers. Eugster et al. [11] define decoupling as “removing all explicit dependencies between the interacting participants.” The true impact of this principle is the increase of scalability [11].

<sup>1</sup>Adapted from Cugola and Margara’s functional model [8]

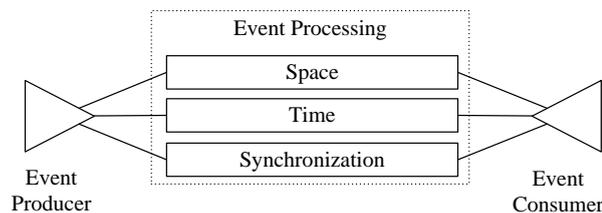


FIGURE 1.2: Decoupling dimensions

Eugster et al. [11] recognize three dimensions of decoupling as shown in Figure 1.2: *space*, *time*, and *synchronization* which are concerned with addresses, activity time, and blocking, respectively. Decoupling results in dependencies on events and the now autonomous events may lead to ambiguities in semantics, which require participants to collaborate again to resolve. This leads into limitations in scalability and undermining the fundamental reason participants are decoupled.

## 1.4 Problem Description

The first problem arises when events and subscriptions contain heterogeneous terms and need to cross significant semantic boundaries. For instance, events contain terms such as ‘*energy consumption*’, ‘*energy usage*’, and ‘*power consumption*’ to refer to the same thing. The second problem comes from the nature of a decoupled event processing system, as data consumers in many situations need more information than that is included in events. For instance, when a user is interested in situations where energy consumption of a building is excessive, the user tends to include higher level business concepts in their subscriptions to events. Examples of these are the ‘*room*’ or the ‘*floor*’ where the event was originated, or the ‘*business unit*’ or ‘*project*’ with which the device is associated. The events do not have information about the ‘*floor*’ to answer the subscription.

Event agents form an overall system of systems that have boundaries which they have to cross in order to communicate with other systems. Such boundaries are syntactic, semantic, and pragmatic. Events are not a mere exchange of symbols, but rather meanings signified by symbols. Events must effectively cross the three levels of boundaries to establish communication between event agents. I argue that the current event processing paradigm is focused on crossing lower boundaries, i.e. syntactic, for achieving the task of event transfer rather than that of event-based communication. Thus, human

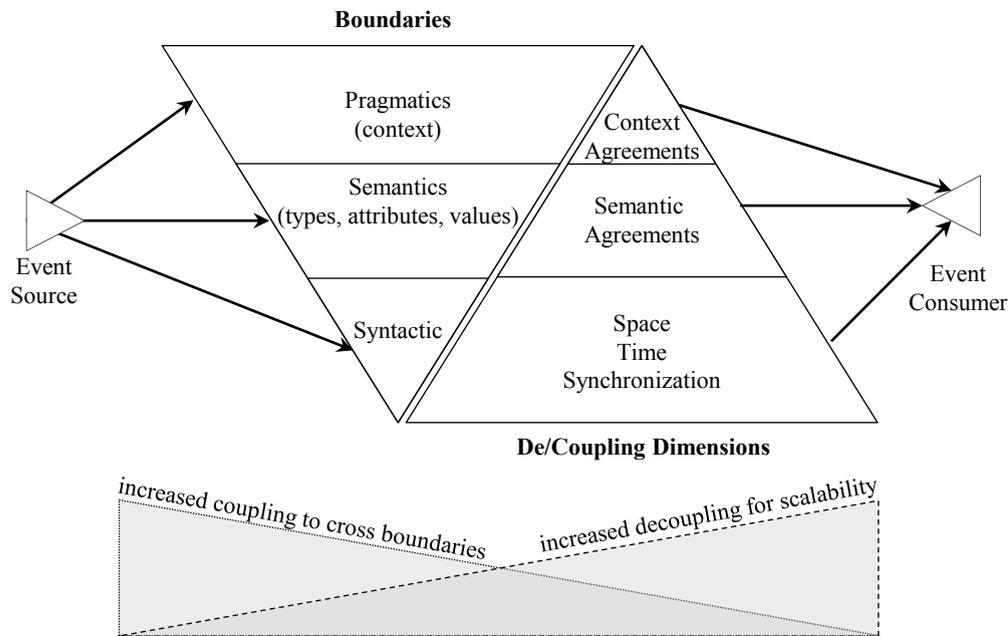


FIGURE 1.3: Trade-off between decoupling and event exchange across boundaries

agents are needed in the loop to cross semantic and pragmatic boundaries which leads to limiting the paradigm as these tasks are external to it rather than being at its core.

Space, time, and synchronization decoupling dimensions contribute to event transfer across syntactic boundaries only. A trade-off can be concluded between decoupling and event exchange across boundaries as Figure 1.3 illustrates. I recognize two new coupling dimensions that exist in current event processing systems as shown in Figure 1.4:

- *Semantic Coupling* is the amount of agreement between participants in the event processing environment on mappings between symbols used in event messages and the meanings they refer to.

For instance, a high semantic coupling results from the granular agreements on the individual meanings of the terms ‘energy’, ‘power’, and ‘electricity’ as follows:

‘energy’  $\Rightarrow$  usable power that comes from heat or another source.

‘power’  $\Rightarrow$  a source or means of supplying energy.

‘electricity’  $\Rightarrow$  a wire-carried energy used to operate appliances, machines, etc.

A looser semantic coupling can be achieved by establishing a quantifiable relationship between the three terms and meanings above, e.g. frequency of co-occurrence, and having a coarse-grained agreement over this relationship, through agreeing on a corpus that encompasses the terms in use.

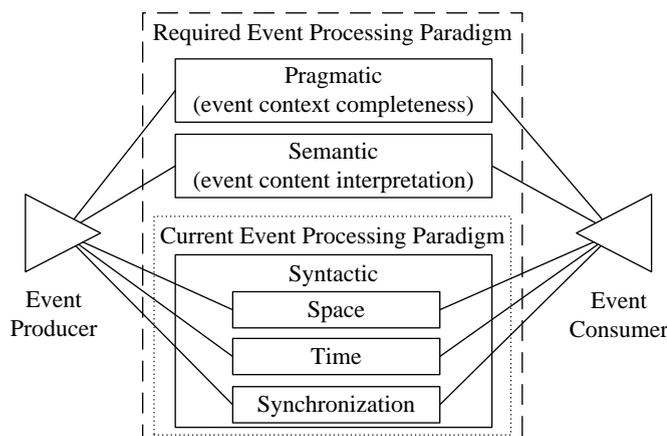


FIGURE 1.4: Dimensions of de/coupling

- *Pragmatic Coupling* is the amount of agreement between participants in the event processing environment on contextual information needed to complement event messages to evaluate users interests. For instance, an event consumer and producer who agree that an energy event should have both the 'room' and the 'floor' of the energy consuming device, assume more pragmatic coupling than agreeing on having only the 'room' of the device in the event.

## 1.5 Core Requirements and Research Questions

This work tackles four main requirements as follows:

- *R1*. Loose coupling of event processing systems on the semantic dimension. It can be defined as a low cost to define and maintain rules with respect to the use of terms, and to building and agreeing on an event semantic model.
- *R2*. Loose coupling of event processing systems on the pragmatic dimension. It can be defined as a low cost to define and maintain the context parts of rules, and to agree on contextual data that is needed in events.
- *R3*. Efficiency of event processing. It can be defined as the timeliness in matching event semantics, and precision in integrating contextual data with events.
- *R4*. Effectiveness of event processing. This can be quantified by the proportion of true positives and negatives achieved by the decider (or matcher), and the effectiveness in completing events with contextual data.

To meet requirements, two main research questions need to be investigated:

- *Q1.* The first research question is concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rules and their meanings but rather a form of statistical loose agreements on the meanings (Requirement *R1*). The question is how to achieve timely event matching (Requirement *R3*) with high true positives and negatives (Requirement *R4*) in such a loosely semantically coupled environment?
- *Q2.* The second research question is concerned with the case when event producers and consumers do not have equal assumptions on the amount of contextual information included in events and how much they are complete with respect to evaluating some consumers' rules (Requirement *R2*). The question is how to complement events with context at high precision (Requirement *R3*) and completeness needed to meet consumers expectations (Requirement *R4*) in such a loosely contextually coupled environment?

## 1.6 Existing Approaches

The event processing literature related to this work can be classified into two major categories: (1) approaches to cross semantic boundaries of event-based systems, and (2) approaches to cross pragmatic boundaries of event-based systems. Approaches to cross semantic boundaries of event-based systems can be classified as follows:

- *Content-based event processing:* In content-based event processing, event sources and consumers use the same event types, attributes, and values without any additional description of meaning external to the rules and events. The main works in this category are those by Carzaniga et al. [22] (SIENA), Eugster et al. [23], and Fiege et al. [24] (Rebeca). Such approaches are effective with the timely matching and routing of events but they assume an implicit agreement on semantics of events outside of the event engine, which is a type of semantic coupling that does not scale in heterogeneous environments.
- *Concept-based event processing:* In this category participants can use different terms and values and still expect matchers to be able to match them properly

thanks to explicit knowledge representations such as thesauri and ontologies that encode semantic relationships between terms. The main works in this category are those by Petrovic et al. [25] (S-ToPSS), Wang et al. [26] (OPS), Zeng and Lei [27], and Blair et al. [28] (CONNECT). Given agreements on explicit models, efficient and effective detection of positive and negative matching can be achieved. Nonetheless, agreements on explicit models may become an unfeasible task to achieve due to high levels of heterogeneity at large scales.

- *Approximate event processing:* Approaches in this category are distinguished by a matching model that is not Boolean. The main works in this category are those by Zhang and Ye [29] (FOMatch), Liu and Jacobsen [30, 31] (A-TOPSS), Drosou et al. [32] (PrefSIENA), and Wasserkrug et al. [33]. These approaches reduce semantic coupling due to their ability to deal with uncertainties of users about semantics. Time efficiency is high, but effectiveness is lower due to the approximate model, which allows some false positive/negatives to occur.

Approaches to cross pragmatic boundaries of event systems can be classified as follows:

- *Dedicated enrichers:* This category is mainly concerned with event enrichment via ad-hoc dedicated agents that are tailored specifically to particular situations. The main works in this category are those by Schilling et al. [34] (DHEP), and Hohpe and Woolf [18]. These approaches mainly focus on integrating events with their contexts and can efficiently and effectively complete events to be matched later. However, they depend on an implicit understanding of the pragmatics around events that are implemented by developers through a set of ad-hoc enrichment logic. This keeps context handling out of the event engine and represents a level of coupling that hinders scalability where significant pragmatic boundaries exist.
- *Query-based fusion:* Approaches in this category adopt declarative languages similar to SQL. Such languages support operators of semantics similar to relational join, enabling the fusion of streams of events with background context data. The main works in this category are those by Arasu et al. [35] (CQL), Teymourian et al. [36], Le-Phuoc et al. [37] (CQELS), and Anicic et al. [38] (EP-SPARQL). These approaches are effective and efficient in integrating events with their contexts. However, a full understanding of event contexts is assumed and encoded by

developers as join statements and thus causes a pragmatic coupling that makes them not scalable with contextual boundaries.

- *Semantic and context transformation:* Approaches in this category handle events individually and perform a set of transformations on them to move from one semantic and pragmatic model to another. The main works in this category are those by Freudenreich et al. [39] (ACTrESS), and Cilia et al. [40, 41] (CREAM). These approaches consider semantics and contexts to have one nature and impact on event matching. They are effective and efficient in matching and completing the events. Nonetheless, semantic and pragmatic models that depend on ontologies and conversion functions require agreements which form a coupled mode that limits scalability in heterogeneous environments.

## 1.7 Proposed Approach

The proposed approach stands on three main models: the approximate semantic event matching model, the thematic event matching model, and the dynamic native event enrichment model. These models can be conceptually decomposed into four main elements:

- *Subsymbolic Distributional Event Semantics.* Assuming that semantic coupling can be quantified by the number of mappings between symbols, i.e. terms, and meanings, then a semantic model that condenses these mappings can be very useful. Ontological models require granular agreements on the symbol-meaning mappings while distributional vector space semantics leverages the statistics of terms co-occurrence, e.g. ‘power’ and ‘electricity’, in a large corpus to establish semantics [42] leaving event producers and consumers to loosely agree on the corpus as common knowledge.
- *Free Event Tagging.* This element allows users to adapt the conveyed events’ meanings in different domains and situations without introducing any coupling components between participants. This element, called thingsonomies, builds on the success of free tagging, known as folksonomies, within social media research [43]. For instance, the term ‘energy’ when used in an event tagged by the tags

{‘*building*’, ‘*appliance*’} helps the matcher distinguish the meaning of ‘*energy*’ and associate it with the domain of power management, rather than associating it with the domain of sport or diet for example.

- *Dynamic Native Event Enrichment*. Pragmatic coupling that is caused by mutual agreements on contextual information can be reduced by allowing the event processing system to discover contextual data dynamically. For instance, an *energy consumption event* could include information about the consuming ‘*device*’ and its ‘*power consumption*’. The event engine shall be able to dynamically look up the device in a building management system database to get information about the ‘*room*’ and ‘*floor*’ where the device exists. Thus, events are assumed to be incomplete and contextual data is dynamically added through an enrichment process that is moved to the core of the event engine.
- *Approximation*. Loose coupling introduces uncertainties to the event processing engine which results from not exactly knowing which event’s tuples shall be mapped to which subscription’s tuples, and which information can be assumed in an event that is incomplete. For instance, with the loose agreements on terms semantics, there are various possible mappings between an event and a subscription such as:

$$\sigma_1 = \{(\mathbf{device} = \mathbf{laptop} \leftrightarrow \text{device:computer}),$$

$$(\mathbf{room} = \mathbf{room\ 112} \leftrightarrow \text{office: room 112})\}$$

$$\sigma_2 = \{(\mathbf{device} = \mathbf{laptop} \leftrightarrow \text{office: room 112}),$$

$$(\mathbf{room} = \mathbf{room\ 112} \leftrightarrow \text{device:computer})\}$$

Each mapping has a different probability that reflects the uncertainty of the matching. The same applies to the uncertainty about which tuples complement an event. Approximation at the core of the event processing engine can tackle uncertainties and complement the elements mentioned above.

The main elements of the proposed approach can be unified and fit into the event processing functional model proposed by Cugola and Margara in [8] as shown in Figure 1.1. Figure 1.5 illustrates how the elements work together, along with non-impacted components, to fulfil the role of an event processing engine as follows:

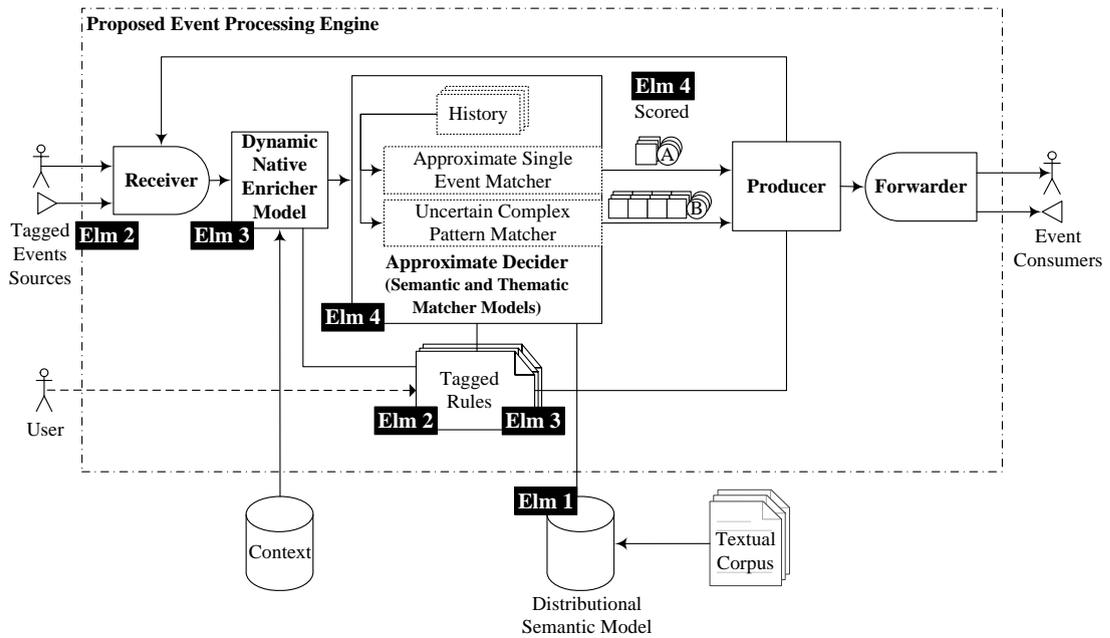


FIGURE 1.5: The proposed event processing model

- **Elm1** *Subsymbolic Distributional Event Semantics*. The actual distributional semantic model could be built outside of the event processing engine by indexing a textual corpus. The resulting model forms the basis to compare any two strings in events and subscriptions, as those get decoded into their subsymbolic vector representations, the basis for distance and similarity measures.
- **Elm2** *Free Tagging*. Events flowing from event sources, and subscriptions get tagged by users before they are considered for matching. Users use free tags to enhance events and subscriptions and improve their interpretation and meaning disambiguation by the matcher.
- **Elm3** *Dynamic Native Event Enrichment*. A new functional component, the *Dynamic Enricher*, is added to the model. Enrichment guidance elements are added to the subscriptions to identify enrichment sources, retrieval, search, and fusion mechanisms of contextual information with events. Events get enriched before being passed to the decider.
- **Elm4** *Approximation*. Events are matched in the decider against subscriptions and the result of event matching is a scored ranking of events that signifies their relevance to each subscription.

## 1.8 Hypotheses

The proposed approach stands on four main hypotheses respectively with the elements.

*H1.* Subsymbolic distributional event semantics decreases the cost needed to define and maintain rules with respect to the use of terms, and to build and agree on an event semantic model more than symbolic semantic models; and at the same time it can achieve timely event matching with high true positives and negatives of magnitudes comparable to that of event processing based on semantic models.

*H2.* Free tagging of events and subscriptions does not add to the cost of defining and maintaining rules with respect to the use of terms, and the cost of building and agreeing on an event semantic model required by subsymbolic event semantics; and at the same time it can achieve timely event matching with high true positives and negatives more than event processing based on non-tagged subsymbolic event semantics.

*H3.* Dynamic native event enrichment decreases the cost needed to define and maintain the context parts of rules, and to agree on contextual data that is needed in events more than dedicated enrichers; and at the same time it can achieve high precision integration of event context with high completeness of events comparable to that of event processing based on dedicated enrichers.

*H4.* Approximate event processing can operate in event environments with low-cost agreements on event semantics and pragmatics more than exact event processing; and at the same time achieve timely event matching with high true positives and negatives, and high precision integration of event context with high completeness of events, comparable to that of event processing based on exact models.

The rationales for the hypotheses are detailed in Chapters 5, 6, and 7.

## 1.9 Research Methodology

The methodology followed in this work consists of the following steps:

1. Review the literature and the related work and define the problems of semantic and pragmatic coupling in event systems.
2. Formulate the research questions of effective and efficient event processing in loosely coupled environments.
3. Formulate the main four hypotheses to answer the questions and design experiments for testing.
4. Synthesize a workload of events with high semantic heterogeneity and velocity. An evaluation event set of 50,000 events has been semantically expanded out of seed event sets from actual deployments of IoT smart city, energy management, building, and relevant datasets, to evaluate the approximate semantic event matching model. Similarly, 14,743 events were generated to evaluate the thematic event matching model, and 24,000 events from DBpedia, a Linked Data version of Wikipedia have been used for evaluating dynamic native event enrichment.
5. Synthesize a workload of situations of interest representative to the supposed users of IoT and the need for context.
6. Implement the four proposed elements: subsymbolic semantics, tagging, dynamic native enrichment, and approximation in an event processing engine.
7. Test the workload and compare with suitable baselines to validate the hypotheses.
8. Recognize the trade-offs and limitations of the proposed approach and where it outperforms the current event processing paradigm in the supposed environment.

## 1.10 Contributions

The contribution of this work is manifold:

- A new analytical framework of event systems based on communication models that cross system boundaries, and loose semantic and pragmatic coupling.

- An effective and efficient approximate event processing model:
  - *Loose semantic coupling*: coarse-grained agreement on semantics where 100 approximate subscriptions can compensate for 74,000 exact subscriptions otherwise needed.
  - *Efficiency*: a magnitude of 1,000 events/sec of throughput reflecting an efficient matching model.
  - *Effectiveness*: over than 95% F<sub>1</sub>Measure reflecting a high accuracy of the matching model.
  
- An effective and efficient thematic event processing model that outperforms the non-thematic approach.
  - *Loose semantic coupling*: a lightweight amount of tags to describe events, around 2 – 7, and subscriptions, around 2 – 15.
  - *Efficiency*: a magnitude of 800 events/sec in the worst case.
  - *Effectiveness*: 85% F<sub>1</sub>Measure as opposed to 62% worst case for non-thematic processing.
  
- An architecture for IoT based on thingsonomies and thematic matching.
  
- A formal framework and evaluation for the proposed model based on an ensemble of semantic and top-*k* matchers in addition to a probability model for uncertainty management.
  
- A unified and native model of event enrichment is proposed along with its formalism and evaluation framework.
  
- An instantiation of the enrichment model based on dereferenceable Linked Data, spreading activation, and semantic relatedness.
  - *Loose pragmatic coupling*: high-level enrichment clauses in the subscriptions.
  - *Efficiency and effectiveness*: up to 44% F<sub>5</sub>Measure of enrichment precision and completeness, 7 times more than other instantiations of the enrichment model on average.

## 1.11 Thesis Outline

The rest of this thesis is organized as follows:

- *Chapter 2- Problem Analysis: Crossing Boundaries in Distributed Open Systems:* This chapter motivates the problem and provides a thorough investigation of the shift in the data landscape and the evolution towards the event processing paradigm. It defines the terminology used and analyses event systems based on a framework of communication models and knowledge exchange among systems boundaries. Limitations of current event processing are discussed, leading to the development of a set of requirements and research questions of this work.
- *Chapter 3- Related Work:* This chapter elaborates on the requirements and classifies the related work to the studied problem into six categories. The chapter charts the current approaches against requirements and analyses their features. It provides a gap analysis that gives guidance for the proposed approach.
- *Chapter 4- Overview of the Approximate Semantic Event Matching and Dynamic Enrichment Approach:* This chapter presents the proposed approach in terms of three main models, and the four main elements on which they stand. Each element is then mapped into a hypothesis.
- *Chapter 5- The Approximate Semantic Event Matching Model:* This chapter focuses on research question *Q1* of loose semantic coupling and the associated hypotheses *H1* and *H4*. It details the model, experiments, and results.
- *Chapter 6- The Thematic Event Matching Model:* This chapter focuses on research question *Q1* of loose semantic coupling, and the associated hypothesis *H2*. It details the model, experiments, and results.
- *Chapter 7- The Dynamic Native Event Enrichment Model:* This chapter focuses on research question *Q2* of loose pragmatic coupling, and the associated hypotheses *H3* and *H4*. It details the model, experiments, and results.
- *Chapter 8- Prototype and Use Cases:* This chapter discusses aspects of this work related to building a complete system for the Internet of Things and the event-based architecture associated with it. The COLLIDER system design and implementation building blocks are discussed along with real-world use cases.

- *Chapter 9- Conclusions and Future Work:* This chapter concludes the thesis and discusses the potential impacts along with future work.

## 1.12 Associated Publications

Various aspects of this work have been published in relevant venues as follows:

- Souleiman Hasan and Edward Curry, “Tackling Variety in Event-Based Systems,” in Proceedings of the 9th ACM international conference on Distributed Event-Based Systems - DEBS '15, 2015, pp. 256–265.
- Souleiman Hasan and Edward Curry, “Thingsonomy: Tackling Variety in Internet of Things Events,” *Internet Computing. IEEE*, vol. 19, no. 2, pp. 10–18, 2015.
- Souleiman Hasan and Edward Curry, “Approximate Semantic Matching of Events for the Internet of Things,” *ACM Transactions on Internet Technology.*, vol. 14, no. 1, pp. 1–23, Aug. 2014.
- Souleiman Hasan and Edward Curry, “Thematic Event Processing,” in Proceedings of the 15th International ACM/IFIP/USENIX Middleware Conference- Middleware '14, 2014, pp. 109–120.
- Souleiman Hasan, Sean O’Riain, and Edward Curry, “Towards Unified and Native Enrichment in Event Processing Systems,” in Proceedings of the 7th ACM international conference on Distributed Event-Based Systems - DEBS '13, 2013, p. 171.
- Souleiman Hasan, Sean O’Riain, and Edward Curry, “Approximate Semantic Matching of Heterogeneous Events,” in Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems -DEBS '12, 2012, pp. 252–263.
- Souleiman Hasan, Kalpa Gunaratna, Yongrui Qin, and Edward Curry, “Demo: Approximate Semantic Matching in the Collider Event Processing Engine,” in Proceedings of the 7th ACM International Conference on Distributed Event-Based Systems - DEBS '13, 2013, p. 337.

- Souleiman Hasan, Richard Medland, Marcus Foth, and Edward Curry, “Curbing Resource Consumption Using Team-Based Feedback,” in *Persuasive Technology*, Springer, 2013, pp. 75–86.
- Souleiman Hasan, Edward Curry, Mauricio Banduk, and Sean O’Riain, “Toward Situation Awareness for the Semantic Sensor Web: Complex Event Processing with Dynamic Linked Data Enrichment,” in *The 4th International Workshop on Semantic Sensor Networks 2011 - SSN11, a Workshop of the International Semantic Web Conference - ISWC’11*, 2011, pp. 60–72.
- Wassim Derguech, Sami Bhiri, Souleiman Hasan, and Edward Curry, “Using Formal Concept Analysis for Organizing and Discovering Sensor Capabilities,” *Computer Journal*, 2014.
- James O’Donnell, Edward Corry, Souleiman Hasan, Marcus Keane, and Edward Curry, “Building Performance Optimization Using Cross-Domain Scenario Modeling, Linked Data, and Complex Event Processing,” *Building and Environment*, vol. 62, no. 0, pp. 102–111, 2013.
- Wassim Derguech, Souleiman Hasan, Sami Bhiri, and Edward Curry, “Organizing Capabilities Using Formal Concept Analysis,” in *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2013, pp. 260–265.
- Edward Curry, Souleiman Hasan, Mark White, and Hugh Melvin, “An Environmental Chargeback for Data Center and Cloud Computing Consumers,” in *First International Workshop on Energy-Efficient Data Centers*, 2012, pp. 117–128.
- Edward Curry, James O’Donnell, Edward Corry, Souleiman Hasan, Marcus Keane, and Sean O’Riain, “Linking Building Data in the Cloud: Integrating Cross-Domain Building Data Using Linked Data,” *Advanced Engineering Informatics*, vol. 27, no. 2, pp. 206–219, 2012.
- Edward Curry, Souleiman Hasan, and Sean O’Riain, “Enterprise Energy Management Using a Linked Dataspace for Energy Intelligence,” in *Sustainable Internet and ICT for Sustainability - SustainIT ’12*, 2012, pp. 1–6.
- Edward Curry, Souleiman Hasan, Umair ul Hassan, Micah Herstand, and Sean O’Riain, “An Entity-Centric Approach to Green Information Systems,” in *The 19th European Conference on Information Systems - ECIS*, 2011, p. Paper 194.



## Chapter 2

# Problem Analysis: Crossing Boundaries in Open Distributed Systems

“We build too many walls and not enough bridges.”

— Isaac Newton

### 2.1 Introduction

Significant trends can be observed in the data landscape in the last decade. Such trends are instantiated in areas such as the Internet of Things [6] and Big Data [44]. They are characterized by an increasing number of information sources and users, an increasing level of heterogeneity, a domain-agnostic nature, and dynamism. Users within large-scale systems have a non-technical expertise and lack organization as they are highly distributed and decoupled [45]. Besides, the need to process information on a timely basis is an significant factor to leverage the potential of such large-scale environments.

Event processing systems have been proposed as a computational paradigm to handle these challenges [8]. Producers of information items fire atomic and instantaneous *events* that carry information to consumers. Producers and consumers are decoupled and they

can only interact by exchanging events. Decoupling is considered as an important factor for the scalability of distributed event-based systems [11].

Nonetheless, events are supposed to cross system boundaries between participants in event-based environments. Boundaries are syntactic, semantic, and pragmatic [46]. The more open, distributed, and heterogeneous the environment becomes, the more significant these boundaries become, especially the latter ones. Crossing semantic and pragmatic boundaries require mutual agreements between participants, which implies coupling. These agreements add to the issue of semantics and pragmatics an important social dimension. Using fixed, centralized, and top-down authoritative semantic and context models is not scalable within large-scale event processing systems [43].

Thus, an inherent trade-off between decoupling for scalability and coupling for crossing boundaries is recognized. Current decoupling dimensions in event processing systems are confined to the lower syntactic boundaries. Coupling at higher boundaries constrains the scalability of event systems within the emerging data landscape such as in the Internet of Things (IoT). The analysis in this chapter has been mainly presented in the ACM International Conference on Distributed Event-Based Systems (DEBS 2015) [47].

In Section 2.2, I motivate the problem, and in Section 2.4, recent trends in the data landscape are discussed. The evolution path towards event processing is discussed in Section 2.5. Section 2.6 provides a precise definition of the used terminology, and in Section 2.7, I discuss the traits of distribution, heterogeneity, and openness that characterize large-scale event environments. The principle of decoupling, which is central to event systems, is discussed in Section 2.8. The current event processing paradigm is projected onto communication models that cross system boundaries as a theory for event exchange in Section 2.9. Analysis of the limitations and problem with current event processing paradigm is described in Section 2.10. In Section 2.11, the set of high-level requirements tackled in this thesis are formulated along with the associated research questions and scope. The chapter is summarized in Section 2.12.

## 2.2 Motivational Scenarios

I start this chapter with two motivational scenarios that are derived from the energy management domain.

### 2.2.1 Scenario 1: Heterogeneous Energy Events

Alice is a sustainability officer in a large corporate in the electronics industry. The organization has many offices and facilities all over the city. Alice’s job is to ensure that the company sticks to its Corporate Social Responsibility (CSR) programs such as saving energy and lowering its overall CO<sub>2</sub> emissions. Most of the company’s buildings are equipped with sensors for energy consumption, temperature, and other environmental parameters.

Alice wants to set up a rule to notify her when an excessive energy consumption situation in the public spaces of the buildings is detected. The intended alert should fire when *energy consumption from heating public halls of the buildings increases*. Such a scenario may happen as employees tend to open the windows if it is warm despite the fact that the heater is still turned on. This rule can be expressed using an Event Processing Language (EPL) such as Esper’s language [48] as follows:

```
pattern [ every a=BuildingsEvents(a.type= ‘heater energy consumption increased’
           and a.location=‘public hall’)]
```

While the sources of the required information are available from the buildings IoT nodes, the semantics of the events differ from one building to another. That is due to different manufacturers of the sensors. For instance, events contain terms such as *‘energy consumption’*, *‘energy usage’*, and *‘power consumption’* to refer to the same thing.

Alice is not interested in an exact number of such behavioural patterns but rather in a rough estimate that helps her take an action. Alice asks the IT department to realize the intended detection scenario. The IT department reports that the scenario requires a large set of rules such as the one above to be deployed on an existing event processing engine with all possible variations of semantics to cover the semantic heterogeneity that exists. These rules take time to be defined and will also need to be updated when the environment or the requirements change (such as adding new sensors).

### 2.2.2 Scenario 2: Incomplete Energy Events

For a sustainability officer Dave to do his job, various energy-related sensors are instrumented, so events flow into a middleware. Events in such a scenario are encapsulated

with minimal information recording, for instance, a device's name and the amount of energy used. An example attribute-value event describing the energy consumption of a heater is as follows:

```
{(type: energy consumption),  
  (device: heater1),  
  (consumption: high)}
```

Non-technical users such as Dave tend to include higher level and business concepts and checks in their subscriptions to events. Examples of these are the *'room'* or the *'floor'* where the event was originated, or the *'business unit'* or *'project'* with which the device is associated. One example subscription is as follows:

```
{(type= energy consumption)  
  and (floor= second floor)  
  and (consumption= high)}
```

The events do not contain information about the *'floor'* to answer the subscription. Thus, to meet the information requirement for this subscription, additional information sources in the enterprise such as data about the building would need to be exploited. Dedicated software agents need to be developed to enrich events with sufficient information. A large number of subscriptions may require dedicated enrichment agents. As a result, enrichment routines can become a burden to develop and maintain.

## 2.3 Challenges

The above motivational scenarios reflect real-world situations with substantial challenges. The first challenge, reflected in Scenario 1, is tackling semantic variety in an event environment with a low cost on the users' side. The second challenge, reflected in Scenario 2, is ensuring the information completeness of events for processing with a low cost on the users' side. These challenges become harder in light of new trends that are taking place in the data landscape and affecting the assumptions made in computational paradigms.

## 2.4 Significant Trends in the Data Landscape

In recent years, there has been a tremendous increase in information sources and volume. The Organization for Economic Co-operation and Development (OECD) estimates that there will be about 50 billion devices connected to the Internet by 2020 [1]. This leads to challenges for information processing solutions as the volume, velocity, and variety of data increase. Environments such as smart cities [2], smart buildings [4], and cyber-physical energy systems [5] are active topics of research throughout the last decade. A key technology enabler for these areas is represented by the Internet of Things [6].

### 2.4.1 Internet of Things

The Internet of Things (IoT) aims to connect physical objects, or things, to the Internet and enable a plethora of applications such as assisted driving and smart cities. From a high-level architectural perspective IoT can be divided into three tiers [6]:

1. **Sensing and communication** technologies form the basic infrastructure for IoT to map the world of things into the world of computationally processable information. Radio-Frequency Identification (RFID) plays a critical role within this tier where RFID tags are attached to real-world things and RFID readers are responsible for instrumenting their information into the Internet. Communication and networking standards such as the IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) and the CoAP protocols [6] serve this layer of IoT.
2. **Middleware layer** abstracts application developers and users from IoT infrastructure details. Technologies here include the Service-Oriented Architectures (SOA) [6], the Message-Oriented Middleware (MOM), or hybrid service computing with event patterns [49]. Event processing systems are a more generic version of MOM which supports functionalities such as early filtering of events, spatio-temporal correlation, sequencing, event enrichment, and complex event processing.
3. **Application layer** builds upon the middleware to provide direct domain-specific benefits to users. IoT promises new applications in domains including transportation and logistics, healthcare, smart environments, personal and social media, and more advanced applications such as robo-taxis and virtual reality.

Actual deployments have started to appear such as the SmartSantander smart city [50], the Oulu smart city [51], Friedrichshafen smart city [52], CitySense urban-scale wireless networking testbed [53], and METRO Future Store for RFID-based retail [54].

Significant efforts in the area of IoT come from communication and networking communities, e.g. the Internet Engineering Task Force (IETF) [6]. However, there is a growing realization that the challenges of the IoT will be more prevalent at the data level [7] including data collection, management, and analytics. At this level, IoT can be linked with the area of Big Data.

### 2.4.2 Big Data

The real potential of Big Data is that it will improve decision making in business and society where it is not based on data and facts, more and more with a data-based process [55] as put by Jagadish et al. [44]:

“In a broad range of application areas, data is being collected at an unprecedented scale. Decisions that previously were based on guesswork, or on painstakingly handcrafted models of reality, can now be made using data-driven mathematical models. Such Big Data analysis now drives nearly every aspect of society, including mobile services, retail, manufacturing, financial services, life sciences, and physical sciences.”

Big Data should not be understood just in terms of data volume. In fact, one of the most commonly used analysis recognize three dimensions of the phenomenon [56]:

- *Volume*: refers to the sheer size of the data.
- *Variety*: refers to the heterogeneity of data representation, types, and semantic interpretation.
- *Velocity*: refers to the rate of incoming data, and the need for low latency to act upon the data.

Jagadish et al. [44] recognize five phases in the Big Data lifecycle being: (1) data acquisition, (2) information extraction and cleaning, (3) data integration, aggregation,

and representation, (4) modelling and analysis, and (5) interpretation. Jagadish et al. [44] argue that much focus has been given to data analysis, with less focus on the other phases. I do agree and think that phases such as integration are of great importance to have a properly complete grounding for analysis to take place and benefit from all possible aspects related to data on the phenomenon of interest.

### **2.4.3 Common Characteristics**

The observed trends of IoT and Big Data signify a considerable shift in the characteristics of information production, communication, and consumption paradigms. A set of common characteristics can be identified as follows:

#### **2.4.3.1 Number of Information Sources**

The number of sources that can create data has been increasing significantly. For example, the International Telecommunication Union (ITU), a United Nations organization, reports that the number of worldwide mobile subscriptions has increased from 738 million in 2000 to 7 billion in 2015 [57], i.e. around 800% increase in 15 years. The percentage of basic mobile devices is shrinking with the use of more smartphones [58] with advanced capabilities such as Global Positioning System (GPS) information tracking and sensors on-phone.

#### **2.4.3.2 Data Heterogeneity**

Heterogeneity appears in various forms including different types of networks, protocols, devices capabilities, data formats, and representation [6]. This phenomenon is also referred to as variety [56]. Heterogeneity in data representation, or semantic heterogeneity, in IoT follows partially from the number of devices and manufacturers of these devices.

For instance, the Semantic Web [59] is a worldwide initiative to provide structured data on the Web that started around 2001. It uses Semantic Web technologies such as the Resource Description Framework (RDF) [60], the RDF Schema (RDFS) [61], and the Web Ontology Language (OWL) [62]. Falcons [63], a search engine for the Semantic Web, could discover in 2008 about 4,000 ontologies on the Web [64], while this number

has increased to more than 6,400 in 2015 [65], i.e. more than 50% increase in 7 years. By analogy, I suggest that a similar trend will be seen in IoT with more data representations used for *things* especially in domains such as smart cities.

#### **2.4.3.3 Number of Users**

The number of users who have access to data has been drastically increasing. On the Internet, for example, the ITU reports an increase in the number of individuals using the Internet from 400 million in 2000, to 3.2 billion in 2015 [57], i.e. around 700% increase in 15 years.

#### **2.4.3.4 Technical Knowledge of Users**

Among a large number of users who have access to the produced data, very few could be expected to have a proper technical knowledge to do so. For example, according to the US National Science Foundation, National Center for Science and Engineering Statistics, about 135,000 people graduated with a bachelor's degree in computer, mathematics, or related disciplines in 2010 [66]. That is about 3% of people in that age bracket [67]. By projecting this ratio into the number of users with access to data, I conclude that billions of users lack the technical knowledge to access and interpret data if they are not supported by a computational paradigm to overcome this lack of technical expertise.

#### **2.4.3.5 Organization and Coordination of Users**

In large-scale environments of data production and consumption, users do not follow any form of organization, but they are rather autonomous and decoupled. This fact has been acknowledged and leveraged in some cases as in *crowdsourcing*, which harnesses the knowledge of widely decoupled users to provide services or content [45, 68], as in *Wikipedia* for instance [69]. Studies of the demographics of crowdsourcers reveal a global diversity and geographical distributions of crowdsourcers [70]. I suggest that similar characteristics can be assumed in IoT settings where users will be decoupled and non-organized, which affects the assumptions of any computational paradigm for IoT.

#### 2.4.3.6 Dynamism

The architecture for platforms such as IoT has been continuously becoming more dynamic, where data producers and consumers continuously join and leave the environment. Vermesan et al. call this phenomenon *fluid* systems that are continuously changing and adapting and put it as follows [71]:

“In IoT systems ... it is very common to have nodes that join and leave the network spontaneously.”

This dynamic nature puts constraints on the assumptions that can be made within a computational paradigm about having full understanding or control over the environment.

#### 2.4.3.7 Domain

The domain of data in large-scale systems is open rather than specific. For example, anticipated IoT applications belong to various domains and even some of them such as smart cities can be domain-agnostic [72]. While individual users could have interests influenced by a specific domain, data itself comes from many sources with different contexts. Thus, I argue that data representation should be domain-agnostic in general, but easily adaptable to specific domains when users have this interest.

#### 2.4.3.8 Timeliness

This is called velocity too [56], and means that due to the high volume of data and the high number of data sources available in Big Data settings such as IoT, it becomes crucial to filter important data items as early as possible [44]. It could be economically expensive to store raw data [44] and thus computational paradigms for these scales should employ early filtering and detection, as well as indexing to meet the timeliness requirement.

### 2.4.3.9 Information Completeness

Data at large scales coming from distributed sources could be erroneous, inconsistent, and incomplete for some users' requirements. Jagadish et al. [44] state that:

“Big Data increasingly includes information provided by increasingly diverse sources, of varying reliability. Uncertainty, errors, and missing values are endemic, and must be managed.”

Given this shift in the data landscape, there has been an evolution in the information processing landscape to meet these new challenges.

## 2.5 The Event Processing Paradigm

Throughout the end of the twentieth century and the first decade of the twenty first, there has been a realization among researchers and practitioners that a new class of information processing systems is needed. The new class, the event processing paradigm, has been motivated by a plethora of distributed applications that require on-the-fly and low-latency processing of information items. Example applications include environmental monitoring from sensors [73], stock market analysis for emerging trend identification [74], RFID-based anomaly detection in inventories [75], and security systems such as intrusion detection [76].

Hinze et al. [9] analyse various applications that could justify the need of the new paradigm of event processing. They abstract the features required in such applications and develop a framework that correlate them to their application classes. Features include for example “spatio-temporal correlation,” “event sequencing,” “out of order events,” “homogeneous aggregation,” “derived events,” “event enrichment,” “outlier handling,” “early filtering,” “volume,” “security,” “mobility of event source,” “mobility of event subscriber,” etc. Cugola and Margara [8] complement this picture to justify a new paradigm stating that: “The concepts of timeliness and flow processing are crucial for justifying the need for a new class of systems.”

The concept of *timeliness* has been expressed in the literature using various terms such as *low latency* [77, 78], *high throughput* [78, 79], *low delay* [80], *volume* [9], and *real-time*

*processing* [81, 82]. All of these terms, except real-time processing, can be classified under the umbrella of *fast* computing. This term means essentially that the system is *efficient* in processing information items in a way that the ratio  $\frac{\text{value}}{\text{time}}$  is maximized.

On the other hand, real-time processing includes the other concept of executing the information processing task within a time constraint, called a *deadline* [83]. Timeliness, as described by Cugola and Margara [8], is much more similar to the concept of fast computing. Technically, it can be measured by the related concepts of latency and throughput. Latency is defined in this work as the total time required to process an information item starting from its arrival to the processing agent until its completion. Throughput is the number of information items completely processed within a time unit.

The concept of *flow processing* refers to the processing of information items without the need to store them. A very relevant concept to both timeliness and flow processing is the processing of information items as they become available. Thus, *timeliness* whenever used in this thesis means low latency, high throughput, and processing as soon as the information items are available. Real-time processing if used should also be taken to have this meaning. Stonebraker et al. [81] call this an active model of processing:

“Ideally the system should also use an active (i.e., non-polling) processing model.”

As an example that encompasses both aspects, consider a system that needs to generate an alert for excessive energy usage in a room. A situation that could occur when a projector and lights are turned on in an empty room. Assume that the room is equipped with sensors to monitor these phenomena. On the one hand, the alert shall be fired once the situation is detected so an action is taken and no more energy is wasted. On the other hand, streams from the sensors do not need to be stored if they are non-relevant to the situation of interest.

### 2.5.1 Evolution Towards Event Processing

The event processing paradigm has evolved through the work of several communities in whose artefacts elements of the paradigm can be detected. In the following sections, I consolidate an evolution path based on works by Cugola and Margara [8], Hinze et al. [9], Etzion [10, p. 16–21], Eugster et al. [11], and Mühl et al. [12, p. 7–8].

### 2.5.1.1 Active Databases

Active databases started to appear during the last two decades of the twentieth century [13]. The term *active* is put in contrast to the term *passive* that is assumed to exist in database systems before the appearance of active databases. Paton and Díaz define a passive mode as:

“Traditional database management systems (DBMSs) are passive in the sense that commands are executed by the database (e.g., query, update, delete) as and when requested by the user or application program.”

The challenge when applications are developed on top of passive databases is that it is sometimes needed to detect a situation in the database as soon as it happens so that an action can be taken upon it. For example, a database system stores information about a bus transportation company and is available to booking applications from multiple travel agencies. If the booked seats of a particular route exceed some threshold, the system administrator shall notify the operations department to provide more buses or consider double-deck buses in advance. With passive databases, this behaviour must be incorporated in every booking application to query the database, check the condition after each booking, and notify the system administrator. This causes a redundancy of business logic among distributed applications and affects maintenance. Active databases move the reactive behaviour from the application layer to the database layer.

Three parts of an active behaviour are recognized: *event*, *condition*, and *action* [13]. These parts are encoded in active databases using *rules* that are called in this context Event Condition Action (ECA) rules. HiPAC [84, 85] has been the first system to propose the ECA model along with an architecture of the system [8]. Ode [86, 87] is an object-oriented active database system that uses triggers and supports detection of a set of events to fire the trigger. SAMOS [88] is similar to Ode and can also consume external events. Snoop [89] defines an active database rule language independent from the underlying data model, making it suitable for relational or object-oriented models. As active databases are centred around a persistent storage as opposed to processing in-flow, their performance degrades when the number of updates or the number of active rules become very high.

### 2.5.1.2 Reactive Middleware

In a distributed heterogeneous application network of different operation systems, applications need a homogeneous view, so developers are abstracted from low-level issues of distributions and focus on the application logic [12, p. 2]. Within the context of static networks, middleware systems view data and services stationary in objects of databases. This fixed topology allows an interaction model of request/reply to and from the stationary nodes. That has given rise to the Remote Procedure Call (RPC) paradigm [16] and its derivative client/server architecture. Many middleware systems have been developed such as the Object Management Group (OMG)'s Common Object Request Broker Architecture (CORBA) [15] developed for object-oriented software distribution, and the Java Platform Enterprise Edition (J2EE) [14].

When the stationary assumption of networked applications is not valid, the request/reply paradigm is limited as it imposes a tight coupling between the communicating parties [11]. Clients use more resources to check for data integrity, and more-than-necessary polling of remote data stores causes unnecessary waste of resources [12, p. 2]. Thus, asynchronous extensions have been added to the existing middleware systems [9] such as a J2EE extension [90] and the Real-time CORBA Event Service [91].

### 2.5.1.3 Event-based Software Engineering

Complex software systems consist of many integrated components that collaborate to achieve the overall system's goal. Consider for example an object-oriented architecture, the classical way for components to interact with each other is by *explicit invocation* where objects explicitly have references to each other. Each object then explicitly calls methods contained in other objects.

An alternative to this mechanism is called *implicit invocation* [17] where a component fires events that announce some actions, which have occurred. Other components, which are interested in particular actions, register their interest and, as a result, they get notified whenever such events happen so they can react properly. This principle is widely used now in Enterprise Application Integration (EAI) [18] and software design patterns such as the observer pattern [92, p. 293–304]. Implicit invocation has been employed

in various areas such as Aspect-Oriented Programming (AOP) [19] and Graphical User Interfaces (GUIs) [20].

#### 2.5.1.4 Message-Oriented Middleware

The Internet is based on a set of protocols and primarily the Internet Protocol (IP). According to the IP-based architecture, nodes on the Internet can communicate via a coupled, synchronous, and end-to-end interaction scheme [11]. Nonetheless, the Internet has become a platform for distributed applications that exchange information in a way that the location and behaviour of these applications are dynamic. Thus, a decoupled interaction scheme has become crucial to the development of large-scale applications as stated by Eugster et al. [11]:

“Individual *point-to-point* and *synchronous* communications lead to rigid and static applications, and make the development of dynamic large-scale applications cumbersome ”

A middleware layer can serve as the distribution platform over the IP-based Internet in a transparent manner to the dynamic higher level applications. Eugster et al. [11] give the coupling dimensions of the interaction scheme a great importance to scalability, as discussed in Section 2.8. They recognize three dimensions of decoupling: *space*, *time*, and *synchronization* that are concerned respectively with addresses, activity time, and blocking.

Communication paradigms such as remote procedure call [93] and shared spaces [94] are coupled on one or more of these dimensions. Thus, the publish/subscribe paradigm evolved to overcome this issue [11]. In publish/subscribe the publishers send messages to the middleware, and the consumers subscribe to particular messages of interest (also called events in this context). The way subscribers can express their interests vary and thus various publish/subscribe schemes exist:

- *Topic-Based Publish/Subscribe* suggests that producers publish their messages on named logical channels. The names of the channels are derived typically from the content of the messages to form a logical grouping of them. Topics may have a

hierarchical naming scheme similar to a Uniform Resource Identified (URI) and can use wildcards. Early examples are iBus [95] and TIBCO Rendezvous [96].

- *Content-Based Publish/Subscribe* uses the actual content of event messages as the routing mechanism and the matching basis between events and consumers interests [97]. Consumers register their subscriptions in terms of filters that are used for matching. Filters can comply with a tuple model, an attribute-value model, a hierarchical model for XML, or a graph model [12, 26]. Early examples of content-based publish/subscribe approaches include Siena [22], Elvin [98], Jedi [99], and the Java Messaging Service (JMS) [100].
- *Type-Based Publish/Subscribe* depends on the use of event types from the type hierarchy of programming languages to address that events have structure in common as well as the topic name [23].
- *Concept-Based Publish/Subscribe* defines a mediator layer based on an ontology for resolving data heterogeneity between events and subscriptions [40]. This topic will be tackled in further detail throughout this thesis.

I argue that the publish/subscribe paradigm be the main cornerstone to the modern event processing paradigm. That is due to the view of a message as an individual data item that may have temporal semantics to qualify for an event. Thus, discussions of the current principles and fundamental issues of event processing, especially decoupling, could be based on previous work in the publish/subscribe research.

### 2.5.1.5 Wireless Sensor Networks

Wireless Sensor Networks (WSNs) advanced the traditional one sensor architecture to build a large ad-hoc network of communicating sensor nodes that have a better sensing coverage to some stimuli. WSNs have been enabled by advancements in wireless communication and low production costs for sensor devices that hold sensing, data processing, and communication elements [101]. Akyildiz et al. [101] recognize several design factors that have been addressed by researchers including “fault tolerance,” “scalability,” “production costs,” “hardware constraints,” “network topology,” “environment,” “transmission media,” and “power consumption.” Power consumption forms one of the

main challenges which requires power efficient routing protocols to move data sensed by the nodes to interested *sink* nodes [9].

I argue that each node in a WSNs can be compared to an event processing engine. However, abilities of sensor nodes to process data are limited due to power and hardware constraints. Although the scale of sensor networks can reach a magnitude of thousands of nodes, the variety in the sensed data is still manageable when compared to that which exists in data generated in enterprise applications or humans on the Web. They also form ad-hoc solutions as opposed to a generic event processing paradigm where events can come from sensors or other sources [10, p. 16].

### 2.5.1.6 Data Stream Management Systems

Active databases do not scale under high rates of database updates or large numbers of rules as the discussion in Section 2.5.1.1. Thus, the database community developed Data Stream Management Systems (DSMSs) to cope with high rates of data updates. Streams form the basic concepts in a DSMS as opposed to tables in conventional databases. A *stream* is an unbounded table of tuples. Tuples have no assumption on their arrival order and typically have no temporal semantics [8].

DSMSs adopt an interaction paradigm based on *continuous queries* where users register a set of queries with the DSMS [102]. DSMS is responsible for evaluating the queries periodically and notifying users accordingly. This model is a Database-Active Human-Passive (DAHP) model as opposed to the Human-Active Database-Passive (HADP) model exemplified by conventional database systems [103]. The time constraints of DSMS applications, such as real-time monitoring, prevents the storage for delayed processing and requires an in-flow processing model. *Windows* are associated with DSMSs as operators to limit the stream from which input data items are considered for processing.

Example DSMS engines include: TelegraphCQ [104], NiagaraCQ [105], OpenCQ [106], Tribeca [107], CQL/Stream [35], Aurora/Borealis [103], Gigascope [108], and Stream Mill [109]. Commercial DSMSs also exist such as Sybase Coral8 Engine [110], SteamBase [111], and IBM System S [112].

DSMS extend the event-based model and the reactive behaviour with *transformation operators*, that is operators that consume one or more streams and produce a transformed

stream. Cugola and Margara [8] analyse DSMSs and find basic commonalities that distinguish this category of systems:

- *Data items are homogeneous in a stream:* A stream is similar to a conventional relational table of data tuples that share the same set of properties such as in TelegraphCQ [104], NiagaraCQ [105], OpenCQ [106], Tribeca [107], CQL/Stream [35], Aurora/Borealis [103], Gigascope [108], and Stream Mill [109].
- *Data items do not typically have temporal or causal semantics:* There is no total order assumed for the data items and thus *sequence* or *caused by* operators, for example, cannot be found in pure instances of such systems, e.g. [35, 103, 105–107, 109]. However, some DSMSs such as TelegraphCQ [104] and Gigascope [108] have some form of temporal or causal order.
- *Languages are typically of a transformation nature:* Examples can be found in [35, 103–109]. Consequently, a rule processes input streams by filtering, joining, or aggregation and produces an output stream. Detection languages of clearly recognized condition and action parts are not the dominant type.
- *Languages can be declarative or imperative:* SQL-like declarative languages can be found in [35, 104–106, 108, 109]. Imperative languages as in [103, 107] depend on plans of operators that can be graphically expressed.

I argue that the critical and most important feature that can distinguish DSMSs is the handling of streams as the most atomic information item. Although streams contain smaller items, those are homogeneous and lack total temporal or causal ordering. This fact has deep impacts on other features of DSMS such as the type of operators that are found in DSMSs to deal with streams as inputs and streams as outputs. Events, as I define and use in this thesis, are the information items when the system can handle them separately not necessarily in a stream. That is the view adopted in complex event processing systems as discussed in Section 2.5.1.7 and Section 2.6.

### 2.5.1.7 Complex Event Processing

Data stream management systems do not associate particular semantics to their data items. They serve as generic systems that process generic data items similarly to the

case of conventional database systems. On the other hand, event processing systems associate specific semantics to their data items. They are computer objects that represent notifications of actual or virtual happenings as gathered by sources. That is, events are the most atomic data items in event processing systems as opposed to streams in DSMSs.

Publish/subscribe systems [11] form the basis for event processing systems, with processing focused on filtering and routing. Typical publish/subscribe systems process one message, i.e. event, at a time. Events are matched against subscriptions without looking at what previous events happened before in the history [113]. Thus, the publish/subscribe systems have been extended with the notion of matching multiple events against a single subscription, or rule. This set of events is called a *pattern* and they signify a *composite* event [114], a *complex* event [115], or a *situation* of events [116].

A Complex Event Processing (CEP) engine thus emphasizes the matching of event patterns specifically with ordering conditions such as temporal sequencing and causal relationships. They take the name from Luckham's book "The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems" [115]. For instance, a complex event engine fires an *excessive energy consumption event if an occupancy sensor detected that a room is empty and a light sensor detected that the lights are on for at least 10 minutes after the last person left the room.*

Examples of CEP engines include: Rapide [117], GEM [118], Padres [114], DistCED [119], Cayuga [120], NextCEP [121], PB-CED [122], Raced [123], Amit [116], Sase [124], Sase+ [125], Peex [126], and TESLA/T-Rex [127, 128]. Commercial systems do also exist such as SAP Event Stream Processor [129], Oracle Event Processing [130], Esper [48], TIBCO Business Events [131], and IBM WebSphere Business Events [132].

Use cases for CEP engines such as environmental and business process monitoring as well as actuation systems, usually require CEP engines to interact with a large set of distributed components such as events sources and consumers. Thus, the research agenda of complex event processing has been impacted by distributed views of CEP engines (or agents), over a network. CEP research is typically concerned with relevant topics such as bandwidth, throughput, latency, the distribution architecture, forwarding schemes, and the placement of various pieces of processing logic over the network.

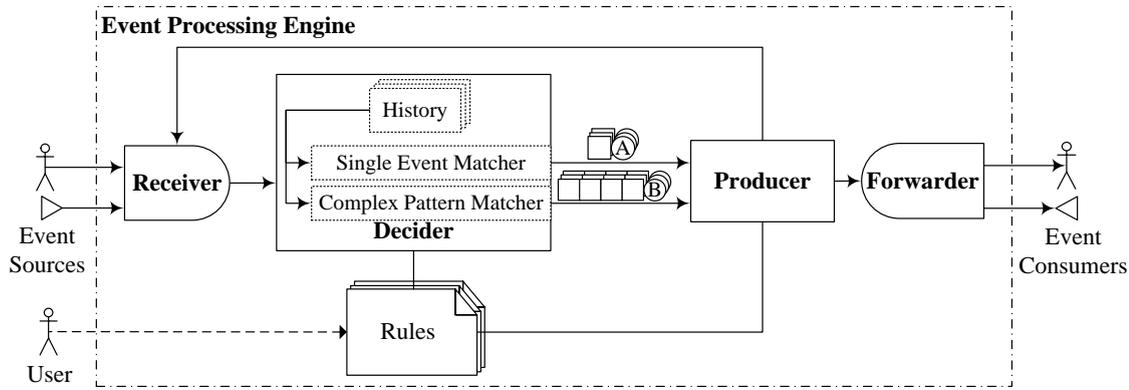
Cugola and Margara [8] present the main common aspects that distinguish CEP engines as follows:

- *Data items are heterogeneous:* Events are not supposed to fall into one data model within a stream and multiple types of events can feed into one input of the CEP engine as in Rapide [117], GEM [118], Padres [114], DistCED [119], PB-CED [122], Raced [123], Amit [116], Sase [124], Sase+ [125], Peex [126], TESLA/T-Rex [127, 128], and TIBCO Business Events [131].
- *Data items typically have temporal and/or causal semantics:* That is a total order is assumed for the data items and thus *sequence* or *caused by* operators can be found in most instances of these systems as in Rapide [117], GEM [118], Padres [114], DistCED [119], PB-CED [122], Raced [123], Amit [116], Sase [124], Sase+ [125], Peex [126], SAP Event Stream Processor [129], Oracle Event Processing [130], Esper [48], TIBCO Business Events [131], and IBM WebSphere Business Events [132].
- *Languages are of a detection nature:* Clearly distinguishable condition and action parts can be found in these languages as in Rapide [117], GEM [118], Padres [114], DistCED [119], PB-CED [122], Raced [123], Amit [116], Sase [124], Sase+ [125], Peex [126], TESLA/T-Rex [127, 128], and TIBCO Business Events [131].

Event processing within this scope is understood to deal with atomic items that are records that could exist independently of others and not necessarily in a stream. Such events have partial/total temporal or causal order that allows CEP authors to write rules with complex patterns that leverage such relationships to detect complex events. The result of detection is a derived, and possibly abstract, event that is produced by the CEP engine to join the overall set of events and can feed back into the input or be forwarded to the user for consumption.

### 2.5.2 The Information Flow Processing Domain

There have been multiple works for consolidating the picture of event processing systems and providing a unified version over the communities they emerged from. Descriptions

FIGURE 2.1: The IFP functional model <sup>1</sup>

and classifications of active database systems, for instance, have been studied by McCarthy and Dayal [85], and Paton and Díaz [13]. Analysis of data stream management systems have been done by Babcock et al. [133], Golab and Özsu [134], and Babu and Widom [102]. Similar surveys and analysis have been conducted for publish/subscribe systems by Eugster et al. [11] and Mühl et al. [12], as well as work for event processing systems by Luckham [115], Etzion and Niblett [10], Hinze et al. [9], and the glossary by the Event Processing Technical Society (EPTS) [21].

I think that among those studies, the analytical survey done by Cugola and Margara [8] is the most comprehensive for the following reasons: (1) they provide a wide coverage of relevant paradigms, (2) they define an umbrella paradigm of the emerging systems and call it the Information Flow Processing (IFP) Domain, and (3) they build their analysis on several models in order to study the IFP domain.

Figure 2.1 presents an elaboration of Cugola and Margara’s first model, **functional model**, and shows the main functionalities of an IFP engine. Event *Sources* are human, software, or hardware agents that create events. Events propagate get received by the *Receiver* that sends the events to the *Decider*. The *Decider* is responsible for the detection of conditions or patterns that hold in single events or a set of events according to the condition parts of the *Rules* registered in the engine. When a condition is detected in the *Decider*, the participating events that caused the detection are propagated along with the condition to the *Producer*, refer to *A* and *B* at the centre of Figure 2.1.

The *Producer* generates an event as dictated by the action parts of the *Rules* whose conditions or patterns are detected with possibly binding of placeholders with actual values

<sup>1</sup>Adapted from Cugola and Margara’s functional model [8]

from events. The generated events may feed back to the *Receiver* and/or propagate to the *Forwarder* that sends them to the external event *Consumers*, which may be human agents, software applications such as user interfaces, or hardware agents. The *Decider* keeps in its working memory a *History* of events that may be eligible to trigger a detection. The *Single Event Matcher* detects only single events that match some filtering conditions while the *Complex Pattern Matcher* detects a pattern of events such as the sequence of two or more events that have passed single event matching.

The second model for analysing IFP engines is the **processing model**. When an event arrives at the engine, a detection-production cycle occurs. Its output is determined by the decider, rules, history, and the knowledge base. The processing model is concerned with three important factors: the *selection*, *consumption*, and *load shedding* policies.

Selection policy can be single, multiple, or programmable to determine how many times a rule is fired upon an incoming event. Consumption policy can be zero, selected, or programmable to determine in how many detection-production cycles an incoming event can participate. Load shedding includes techniques to drop some events where their incoming rate is higher than the processing capacity of the engine.

The third model to classify IFP engines is the **deployment model** being centralized or distributed. The fourth model for IFP engines is the **interaction model**, which can be push-based where the first component initiates the interaction, or conversely pull-based. The fifth model to analyse IFP engines is the **data model**: events being homogeneous or heterogeneous, the format being: tuples, records, objects, hierarchical like XML, or graphs, and whether the engine can deal with uncertainty.

The sixth model is the **time model**. It defines whether the engine can establish a happened-before or causal, partial, or total order between events. The seventh model is the **rule model** being transforming or detecting rules. The eighth and final model to classify engines is the **language model** being either declarative or imperative. Languages can also be studied according to the available operators.

## 2.6 Terminology and Definitions

The central concepts of the event processing paradigm which are used within this work are: events, producers and consumers, subscriptions and rules, and event engines.

### 2.6.1 Event

The first aspect to consider about an event is the fact that it reflects some activity in the real world or some virtual realm. For example, ‘*a person left the room 10 minutes ago*’ is an activity that happened in a real world building at a recent point in time. This meaning is typically found in English dictionaries. Merriam-Webster online dictionary [135] defines an event as:

“Something (especially something important or notable) that happens.”

Oxford British and World English online dictionary [136] defines it as:

“A thing that happens or takes place, especially one of importance.”

Within the event processing research and technical community, this meaning has been captured by Luckham [115, p. 88–90] as the *significance* of the event, by Mühl et al. [12, p. 11], Etzion and Niblett [10, p. 4], and the EPTS glossary [21] as the *event*.

The other important aspect is related to the information object or message that represents the first meaning in a computing system. For example, a Java object of type ‘*PersonLeftEvent*’ and suitable data fields can represent the real world happening that ‘*a person left the room 10 minutes ago*’ in an Esper deployment. This meaning has been captured by Luckham [115, p. 88–90] as the event’s *form*, by Etzion and Niblett [10, p. 4] as the *event*, by Mühl et al. [12, p. 11] as *notification*, by Cugola and Margara [8] as an *information item*, and in the EPTS glossary [21] as the *event*, *event object*, *event message*, and *event tuple*.

I adopt in this work the term *event* to refer to both meanings, including the meaning of information items of the generic IFP domain (Section 2.5.2). Nonetheless, the term event as used throughout this work refers mostly to the meaning of the digital object

that represents the real world occurrence. Other terms, such as *message*, are also used to refer to this meaning. Thus, I herein adopt the EPTS glossary [21] definition as:

**Definition 2.1** (Event). “An object that represents, encodes, or records an event, generally for the purpose of computer processing.”

Other terms that are usually associated with events are defined below:

**Definition 2.2** (Event Stream). A totally ordered set of homogeneous events. Ordering refers to time, causality, or aggregation, also called *event relativity* by Luckham [115, p. 88–96].

**Definition 2.3** (Event Cloud). “A partially ordered set of events.” [21]

**Definition 2.4** (Single Event, or Simple Event). “An event that is not viewed as summarizing, representing, or denoting a set of other events.” [21]

**Definition 2.5** (Complex event). “An event that summarizes, represents, or denotes a set of other events.” [21]

**Definition 2.6** (Event Type, Event Class, Event Definition, or Event Schema)). “A class of event objects.” [21]

I argue that time be not a very central concept in the paradigm as it is conceived. In fact, time has two faces that occur in literature. The first one, *timeliness*, is about in-flow processing and latency as discussed in Section 2.5 which is a system feature rather than an event feature. The other one, *timestamp*, is about the point in time when the event happens or is detected. This latter one makes sense mainly when events are projected together, so it is about the relationship and order between events, or event relativity in Luckham’s terms [115, p. 88–96]. Thus, I argue that it be an extrinsic property of events, rather than an intrinsic property.

### 2.6.2 Producers and Consumers

Events flow within networks of event agents that produce, process, or react to events [115, p. 176–177]. This entails the following definitions:

**Definition 2.7** (Event Producer, or Event Source). “An event processing agent that sends events.” [21]

**Definition 2.8** (Event Consumer, or Event Sink). “An event processing agent that receives events.” [21]

Let us take an example from software engineering. A software component that plays the role of a producer is a “*self-focused*” component as called by Mühl et al. [12, p. 12] as it observes its own internal state. The event is then a change in that state. The way components are programmed to define what and when to publish and event is the topic of areas such as debugging [137], reflection [138], and Aspect-Oriented Programming (AOP) [19]. Event consumers react to events delivered to them via the network. An agent can play both roles at the same time.

### 2.6.3 Subscriptions and Rules

Users exist in event processing models as shown in Figure 2.1. Subscriptions and rules are entities that allow users to express their interest in an event or situation and possibly react to it. That entails the following definitions:

**Definition 2.9** (Subscription, or Event Template). A filter that describes an event of interest to the user. A template is a subscription which has some parameters as variables.

**Definition 2.10** (Rule). “A prescribed method for processing events.” [21]

Subscriptions are typically filters, while rules usually have explicit parts for the event(s), and an action to be taken when the events are detected. Subscriptions and rules conform to a language model which is dependent on the underlying event model as described by the rule model in Section 2.5.2. The detection part of subscriptions and rules may refer to single events, or to a pattern of events.

**Definition 2.11** (Event Pattern). “A template containing event templates, relational operators and variables. An event pattern can match sets of related events by replacing variables with values.” [21]

The action part in complex event processing rules are typically events that are generated upon detection.

**Definition 2.12** (Derived Event). “An event that is generated as a result of applying a method or process to one or more other events.” [21]

### 2.6.4 Event Processing Engine

Event engines are central to the paradigm of event processing.

**Definition 2.13** (Event Processing Engine, or Agent). “A software module that processes events.” [21]

The term processing is understood within the functional model of Section 2.5.2. It consists of receiving events, management of rules, deciding on a detection condition, producing derived events, and forwarding events.

## 2.7 Traits of Large-Scale Event Processing

Herein, I analyse three main traits of event processing systems from a technical perspective. I suggest that these traits are fundamental characteristics for event systems when dealing with large-scale environments.

### 2.7.1 Distribution

Distribution can be understood from two complementary aspects. This first aspect of distribution is the placement of processing workloads on different nodes and as a result makes use of parallel computing. This can be done on a cluster or a network of connected processing units.

The second aspect is that environments at large-scales are inherently distributed with event production and consumption happening at distributed components. As put by Cugola and Margara [8]:

“This feature also has an impact on the architecture of CEP engines. In fact, these tools often have to interact with a large number of distributed and heterogeneous information sources and sinks which observe the external world and operate on it.”

Thus, even when dealing with a centralized event processing engine, considerations of the innate nature of distribution of the environment of event producers and consumers

shall be taken into account. For this purpose, I define a distributed event processing environment as follows:

**Definition 2.14** (Distributed Event Processing Environment). A deployment of event processing network where event producers, consumers, and processing engines may be distributed across multiple physical networks, computers, and software artefacts [21]. The functionality of event processing engines does not need to be distributed or parallelized.

### 2.7.2 Heterogeneity

Heterogeneity occurs at large scales in the form of differences in hardware components, protocols, operating systems, middleware, and data [24]. This work is concerned with data heterogeneity in event systems as described by Mühl et al. [12]:

“Syntax and semantics of notifications are likely to vary and there are inevitably different data models in use.”

This work deals with semantic heterogeneity. I start by defining semantics first. I draw here on Gärdenfors [139, p. 151]:

**Definition 2.15** (Semantics). Semantics can be defined as the mapping  $\mathcal{S}$  between symbolic words and expressions of a language  $\mathcal{L}$  and their meanings  $\mathbb{M}$ .

Two crucial aspects can be recognized in this definition. The first one is the set of meanings  $\mathbb{M}$ . What meaning is and how it is represented is discussed in more detail in Section 5.4. The other crucial aspect of Definition 2.15 is the language  $\mathcal{L}$  which is used to describe event content. A language can be understood as a set of terms, or lexicons, and a syntax to connect these terms and form sentences. I deal mainly with terms in this work without focus on syntax. Thus, I reduce the mapping  $\mathcal{S}$  to a relation between the terms of  $\mathcal{L}$  and the set of meanings  $\mathbb{M}$ .

In a distributed environment, each event processing agent of  $A = \{a_1, a_2, \dots, a_n\}$  (producers or consumers) has a set of meanings  $\mathbb{M}_{a_i}$ . Consequently, sets  $\mathbb{M}_{\mathcal{D}_{\square}}$  come from the human users or developers who configure or program the event agents. For simplicity,

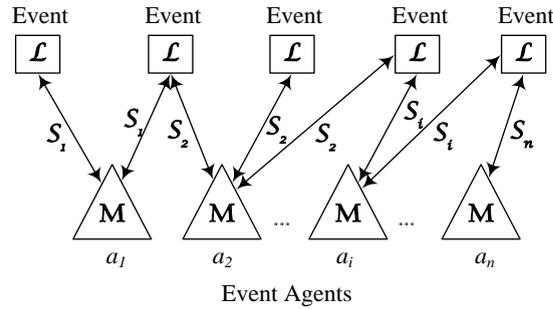


FIGURE 2.2: Event semantic heterogeneity

I assume that all humans have the same set of meanings which contain all the sets, i.e.

$$\mathbb{M} = \bigcup_{1 \leq i \leq n} \mathbb{M}_{a_i}.$$

Let us now assume that each event agent uses a different set of terms, or language, to describe events, being languages  $\mathcal{L}_{a_i}$  for agents  $A = \{a_1, a_2, \dots, a_n\}$ . By discarding the syntax structure of languages and assuming that there is a super set of symbols used by all the agents I get the language  $\mathcal{L} = \bigcup_{1 \leq i \leq n} \mathcal{L}_{a_i}$ . This assumption can be valid for the sake of simplicity. Now, semantic heterogeneity, as shown in Figure 2.2, can be defined as follows

**Definition 2.16** (Semantic Heterogeneity). It is the use of different mappings  $\mathcal{S}_i$  from  $\mathbb{M}$  to  $\mathcal{L}$  by each event agent  $a_i$ .

Last but not least, to make this trait more realistic, it is important to notice that moving between meanings and symbolic terms of the language used to describe the event is guided by surrounding data, or context. A context  $\mathcal{C}_i$  at the event agent  $a_i$  serves then as a parameter to how the semantic mapping  $\mathcal{S}_i$  is interpreted.

### 2.7.3 Openness

While the term ‘open’ has been frequently used in the literature to describe large-scale distributed event systems, e.g. [140], it has not been defined precisely. Thus, herein I draw upon the definition commonly used in systems theory [141, p. 139–153] as *the system that has external interactions in the form of information, energy, or matter transfer through the system boundary*. A boundary here separates the system from its environment. For example, in biology a cell exchanges chemicals with its environment through its membrane and thus it is an open system from this perspective.

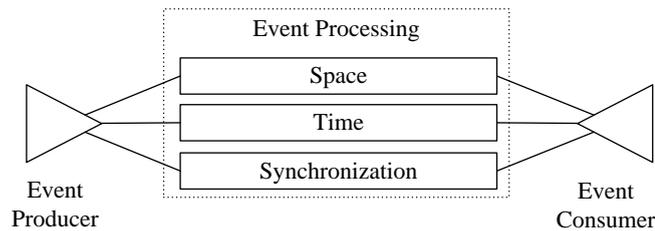


FIGURE 2.3: Dimensions of decoupling

In event processing environments, I consider each event agent as a system. Thus, I define an open system, or environment, from different perspectives which are the concern of this work as follows:

**Definition 2.17** (Open Event System- The Semantics/Context Perspectives). It is the event environment where any event agent can in theory exchange events with any other event agent that uses different event semantics/context i.e. event agent  $x$  which has semantic mapping  $\mathcal{S}_x$  and context  $\mathcal{C}_x$  can in theory exchange events with another event agent  $y$  which has semantic mapping  $\mathcal{S}_y$  and context  $\mathcal{C}_y$ .

## 2.8 The Principle of Decoupling

A principle that can be considered very fundamental to the event-based interaction paradigm is the use of the event to decouple producers and consumers as stated by Etzion and Niblett [10, p. 34]:

“An event has meaning that is independent of its producer and of its consumers, and as a result event producers and event consumers can be completely decoupled from each other. The idea of using the event itself to decouple the event producer and event consumer is a significant difference between event-based programming and application design based on request-response interactions.”

I herein define decoupling using the analysis of Eugster et al.:

**Definition 2.18** (Principle of Decoupling). It is “removing all explicit dependencies between the interacting participants.” [11]

The true impact of this principle is the increase of scalability [11]. Eugster et al. [11] recognize three dimensions of decoupling as shown in Figure 2.3:

- *Space decoupling* suggests that participants do not need to know each other. Producers do not hold references to consumers or know how many of them are actually interacting and vice versa.
- *Time decoupling* means that participants do not need to be active at the same time.
- *Synchronization decoupling* suggests that event producers and consumers are not blocked while producing or consuming events.

Decoupling is also called *implicit interaction* [12, p. 150]. It means that the control over an event-based system has been decentralized into an autonomous version. Mühl et al. [12, p. 150] describes this phenomenon as follows:

“This control has been relinquished deliberately in favour of the loose coupling. It is withdrawn from the components, replacing explicit addressing with the matching of notifications to subscriptions. The explicit control of interaction given in request/ reply approaches is replaced by the implicit interaction in event-based systems.”

I argue that the hypothesis that removing explicit dependencies between event producers and consumers leads to an increased scalability needs to take into consideration that dependencies in fact have been moved to events and thus extra importance and meaning is put inside the event objects. Thus, in my opinion this hypothesis can not be accepted in an absolute sense. But rather, with taking other assumptions into considerations.

As autonomous events can lead to ambiguities in semantics or context, that requires participants to collaborate again in order to solve. That leads to limitations on scalability and as a result undermines the very fundamental reason of why participants are decoupled. I believe that any computational paradigm that tackles event processing at large scales, in distributed, open, and heterogeneous environments must take into consideration that it has valid assumptions that do not break the principle of decoupling, and thus do not affect scalability.

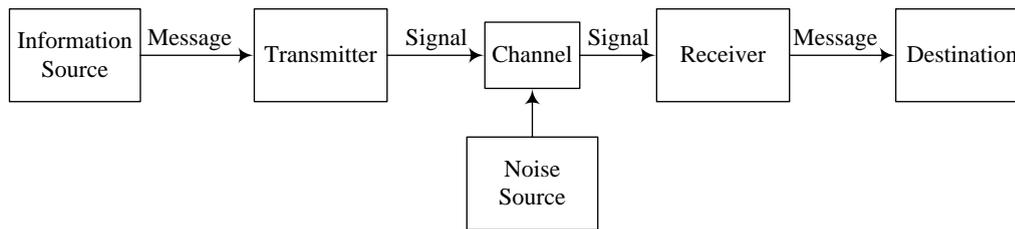


FIGURE 2.4: Schematic diagram of Shannon-Weaver general communication system <sup>2</sup>

## 2.9 A Theory for Event Exchange

Event processing systems at large scales are distributed, open, heterogeneous, with decoupled components that exchange messages. This requires an abstraction which helps better analyse these systems and their challenges. A useful abstraction is a communication model as discussed in Section 2.9.1.

### 2.9.1 The Model of Communication

One of the earliest models is the mathematical model of communication developed by Shannon and Weaver [142] and shown in Figure 2.4. The purpose of the Shannon-Weaver model is to provide a model for technical transfer of information supported by quantification means of information based on the mathematical theory of probability and entropy. It has similarities at an abstract level with the event processing model as presented in Sections 2.5 and 2.6.

The model consists of six elements as illustrated in Figure 2.4: an information source, a transmitter, a channel, noise, a receiver, and a destination. Chandler [143] in his work on semiotics, the theory of signs and meanings, analyses communication models as an aspect of the theory. He recognizes transmission as a basic level of moving signs, or symbols, between participants but which by itself constitutes a small and mechanical fraction of communication [143, p. 178–179]. Chandler describes the Shannon-Weaver model as a model of information *transmission* rather than of information *communication*.

That is due to the fact that it ignores semantic and contextual aspects of communication, which are crucial for communication to succeed. In fact, those has been left out of the model deliberately as stated by Shannon and Weaver themselves for example on semantics:

---

<sup>2</sup>Based on [142]

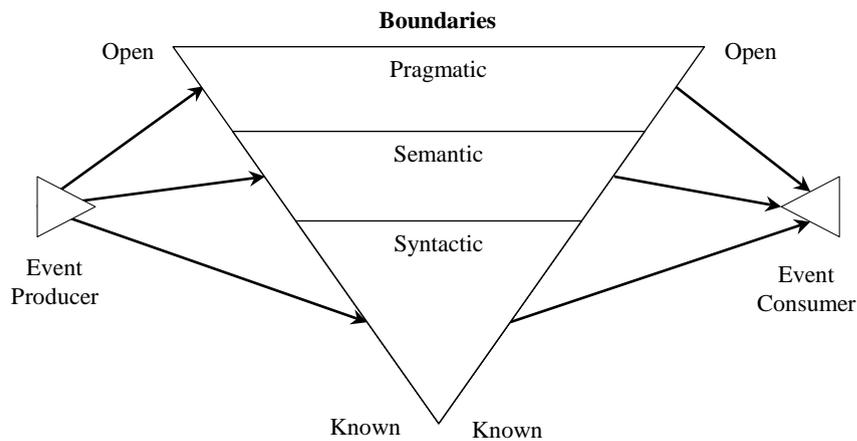
“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem.” [142]

### 2.9.2 Event Exchange as Crossing System Boundaries

A framework for knowledge exchange has been proposed by Carlile [46, 144] within the area of organization science. Its foundations can be traced back to the Shannon-Weaver model with implications for information systems.

Carlile’s framework is concerned with the exchange of knowledge between systems (product development teams in his concrete case). He defines the task of knowledge exchange as a task of crossing boundaries between systems. Carlile recognizes three main levels of boundaries that may exist in a given knowledge exchange scenario:

- *A Syntactic Boundary* describes the boundaries between systems that focus on the sharing and establishment of a common syntax across a given boundary. This view has been established by Shannon and Weaver [142] in their communication theory where syntax has the form of zeros and ones. They claim that once such a syntax is shared, accurate communication can be ensured and the task becomes that of information processing rather than communication. This view has been adopted by system theorists such as Bertalanffy [141]. Carlile [46] sees that crossing syntactic boundaries is synonymous to *transferring* knowledge across those boundaries.
- *A Semantic Boundary* starts to appear when some meanings become unclear or ambiguous. Even when syntax is established, interpretations can be different between the two sides of a boundary. This issue has been left out of the Shannon-Weaver theory as discussed in Section 2.9.1. The essential premise is that a message conveys meanings rather than mere symbols. This has been the emphasis of linguists such as Reddy [145] in his theory of the “*metaphor of conduit*” which states

FIGURE 2.5: Boundaries in knowledge exchange<sup>3</sup>

that language reveals metaphors about communication itself as meanings are conveyed through language containers. Carlile [46] sees crossing semantic boundaries as synonymous with *translating* knowledge across the boundaries.

- A *Pragmatic Boundary* appears when assessing the exchanged knowledge requires a bigger picture of the interacting parties' mutual and conflicting interests and contexts. The origin of the pragmatic view can be traced back to work by semioticians such as Peirce [146]. Carlile [46] sees crossing pragmatic boundaries as synonymous to *transforming* knowledge across the boundaries.

Carlile [46] views boundaries as existing in every scenario, but with different importance. He suggests that boundaries exist in an incremental manner, that is syntactic boundaries always exist, then semantic boundaries can be significant above that, followed by the pragmatic boundaries as shown in Figure 2.5. Carlile argues that the complexity and dominance of semantic and pragmatic boundaries increases when uncertainty increases. That is the case when moving from known environments to open environments with novel types of participants requirements.

Carlile's framework is complemented with the notion of *boundary objects*. This concept has been developed by Star and Griesemer [147] to analyse the heterogeneity in distributed scientific communities. Carlile [46, 144] reuses the concept of boundary objects within his boundary-based knowledge exchange framework. Carlile suggests that

boundary objects, such as design documents in the concrete case of product development, are the key to make tacit knowledge explicit, which leads to the effective crossing of boundaries.

### 2.9.3 Discussion

I herein project the event processing paradigm onto the previously discussed abstractions of communication and knowledge exchange across boundaries. Each event agent can be considered as a system by itself. This system can exchange knowledge with the external world via events. The whole event-based system then becomes a system of systems.

Event agents, i.e. systems, have boundaries (hence Carlile's boundaries [46]) that they have to cross in order to communicate (hence Shannon-Weaver's model [142]) with other systems. Boundaries are syntactic, semantic, and pragmatic. Events are not a mere exchange of symbols, but rather meanings signified by symbols (hence the semiotics view [145]).

Events must effectively cross the three levels of boundaries in order to establish communication between event agents. For this to happen, I think events should be thought of as the boundary objects (hence Star and Griesemer [147]) that must have the effectiveness characteristics of Carlile, i.e. at the syntactic, semantic, and pragmatic levels.

I argue that the current event processing paradigm is focused at crossing lower boundaries, i.e. syntactic, for achieving the task of event transfer rather than of event-based communication. Thus, human agents are needed in the loop to cross semantic and pragmatic boundaries which leads to hindering the paradigm as these tasks are external to it rather than being at the core of it.

The space, time, and synchronization decoupling dimensions of Eugster et al. [11] contribute to event transfer across syntactic boundaries only. Semantic and pragmatic boundaries are inherent in large-scale, open, distributed and heterogeneous environments such as the Internet of Things. This in turn leads to magnifying the problematic nature of *semantic and pragmatic coupling* which contradicts with the fundamental basis of event systems as decoupled and scalable systems as discussed in Section 2.10.

---

<sup>3</sup>Adapted from Carlile's framework [46]

## 2.10 Limitations of the Current Event Processing Paradigm

One of the main requirements for event processing systems is scalability. From a software performance perspective, Bondi [148] defines Scalability as follows:

**Definition 2.19** (Scalability). “The concept connotes the ability of a system to accommodate an increasing number of elements or objects, to process growing volumes of work gracefully.” [148]

On the other hand, a system is non-scalable according to Bondi [148] if:

“we usually mean that the additional cost of coping with a given increase in traffic or size is excessive, or that it cannot cope at this increased level at all.”

From the above discussion, two aspects of scalability can be recognized:

- *Load*, which is the volume of work that a system is supposed to handle.
- *Cost*, that must be paid in return for scalability to meet the load.

Load as a general term has been instantiated in event processing research in different ways including: volume of input streams and the complexity of processing [81], number of subscriptions and the volume of event messages [149], increase in event sources [34], number of producers, number of consumers, number of agents, and size of state [150]. Cost on the other hand is usually realized by adding new machines or processors [81]. In this work, I define a model of load based on an increase in: the number of event producers and consumers, the number of users, the proportion of non-technical users, the level of semantic heterogeneity in the event environment, and the level of context-dependent event processing in the event environment.

Within the current event processing paradigm, the *cost model* to meet scalability for this load can be described based on:

1. The level of agreement between event producers and consumers on the semantic interpretation of events and users’s interests to cross the semantic boundaries.

For instance, a high level of agreement results from the granular agreements on the individual meanings of the terms ‘*energy*’, ‘*power*’, and ‘*electricity*’ as follows:

‘*energy*’  $\Rightarrow$  usable power that comes from heat or another source.

‘*power*’  $\Rightarrow$  a source or means of supplying energy.

‘*electricity*’  $\Rightarrow$  a wire-carried energy used to operate appliances, machines, etc.

A lower level of agreement can be achieved by establishing a quantifiable relationship between the three terms and meanings above, e.g. frequency of co-occurrence, and having a coarse-grained agreement over this relationship, through agreeing on a corpus that encompasses the terms in use.

2. The level of agreement between event producers and consumers on the necessary contextual data to complement events, to cross the pragmatic boundaries. For instance, an event consumer and producer who agree that an energy event should have both the ‘*room*’ and the ‘*floor*’ of the energy consuming device, assume more pragmatic coupling than agreeing on having only the ‘*room*’ of the device in the event.

I dub the first point as *semantic coupling*, and the second one as *pragmatic coupling*. I argue that decoupling in the current event processing paradigm is practised only at the syntactic transfer level, including space, time, and synchronization. However, when higher boundaries become significant at large-scale, distributed, open, and heterogeneous environments such as the Internet of Things, coupling is re-introduced. This leads to a trade-off which hinders the paradigm.

Two problems can be concluded from this trade-off as shown in Figure 2.6:

1. **Problem 1:** decoupling is important for scalability, but scalability to the defined environments needs semantic and pragmatic coupling within the current event processing paradigm, which leads to a trade-off.
2. **Problem 2:** scalability to the assumed environments needs semantic and pragmatic coupling, i.e. agreements, which may not even be feasible due to the large number of participants, non-technical background of users, and the lack of organization of the inherently decoupled and distributed users.

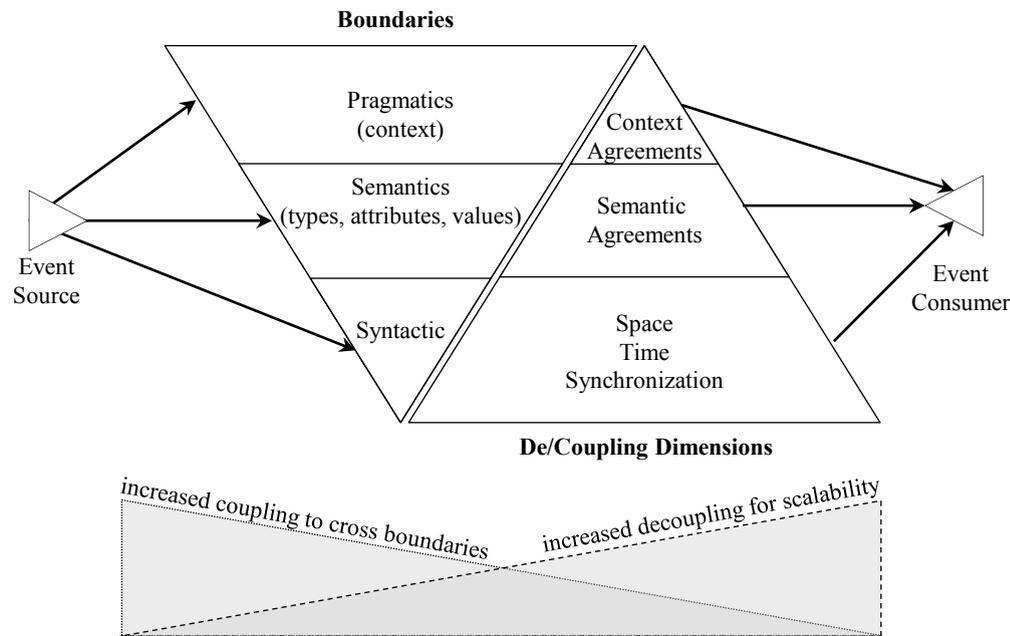


FIGURE 2.6: Trade-off between decoupling and knowledge exchange across boundaries

Thus, I recognize two new dimensions of coupling that exist in current event processing systems that can be added to the coupling dimensions of Eugster [11] as shown in Figure 2.7. I define in the following semantic and pragmatic coupling dimensions.

**Definition 2.20** (Semantic Coupling). The amount of agreement between participants in the event processing environment on mappings between symbols used in event messages and the meanings they refer to. This amount is dependent on the model of mappings, i.e. the semantic model, whether it is symbolic or not, explicit or implicit, and on its granularity as discussed in Sections 5.4, 5.5, and 6.3.

One of the main root problems of semantic coupling dimension is the that most current event systems are based on an exact matching model and symbolic semantics which are not tolerant towards uncertainties of semantics and is very dependent on rigid semantic agreements.

**Definition 2.21** (Pragmatic Coupling). The amount of agreement between participants in the event processing environment on contextual information needed to complement event messages in order to better evaluate users' interests. This amount is dependent on the model of context, where and how the context is found, how it is retrieved, and integrated with events, Sections 5.5 and 7.3.

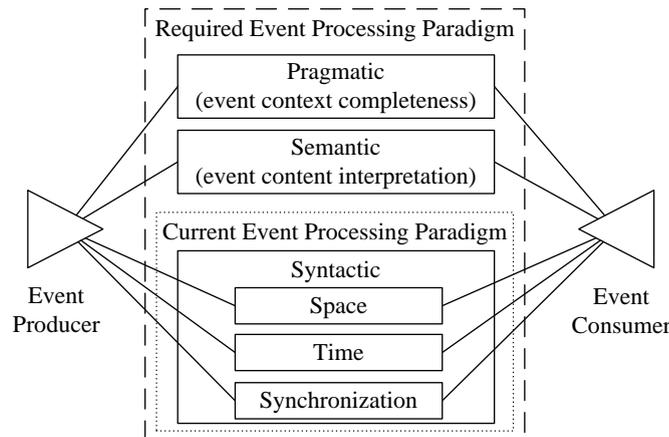


FIGURE 2.7: Dimensions of de/coupling

One of the main root problems that is relevant to the pragmatic coupling is the fact that current event systems are based on a Closed World Assumption (CWA). That is, the processing model assumes that events are complete objects from an information point of view, while they are not.

## 2.11 Requirements, Questions, and Scope

The discussion throughout this chapter leads to a need for extending the current event processing paradigm with a compromise to the trade-off between knowledge exchange across boundaries and the semantic and pragmatic coupling dimensions. Such an extension can be mapped to a set of requirements as follows:

- *R1.* Loose coupling of event processing systems on the semantic dimension. It can be defined as a low cost to define and maintain rules with respect to the use of terms, and to building and agreeing on an event semantic model. For instance, the cost in terms of the effort required to define three rules to cover three events of types: ‘*energy consumption event*’, ‘*energy usage event*’, and ‘*power consumption event*’ is higher than that for defining one rule that can cover all the heterogeneity. Besides, the labour needed to build a taxonomy to establish explicit relationships between the terms ‘*energy*’, ‘*power*’, and ‘*electricity*’ is higher than an automatic or semi-automatic approach that can estimate such relationships.
- *R2.* Loose coupling of event processing systems on the pragmatic dimension. It can be defined as a low cost to define and maintain the context parts of rules,

and to agree on contextual data that is needed in events. For instance, the cost to define an enricher that adds the ‘room’, ‘floor’, and ‘project’ which a device belongs to in an energy consumption event is higher than the cost to define an enricher that could search and discover the needed pieces of information on the fly at the time of matching.

- *R3*. Efficiency of event processing. It can be defined as the timeliness in matching event semantics, and precision in integrating contextual data with events. For instance, an event matcher that can match 1,000 events/sec is more efficient than a matcher that can match 200 events/sec. Similarly, an enricher that enriches an event with data that 90% of which is useful for later processing is more efficient than an enricher that 10% of its complementary data is useful.
- *R4*. Effectiveness of event processing. This can be quantified by the proportion of true positives and negatives achieved by the decider (or matcher), and the effectiveness in completing events with contextual data. For instance, an event matcher that decides on 95% correctly is more effective than a matcher with only 50% accuracy. Similarly, an enricher that complements events with 95% of required data is more effective than an enricher that complements events with 10% of required data.

Two main research questions are formulated:

- *Q1*. The first research question is concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rule and their meanings but rather a form of statistical loose agreements on the meanings (Requirement *R1*). The question is how to achieve timely event matching (Requirement *R3*) with high true positives and negatives (Requirement *R4*) in such a loosely semantically coupled environment?
- *Q2*. The second research question is concerned with the case when event producers and consumers do not have equal assumptions on the amount of contextual information included in events and how much they are complete with respect to evaluating some consumers’ rules (Requirement *R2*). The question is how to complement events with context at high precision (Requirement *R3*) and completeness

needed to meet consumers expectations (Requirement  $R_4$ ) in such a loosely contextually coupled environment?

Finally, I scope the work in this thesis according to the models of information flow processing of Cugola and Margara [8] discussed in Section 2.5.2 as follows:

1. *The functional model.* This work is scoped to a sub-component of the Decider, which is single event matching. Its impacts on related aspects such as pattern matching and complex event processing are partially addressed. A new Enricher component is added to the function model as discussed in Section 4.3.5.
2. *The processing model.* This work is scoped to a single selection policy and a selected consumption policy. Load shedding is out of the scope of this work.
3. *The deployment model.* This work assumes a distribution of the participants of an event processing environment. Nonetheless, a single event processing engine is considered in a centralized deployment model, with the possibility of the existence of multiple distributed event engines.
4. *The interaction model.* This work follows a push-based model of interaction.
5. *The data model.* This work can be generalized to various data models. The specific model of attribute-value records has been used for experimentation in Chapters 5 and 6, while a graph model has been used in Chapter 7.
6. *The time model.* As this work is scoped to single event matching, no semantics of partial or total time order such as happened-before relationships are considered. Nonetheless, impacts on related aspects such as pattern matching and complex event processing are partially addressed.
7. *The rule model.* Rules considered in this work are detection rules, with the awareness of uncertainty in semantics and pragmatics.
8. *The language model.* The language considered in this work is a detection language, with single-item selection operator.

## 2.12 Chapter Summary

This chapter positions the problem within an energy management scenario. Event producers and consumers can use different terms to describe their events and information needs such as ‘*energy consumption*’, ‘*energy usage*’, and ‘*power consumption*’ to refer to the same thing. Consumers may also expect contextual information in events that are not complete such as the *room* or the *floor* where the event originated. To address these challenges, traditional event processing systems depend on explicit agreements on semantics and contexts (or pragmatics) between producers and consumers.

Event processing systems are the outcome of an evolution path of computational paradigms which includes: active databases, publish/subscribe systems, and data stream management systems. The principle of decoupling for scalability represents a cornerstone in the event processing paradigm. It means that event producers and consumers have no explicit interdependencies, that is they do not hold references to each other (space), they are not active simultaneously (time), and they do not block each other (synchronization). Nonetheless, I recognized agreements on semantics and pragmatics as additional coupling dimensions that can hinder the decoupling for scalability principle.

The current data landscape is characterized by a distributed, open, and heterogeneous nature, which magnifies the problem of semantic and pragmatic coupling. This problem has been analysed in this chapter from a communication model perspective and knowledge exchange across system boundaries where events are boundary objects which have to carry semantic and pragmatic information along the way but without introducing coupling in the overall system of systems.

An apparent trade-off has been detected between the need for coupling to establish meaningful communication and the need for decoupling to enable scalability. As a result, this chapter defined the requirements of loose semantic and pragmatic coupling in an effective and efficient manner. These requirements are translated into research questions that drive the main investigation in this thesis.

# Chapter 3

## Related Work

“Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop questioning.”

— Albert Einstein

### 3.1 Introduction

In this chapter, I develop the requirements identified in Section 2.11 into a more elaborate and technical version. Based on the new set of full requirements, a set of previous work that targets some of those requirements, and is deemed relevant to this work is identified. Related work is categorized into six classes of approaches: content-based, concept-based, and approximate event processing, as well as dedicated event enrichers, query-based event fusion, and semantic and context event transformation. Representative approaches of each class are then analysed including a description of the approach, its evaluation, and a critique regarding its addressing of the requirements.

Some parts of the related work analysis in this chapter have been presented to various degrees in the IEEE Internet Computing (2015) [151], the International ACM/I-FIP/USENIX Middleware Conference (Middleware 2014) [152], the ACM Transactions on Internet Technology Journal (ToIT 2014) [153], the ACM International Conference on Distributed Event-Based Systems (DEBS 2015) [47], DEBS 2013 [154], DEBS 2012 [155], and the the International Workshop on Semantic Sensor Networks (SSN 2011) at the International Semantic Web Conference (ISWC 2011)[156].

The first three categories of the related work mainly tackle crossing semantic boundaries. Section 3.4 analyses the category of content-based event processing. Section 3.5 analyses concept-based event processing approaches, while approximate event processing related work is discussed in Section 3.6. The last three categories are concerned with crossing pragmatic boundaries. Section 3.7 discusses the category of dedicated event enrichers. Query-based event fusion approaches are analysed in Section 3.8, and semantic and context event transformation related work is discussed in Section 3.9.

Discussion and a gap analysis of the related work from the perspective of requirements and features is detailed in Section 3.10 along with directions for the proposed approach. Other miscellaneous relevant work is discussed in Section 3.11, and the chapter is summarized in Section 3.12.

## 3.2 Requirements

In Section 2.11 four high-level requirements have been proposed to address large-scale, open, distributed, and heterogeneous event systems. The requirements are:

- *R1*. Loose coupling of event processing systems on the semantic dimension.
- *R2*. Loose coupling of event processing systems on the pragmatic dimension.
- *R3*. Efficiency of event processing.
- *R4*. Effectiveness of event processing.

I develop these requirements herein into more technical versions which help the analysis of the related work. Requirements are elicited from the following works:

- **Stonebraker et al.** [81]. The authors suggest eight requirements for stream processing engines: in-stream message processing, support for high-level and uniform languages, resiliency against stream imperfections, deterministic processing, efficient seamless integration of stored and streaming data, data safety and availability, scalability across multiple processors, and instantaneous real-time response. Five of these requirements (in-stream message processing, support for high-level and uniform languages, resiliency against stream imperfections, efficient seamless

integration of stored and streaming data, and instantaneous real-time response) are mapped to the requirements of this work as shown in Table 3.1.

- **Hinze et al.** [9]. The authors outline the underlying technologies of event-based systems being: handling different forms of contextual information in events, supporting event sequencing and out of order events, homogeneous aggregation, heterogeneous composition/fusion, derivation of higher abstraction events, event enrichment, early filtering, event propagation and notification, handling heterogeneity of platform, processing a large volume of events, supporting mobility of event sources, sensibility to false positives and negatives, and supporting mobility of event subscribers. Four out of these (handling heterogeneity of platform, processing a large volume of events, sensibility to false positives and negatives, and event enrichment) are mapped to the requirements as detailed in Table 3.1.
- **Obwegger et al.** [157]. The authors propose a set of requirements for a user-oriented management of rules in event-based systems. These are: aggregation, situation detection, transformation, expressiveness, efficiency of use, full and system-wide access, decoupling of pattern detection and reaction logic, reusability, ease of use, immediate and transparent rule management, hot deployment, and security. Three out of these (ease of use, immediate and transparent rule management, and transformation) are mapped to the requirements of this work as in Table 3.1.
- **Cugola and Margara** [8]. The authors provide a comprehensive survey of event processing and stream management systems. They define a set of dimensions over seven models of the studied systems as discussed in Section 2.5.2. Three among these dimensions (support for heterogeneous flow, declarative languages, selection) are mapped to the requirements of this work as presented in Table 3.1.
- **Etzion** [10]. The author provides a comprehensive account of event processing systems, discussing various aspects of requirements. Among those, two classical requirements (efficiency, and effectiveness) and two anticipated requirements (inexact matching, and sensibility to false positives and negatives) are mapped to the requirements of this work as shown in Table 3.1.
- **Eugster et al.** [11]. The authors analyse publish/subscribe systems and related technologies. They organize their analysis in two main interrelated dimensions that are decoupling and scalability. Decoupling is focused on space, time, and

synchronization, and not within the areas of semantics or pragmatics. However, I map these two requirements (decoupling, and scalability) to the requirements of this work as presented in Table 3.1.

- **Schilling et al. [34]**. The authors examine a set of requirements and challenges to enhancing scalability and interoperability of complex event processing. Among the requirements are: dealing with heterogeneity, purposeful deployment, context modelling, rule management, and event context enrichment. Three out of these (dealing with heterogeneity, context modelling, and event context enrichment) are mapped to the requirements of this work as shown in Table 3.1.

The elaborated set of requirements dimensions are as follows:

- **R1. *Loose coupling of event processing systems on the semantic dimension*** (resiliency against stream imperfections in [81], heterogeneity of platform in [9], support for heterogeneous flow in [8], inexact matching in [10], decoupling in [11], and dealing with heterogeneity in [34]). This requirement can be elaborated into the following technical requirements:
  - **TR1.1. *Low cost to define and maintain rules with respect to the use of terms*** (support a high-level language [81], ease of use in [157], and declarative languages in [8]). This requirement means that the effort and time put by users to define and maintain the rules should be minimized. It can be understood in terms of a small amount of rules to express the users' needs.
  - **TR1.2. *Low cost of building and agreeing on the event semantic model*** (resiliency against stream imperfect missing data in [81], and immediate and transparent rule management in [157]). This requirement means that the overhead to build a common semantic understanding should be minimized, possibly with less granular items, such as individual concepts, to agree on.
- **R2. *Loose coupling of event processing systems on the pragmatic dimension*** (resiliency against stream imperfections in [81], and decoupling in [11]). This requirement can be elaborated into the following technical requirements:
  - **TR2.1. *Low cost to define and maintain context parts of rules*** (uniform integration language in [81], and ease of use in [157]). This requirement means

that the amount of effort and time put by users to define and maintain the parts concerned with integrating the context shall be minimized.

- **TR2.2.** *Low cost of agreement for contextual data that is needed in an event* (seamless integration in [81], immediate and transparent rule management in [157], declarative languages in [8], and context modelling in [34]). This requirement means that the overhead to build a common understanding of context should be minimized.
- **R3.** *Efficiency of event processing* (in-stream processing and real-time response in [81], processing a large volume of events in [9], efficiency in [10], and scalability in [11]). *R3* can be elaborated into the following technical requirements:
  - **TR3.1.** *Timeliness in matching event semantics* (in-stream processing and real-time response in [81], and processing a large volume of events in [9]). This requirement means a high throughput matching as many events as possible within a unit of time.
  - **TR3.2.** *Precision in integrating contextual data* (efficiently access and combine state information in [81]). This requirement means that event integration with context should use only the relevant complementary data out of the overall available context to complete events for further processing.
- **R4.** *Effectiveness of event processing* (selection in [8], and effectiveness in [10]). This requirement can be elaborated into the following technical requirements:
  - **TR4.1.** *Effectiveness in matching event semantics* (sensitivity to false positives and negatives in [9], selection in [8], and sensitivity to false positives and negatives in [10]). This requirement means a high proportion of true positives and negatives achieved by the decider (or matcher).
  - **TR4.2.** *Completeness of events with contextual data* (integrate stored and streaming data in [81], event enrichment in [9], transformation in [157], and event context enrichment in [34]). This requirement means that events should be as complete as possible, with contextual data, before the matching takes place, having none or a minimum number of non-existing attributes or values.

The mapping of the requirements as suggested in the literature, with the requirements tackled in this work is shown in Table 3.1.

TABLE 3.1: Requirements Dimensions as Defined by Previous Work

Requirement	Stonebraker et al. [81]	Hinze et al. [9]	Obweiger et al. [157]	Cugola and Margara [8]	Etzion [10]	Eugster et al. [11]	Schilling et al. [34]
<b>R1.</b> Loose semantic coupling	resiliency against stream imperfections	handling heterogeneity of platform		support for heterogeneous flow	inexact matching	decoupling	dealing with heterogeneity
<b>TR1.1.</b> Low cost to define and maintain rules' terms	support a high-level language		ease of use	declarative languages			
<b>TR1.2.</b> Low cost to build and agree on the event semantic model	resiliency against stream imperfect missing data		immediate and transparent rule management				
<b>R2.</b> Loose pragmatic coupling	resiliency against stream imperfections					decoupling	
<b>TR2.1.</b> Low cost to define and maintain rules' context parts	uniform integration language		ease of use				
<b>TR2.2.</b> Low cost of agreement on contextual data	seamless integration		immediate and transparent rule management	declarative languages			context modelling
<b>R3.</b> Efficiency	in-stream processing, real-time response	processing a large volume of events			efficiency	scalability	
<b>TR3.1.</b> Timeliness in matching event semantics	in-stream processing, real-time response	processing a large volume of events					
<b>TR3.2.</b> Precision in integrating contextual data	efficiently access and combine state information						
<b>R4.</b> Effectiveness				selection	effectiveness		
<b>TR4.1.</b> Effectiveness in matching event semantics		sensibility to false positives and negatives		selection	sensibility to false positives and negatives		
<b>TR4.2.</b> Completeness of events with contextual data	integrate stored and streaming data	event enrichment	transformation				event context enrichment

### 3.3 Categories of Related Work

The literature on event processing systems related to this work, can be classified into two major categories:

- *Approaches to cross semantic boundaries of event-based systems:* These are the approaches that mainly focus on achieving a good transfer of event semantics across boundaries, such as dealing with synonyms and ambiguities.
- *Approaches to cross pragmatic boundaries of event-based systems:* These are the approaches which mainly focus on conveying contextual data with events.

This classification drives from the fact that the two issues have been seen in the literature as two distinct problems. That does not prevent works from dealing with aspects of both, with varying degrees, due to the relatedness between the two topics. I have discussed in Chapter 2 in detail the view of event processing systems as a communication paradigm. Within this paradigm, semantics and pragmatics play an increasing role with more importance given to their boundaries at larger scales.

Both categories can be further classified into subcategories as follows:

- *Approaches to cross semantic boundaries of event-based systems:*
  - *Content-based event processing.*
  - *Concept-based event processing.*
  - *Approximate event processing.*
- *Approaches to cross pragmatic boundaries of event-based systems:*
  - *Dedicated event enrichers.*
  - *Query-based event fusion.*
  - *Semantic and context event transformation.*

Each of these categories is discussed in the following sections. The main approaches under each category are discussed in detail. How these approaches address the requirements is summarized in Table 3.2, and their features are discussed in Table 3.3, along

with discussions and gaps analysis. The main indicators for the proposed approach are the conclusion of this chapter.

## 3.4 Content-Based Event Processing

In content-based event processing, event sources and consumers use the same event types, attributes, and values without any additional description of meaning external to rules and events. This is the case assumed in traditional content-based publish/subscribe and event processing systems where the matcher performs exact string comparison between terms. The approach has high semantic coupling between parties and works effectively in environments with a low level of data heterogeneity.

### 3.4.1 Carzaniga et al. (SIENA)

#### *Description*

SIENA [22] represents a broad class of publish/subscribe systems that match and route an event based on its content and its relevance to subscriptions. Other examples include Elvin [98], Jedi [99], and the Java Messaging Service (JMS) [100]. Carzaniga et al. realize an event notification service that is highly scalable to networks like the Internet. To achieve such scalability, they recognize a set of principles: loose coupling, expressiveness, a best-effort distributed service, primitive typed-attributes data model, along with content-based routing.

Loose coupling is realized through an implicit invocation design style, represented by an event-based software architecture. Expressiveness is provided by a powerful, but yet simple, event model, and a subscription model with a set of simple operators: ordering relations, substring, prefix, suffix, and an operator that matches any value. Best-effort distribution is adopted, where issues of distributions such as race conditions are not handled. The event model is a set of type-attribute-value triples, which is simpler than the typed notifications found for instance in CORBA [15].

Single and patterns of notifications can be matched based on their contents and relationships. Coverage relationships between subscriptions are defined and leveraged for optimization. A distributed topology of servers is assumed, where clients are connected

to each server. The routing algorithm is based on two principles: late downstream replication of events, and early upstream evaluation of subscriptions. Both principles are realized via forwarding subscriptions and forwarding advertisements of intended notifications. As a result, routing paths can be constructed over the topology to minimize traffic and improve scalability.

### ***Evaluation***

In [22], Carzaniga et al. provides a qualitative analytical evaluation of SIENA. The first part of the analysis focuses on positioning SIENA at a trade-off scale between expressiveness and scalability. At one extreme lie the least expressive and highly scalable channel-based event notification services. At the other extreme lie the most expressive, application-defined event structures, types, and operators which could potentially be Turing-complete and lead to undecidable algorithms, leading to less scalability. Carzaniga et al. show that SIENA stands in between the two extremes, and thus addresses the requirements of better expressiveness and scalability.

The second part of the analysis is concerned with the computational complexity of coverage and routing algorithms. In a non-optimized version of the algorithms, matching an event to a subscription has a time complexity of  $O(n + m)$ , linear in  $n$  the number of subscription's predicates and  $m$  the number of event's attributes. Similarly, computing the coverage between two subscriptions or two advertisements is  $O(n.m)$ , polynomial with the number of predicates/attributes in each. Matching a pattern is constrained to matching a sequence, leading to a time complexity of  $O(l(n + m))$  where  $l$  is the length of the pattern, linear to the number of subscriptions. Carzaniga et al. show that algorithms that conform to their proposal could be computed effectively, leading to a scalable notification in terms of time.

In [158], Carzaniga evaluates SIENA quantitatively using a simulation framework and synthetic workloads. Each simulation consists mainly of two elements: a network of clients and servers sites which represents the topology, and a scenario configuration that represents the advertise, publish, subscribe, unadvertise behavioural requests of parties. Carzaniga generates a total of 2,200 simulations, with up to 1,000 sites of publishers and servers, and up to 10,000 subscribers. The links in the network are configured with an abstract *cost*, which is not materialized by the author but can be mapped, in reality,

to delay or bandwidth. The main metric for evaluation is an overall abstract *cost* that reflects the effect of site-to-site costs on various topologies, algorithms, and workloads.

The main result of the evaluation is an empirical proof that distributed topologies show a sub-linear increase of cost with increasing numbers of subscribers and publishers, outperforming a linear case for centralized topologies. For instance, at 1,000 publishers, and 10,000 subscribers, a centralized topology requires an abstract cost of 70 million, compared to just below 2 millions for an acyclic peer-to-peer topology with subscription forwarding. Results also show that the cost-per-service request, e.g. advertise or subscribe, is dominated by internal message passing when the number of publishers and subscribers is low.

### *Critique*

The strength of SIENA is represented in its scalability to a large number of sites, publishers, subscribers, events, and subscriptions. This scalability stems from a good utilization of the network effect by supporting distributed topologies and algorithms. Carzaniga et al. also recognize imperative principles for any event service to scale to environments such as the Internet including: loose coupling, expressiveness, a best-effort distributed service, primitive typed-attributes data model, along with content-based routing. The analytical evaluation of Carzaniga et al. justifies these principles, while empirical validation is only provided for the distribution of the service.

The limitations of SIENA stem from its complete dependence on the content of events and subscriptions. While scalability is the main motivation behind SIENA, the semantic heterogeneity aspect is not covered, with a lack of sources of meaning that can support the content. In the simulations, Carzaniga states that all objects of interest publish the *same events*. For SIENA to scale into heterogeneous environments, a large number of subscriptions will need to be deployed in order to cover the heterogeneity of events. That is due to the exact string matching model that is followed in content-based routing, adopted by SIENA. In terms of requirements, the loose semantic coupling is not covered, while efficiency of matching is high. Effectiveness in matching is high, in fact 100%, given that all required subscriptions are defined. Requirements on context and pragmatics are not covered in this work.

### 3.4.2 Eugster et al.

#### *Description*

Eugster et al. investigate in [23] the integration of publish/subscribe primitives into a strongly typed object-oriented language such as Java, focusing on linguistic primitives based on four principles: type safety, encapsulation preservation, application-defined events, and composable event semantics. The authors implemented the primitives on using the infrastructure offered by the Distributed Asynchronous Computing Environment (DACE) as well as structural reflection for dynamic event methods invocation.

Events are considered as first-class citizens in the programming language where they are defined using normal Java objects called *obvents* by the authors. New constructs to realize publish/subscribe in Java are put into effect through the utilization of a pre-compiler that transforms calls to specifically generated adapters.

A key outcome of this proposal, which makes it relevant here, is the introduction of a type-based publish/subscribe model. That is, the type hierarchy in the programming language, i.e. the Java class hierarchy, is leveraged. As a result, subscribers can make subscriptions to a class called *StockObvent* for instance, and expect events of the subclass *StockRequest* to be matched. This approach forms an extension to content-based matching, where a *kind* of events is supported by a type hierarchy.

#### *Evaluation*

The authors provide an analytical discussion on design issues of the approach, along with its interoperability. The authors state that their approach does not support interoperability in heterogeneous platforms, due to the lack of a neutral specification language such as the one used in CORBA [15] for instance.

In [159] Eugster and Guerraoui conduct an empirical evaluation to analyse the Java reflection contribution to performance. They show that the overhead of dynamic method invocation on objects can reach around 0.13 ms/invoke for 10 objects, compared to just over 0 for static Java invocation. They provide an optimized version of matching based on avoiding redundant invocations, and enforcing static filters. The testbed for evaluation is synthetic, of one producer on a single machine, publishing to a set of subscribers equally distributed over two networks of 20 and 60 stations respectively.

The main metric of evaluation is the throughput, i.e. the number of messages processed per a millisecond. Results show that optimized matchers compete with the static invocation that lacks any optimization as it is a pure Java behaviour. The throughput of optimized and static matching decreases gradually when the number of subscribers increases. It achieves around 1 message per *ms* for 120 subscribers.

### *Critique*

The main problem targeted by this work is the integration of the publish/subscribe with programming languages. The strengths of the approach come from that integration, making the language more ready to have the publish/subscribe logic within an application program. The type-based approach is also more expressive than channel-based publish/subscribe. It frees subscribers to an extent from expecting exact types of events.

However, the approach is limited to types, rather than attributes or values, which are dealt with using Java Boolean expressions called filters. Interoperability is restricted to the levels below semantics as discussed by the authors, making it not ready for interoperability issues such as semantic heterogeneity. Besides, having the language type model as the basis of matching, makes the approach more coupled. In fact, type safety leads by itself to compile-time errors when subscribers use non-supported classes.

Regarding requirements, the loose semantic coupling is not covered, while efficiency and effectiveness in matching are high. This is true given that heterogeneity is handled manually by agreeing on type hierarchies, or defining classes and filters for all expected types, properties, and values. Requirements on pragmatics are not covered in this work.

### **3.4.3 Fiege et al. (Rebeca)**

#### *Description*

Fiege et al. [24] tackle several issues that arise when events have to cross organizational boundaries between event-based applications. These concerns are management, customization, heterogeneity, and security. They propose a model to address these problems based on the introduction of *scoping* into the event-based service. Scopes are meant to control visibility within distributed event-based applications.

A scope can be mapped in reality to an organizational boundary. For instance, an event about temperature in an area of a warehouse can be visible only to subscribers within the warehouse. Thus, a *warehouse* scope defines the boundary that in turn represents the basis for the management of the distributed application's components.

The concept of scopes is orthogonal to publications and subscriptions. Scopes can, in turn, belong to superscopes. Scopes interface with each others via interfaces, which define what events can cross the boundaries. Such interfaces also define how events get transformed when crossing the boundaries, either to accommodate a new vocabulary of the destination scope, or to include more data to convey context. Fiege et al. propose the use of ontologies to define transformations at the boundaries. Scope administrators take the responsibility to define transformation and mapping logic.

### ***Evaluation***

The scoping model is implemented within the Rebeca middleware, which has served as a prototype to evaluate several aspects of publish/subscribe systems. Routing within Rebeca has been evaluated empirically in [160] using synthetic datasets from the stock exchange domain. The main two metrics of interest are the size of the routing tables, and the filter forwarding overhead. Ideally, routing algorithms should target the minimization of both metrics.

Experiments show that the size of routing tables increases sub-linearly with identity-based routing, compared to a linear increase with simple routing. At 60,000 subscriptions, identity-based routing with advertisements require 80,000 routing entries, compared to 230,000 for simple routing with advertisements, on the chosen topology. The average control messages as forwarding overhead correspond to 1% and 4% respectively.

Fiege implements scoping over Rebeca's architecture and routing mechanisms in what is called *integrated routing*. The new algorithms are not evaluated empirically, but the author provides a comprehensive analysis of the correctness of scope-based routing related algorithms in [161]. The particular aspect of semantic mapping and context transformation and the associated effort put by administrators is not evaluated separately from the algorithms underneath.

### *Critique*

The work of Fiege et al. takes its importance from the explicit notion of boundaries, which are important for scalability in event systems as discussed in Chapter 2. Scopes form the actual realization of boundaries. Thus, issues related to crossing boundaries in an effective semantic and pragmatic way are tackled at scope interfaces as proposed by Fiege et al. in this work. Scopes are important as they subdivide the distributed system into stable regions of semantics and pragmatics, which interface with each other.

The realization of scope interfaces as transformations, which are based on symbolic semantics like ontologies, requires administrators to define these transformations. The fact that the underlying routing and matching at scope interfaces follow exact string matching limits the flexibility of crossing scope interfaces and thus limits scalability.

In terms of requirements, an attempt to address loose semantic and pragmatic coupling is made, but the result is not conclusive. Efficiency is high as in content-based routing, given that heterogeneity is handled manually by the user with correct and exact scope interfacing. Effectiveness of semantic and pragmatic boundary crossing is not evaluated.

## **3.5 Concept-Based Event Processing**

In this category, participants can use different terms and values and still expect matchers to be able to match them properly thanks to an explicit knowledge representation that encodes semantic relationships between terms. Example knowledge representations are thesauri and ontologies that describe the meaning of each concept and its properties and relationships with other concepts. Building and agreeing upon such a knowledge representation suggests an explicit dependency between parties via a conceptual model.

### **3.5.1 Petrovic et al. (S-ToPSS)**

#### *Description*

S-ToPSS [25] is a semantic publish/subscribe system meant to solve the problem of selective information dissemination within semantically heterogeneous environments. The system processes incoming events using three phases that can be done optionally or

combined. The first stage accepts incoming events and generates for each original event, called root in S-ToPSS, a set of events by replacing each property with a set of *synonyms* for that property. The second phase operates on the value level and generates for each incoming event from the previous stage a set of events by replacing values with others that have *taxonomic* relationships with them. The last phase is an ad-hoc stage where individual *mapping* functions can be written to generate further events out of incoming events from the previous stages.

The resulting set of new events that have been generated during these three phases are then matched in a Boolean exact matching model with subscriptions. It extends current matchers to do the Boolean matching, so they take advantage of already existing matchers. Thus, the authors extend content-based matching to *semantic matching*. Petrovic et al. also build on the conceptual model to define semantic coverage between subscriptions, where a subscription for events with the term '*vehicle*' covers a subscription with the term '*car*'.

### ***Evaluation***

The authors state that the approach is demonstrated through a job finder scenario, and the workload is generated to simulate publications and subscriptions from companies and clients. Nevertheless, the work does not provide details about the workload or the evaluation criteria that are followed.

### ***Critique***

S-ToPSS can be considered as a general approach for semantic matching in publish/subscribe systems. Its main strength is that it extends the syntactic based matching and coverage relations with a semantic-enabled matching and coverage allowing publishers and consumers to use semantically equivalent terms that are syntactically different.

The work does not provide a concrete method for generating synonyms or for exploiting taxonomic relationships and does not discuss design issues associated with the proposed model. The approach of generating new events out of the original ones can overwhelm the system with many events and thus hinder the efficiency requirement.

Concerning the requirements identified in this work, S-ToPSS externalizes semantics into an explicit model represented by taxonomies. It thus provides a degree of loose coupling in developing the rules. However it is limited by the fact that it is based on an

agreed-upon ontology. Ontologies and taxonomies are labour-intensive to build which means that when the environment scales to include other ontologies the system does not scale without fundamental changes [43]. Effectiveness can be assumed full given a fully agreed-upon semantic model. Pragmatic aspects are not addressed in this work.

### 3.5.2 Wang et al. (OPS)

#### *Description*

Wang et al. [26] target the problem of trade-off between expressiveness and efficiency in publish/subscribe systems through a combination of both. They use Semantic Web technologies such as RDF [60] and DAML+OIL [162] to describe events and subscriptions. The authors propose a graph-based event model and a graph-based subscription model with variables. The problem then becomes one of graph matching.

Subscriptions are decomposed into statement patterns to exploit commonalities between them. An index is built over the statement patterns, taking into account the concept model provided by the ontology, by extending the index with a set of inferred facts about a subscription. Each event is scanned in a breadth-first manner, and the matching process exploits the subscriptions index to match scanned facts from the event. Matching on the vertex level becomes an exact string matching.

#### *Evaluation*

Evaluation is done on a theoretical as well as an empirical level. Theoretical evaluation proves the correctness of the matching algorithm formally. Experiments compare the system with another system that uses a different graph matching algorithm called *Decomposition* by Messmer and Bunke [163]. Experiments focus on the metrics of matching time and memory usage per event, as well as the scalability with the number of subscriptions.

A synthetic workload is generated by varying the number of subscriptions and the number of classes in the ontology. Matching takes place on a single machine. The number of events is not stated, but each event has 50 vertexes with 55 edges. The number of subscriptions ranges from 500 to 10,000, each has 10 vertexes and 11 edges. The ontology has 10 classes, each of which has 2 properties.

The experiment shows an efficient matching time versus the number of subscriptions with a linear relationship for OPS, where the matching time is 1.2 sec for 10,000 subscriptions. The proposed OPS outperforms the Decomposition algorithm where the matching time of the latter reaches 500 ms compared to 1 ms for OPS at 20 subscriptions. Time efficiency improves when subscriptions become very diverse in using classes as early filtering becomes more selective. Space efficiency is high with a constant upper limit of 5 index nodes when the number of subscriptions increases.

### *Critique*

The system extends syntactic matching with semantics-enabled matching allowing publishers and consumers to use semantically equivalent terms. The work proposes an efficient matching algorithm for graph models, which are universal and essential for the heterogeneity in data models. This work opens the black box of the matcher, such as in [25], and proposes a white box approach, allowing for more optimization opportunities.

On the other hand, the work considers only taxonomic relationships, not including relatedness for instance. It also assumes only one ontology with a relatively small number of classes and properties and does not consider multiple semantic models. Regarding requirements, OPS externalizes semantics into an explicit model, providing some loose coupling in developing the rules. Effectiveness and efficiency are high given that a fully agreed-upon semantic model exists. Pragmatic aspects are not addressed in this work.

### **3.5.3 Zeng and Lei**

#### *Description*

Zeng and Lei [27] target the problem of heterogeneous event schema, and how event systems should tackle that. They use a *relational* approach to event-based systems, in which a subscription can select the source event(s) as with SQL in databases. The approach allows the selection of single events, and the correlation of multiple events. The approach transfers the problem of heterogeneity from the application level to the middleware level. It can operate over already existing publish/subscribe systems as black boxes.

To handle heterogeneity, and loosen the semantic coupling, the authors propose an ontology-based model. An ontology, as they define it, is similar to an object-oriented

class diagram, with classes, properties, and inter-relationships such as *subClassOf*. A term also has a set of synonyms defined in the ontology. The approach follows a query rewriting method, where a subscription is translated into various equivalent subscriptions based on the ontological model. The generated subscriptions are then matched against the events in a Boolean exact matching manner.

The work approaches pragmatics with the source search mechanism. This component allows the search for *sufficient* event sources that could at schema-level match a subscription. It also makes use of the ontology repository.

### ***Evaluation***

The approach is not empirically evaluated. However, the authors provide an analytical discussion that compares the proposed approach of subscription rewriting with the approach of event rewriting. The latter can be exemplified by S-ToPSS [25]. The authors make the case for the viability of subscription re-writing on the basis of not overwhelming the matcher with many generated events. Besides, events need to be rewritten every time they arrive, while subscriptions could be rewritten at registration time.

### ***Critique***

The approach's main strength comes from the acknowledgement of the data heterogeneity problem and the use of a novel model based on subscription rewriting based on ontologies. The proposed ontological model supports multiple relationships including taxonomic, properties, and dependencies. Besides, the concept of *sufficient* event sources for a subscription is important, although not fully developed, with respect to contextual data enrichment.

Nonetheless, the approach is limited by the fact that the assumed ontological *repository* is labour-intensive to define. The underlying matching model is Boolean which makes it less flexible with respect to potential uncertainties such as a missing synonym in the ontology. Regarding the requirements of this work, Zeng and Lei slightly address loose coupling through rules rewriting. Effectiveness and efficiency can be high given that a fully agreed-upon semantic model exists. However, it is also not clear how many subscriptions could be rewritten from an original one leading to a combinatorial number of rewritten subscriptions when the number of concepts and synonyms increases, leading

to a reduced throughput Pragmatic aspects are not elaborated beyond the *sufficient source* concept.

### 3.5.4 Blair et al. (CONNECT)

#### *Description*

Blair et al. [28] tackle the problem of increased heterogeneity and increased dynamism. The authors recognize the need for novel approaches to interoperability between networked systems through middleware. Middleware as discussed by Blair et al. includes all systems that could abstract systems data, interfaces, and behaviours in a way that allows easy interoperability. The formed entity, as a result, is a system of systems, enabled by the middleware.

The authors recognize that the current approaches to interoperability, which are dominated by standards, are in fact effort-intensive to develop and agree on. Thus, they propose the approach of *emergent middleware*, the focus of the CONNECT project. Emergent middleware systems are not static entities, but rather dynamically generated and tailored glues between networked systems. Such types of middleware are made real through a set of *enablers*, which are software technologies that collaborate to generate the middleware.

Enablers are three: discovery, learning and synthesis enablers. The discovery enabler is responsible for recognizing the concepts in a system's interfaces. The learning enabler attaches semantic annotations to interfaces based on determined interaction behaviour. The synthesis enabler takes the completed systems models from the previous enablers' output and generates the middleware system.

A central element of the CONNECT framework is ontologies as conceptual models. Ontologies are the basis for describing interfaces, behaviour, and synthesization of emergent middleware. For instance, a domain ontology of the travel domain can be used to find out a subsumption relation between *selectFlight* and *selectTrip*, where the former is a part of the later. As a result, the emergent middleware can translate one request from the first system to a request in the other system.

### ***Evaluation***

This work is qualitatively analysed through two experiments. The first experiment provides an analysis of the interoperability in the application and middleware layers. It is based on two travel agencies systems, one is European and the other is American. Each system uses a set of *affordances* represented by its interface functions such as *selectFlight* and *selectTrip*. One is implemented by SOAP, and the other by HTTP REST. The analysis shows how a domain ontology can, in fact, be used as a basis to create a middleware that can translate one request to another.

The second experiment deals with reasoning about interoperability at the network layer. For this reason, the authors show how ontologies can be used to interoperate two heterogeneous Vehicular Ad Hoc Network (VANET) protocols: BBR and Broadcom. The experiment shows how ontologies and Semantic Query-Enhanced Web Rule Language (SQWRL) based rules can serve as a conceptual model to classify a network packet, and decide which fields in it can be used to fill in an output packet from a different protocol.

### ***Critique***

This work is important as it recognizes the limitations of common approaches such as standards to achieve interoperability. The strength of this approach comes from its identification of a set of enablers that surrounds the middleware and helps generate it. Middleware, as the authors define, is a dynamic entity that can in principle serve for *eternal* interoperability as it is dependent on its networked systems. I think this work can be seen as a framework for interoperability, rather than a single approach. Various ways to realize conceptual models or enablers can still fit in the emergent middleware framework to generate the emergent middleware.

This work is generic for middleware, but not for event-based systems specifically, although the challenges are similar. The assumption of the existence of a domain ontology that networked systems partially use to be composed, may be challenging at very large scales [43]. Ontology alignment techniques may be needed in this case, as suggested by the authors. Nonetheless, ontology alignment and matching need to take the networked systems into consideration and uncertainties embedded in this process need to propagate into the emergent middleware generation. Such details are left out of this work,

and can enhance the fundamental principle of generating emergent middleware suited for networked systems.

From a requirements perspective, this work tackles the loose semantic coupling requirement through emergent middleware systems that ensure decoupling. However, the use of domain ontologies, which are previously agreed on, can introduce coupling. Effectiveness and efficiency of the approach are high, given that two systems are interoperable and that ontologies are available. The contextual pragmatics aspect is out of the scope of this work.

## 3.6 Approximate Event Processing

Approaches in this category are distinguished by a matching model that is not Boolean. They support one form or another of uncertainty, probability, or ranking in event processing. This gives the event engine more flexibility to deal with heterogeneity and thus improves its ability to address the loose coupling requirements.

### 3.6.1 Zhang and Ye (FOMatch)

#### *Description*

FOMatch [29] is similar to the previous category from the point of view that it uses a common understanding of the domain to achieve semantic matching. However, it leverages a fuzzy ontology model of the domain where relationships between concepts are not certain but rather weighted edges. Before matching, a pre-processing step takes place where a closure of the ontology is built in order to construct an index of terms and relations with scores of the degrees of relations using transitivity of some relationships such as the *subClass* relationship.

Actual matching is done by leveraging commonalities between statement patterns of subscriptions and each term in the event is compared to terms in the index to conclude a scored match. Scores are aggregated for each event and the result is compared to a threshold to achieve a conclusive matching result.

### *Evaluation*

The work follows an empirical evaluation where a synthetic workload of 1,000 events is simulated from the touristic tours domain, each of which has 12 attributes. The number of subscriptions range from 10,000 to 100,000, each of which has between 1 and 12 attributes. The main metrics of interest are pre-processing and matching time, along with precision, recall, and F<sub>1</sub>Score concerned with the proportion of correctly matched events out of the relevant ones. The baseline is S-ToPSS [25] on a single machine.

Results show that FOMatch outperforms S-ToPSS for all numbers of subscriptions, with a linear scalability. At 100,000 subscriptions, FOMatch requires 1,500 ms for pre-processing and 400 ms for matching, compared to 3,000 ms and 1,100 ms respectively for S-ToPSS. For effectiveness evaluation, 500 events were created and users were asked to write subscriptions for them. Users manually picked the ground truth of relevance between events and subscriptions. Results show an F<sub>1</sub>Score of 90% for FOMatch versus 77% for S-ToPSS.

### *Critique*

This model is relevant in that it acknowledges the uncertainty that underlies semantics in a semantic model. Besides, it leverages uncertain matching that is flexible at large-scales. The time efficiency of the model is high too. The use of precision, recall, and derived measures also forms an important feature for evaluating event systems at large scales as it acknowledges a best-effort approach to matching.

However, the model depends on a common ontology. Concerning requirements, this work provides partial loose coupling in developing the rules and approximate matching. Effectiveness and efficiency are relatively high, but with less than 100% for effectiveness. Pragmatic aspects are not addressed in this work.

## **3.6.2 Liu and Jacobsen (A-TOPSS)**

### *Description*

A-TOPSS [30, 31] is an approximate publish/subscribe model that addresses the requirement for users to express their interests in numeric values of events using textual values. The authors approach the problem using fuzzy sets where fuzzy membership

functions are trained in order to define the membership of a numeric value in a textually described category. The matching between subscriptions and events is then based on aggregation of fuzzy functions values. The result is a score that reflects the degree of a match.

### ***Evaluation***

A-TOPSS is evaluated empirically using a synthetic workload. Approximate events and subscriptions are used first, and they are transformed into crisp ones to render the cases comparable to an exact paradigm. For efficiency evaluation, the main metric of interest is the matching time. Different mechanisms to implement the approach are compared. An algorithm named the *float-list-based model* is efficient with a minuscule matching time of less than 1,000 ms at 100,000 subscriptions, compared to 140,000 ms for the *bit-10values* algorithm. For effectiveness evaluation, precision, and the F<sub>1</sub>Measure metrics are used. Results show a constant relationship with the number of subscriptions with an F<sub>1</sub>Measure of 95%.

### ***Critique***

A-TOPSS addresses an important aspect of semantic decoupling that is value approximation. It acknowledges the compromise in matching precision at large scales. However, types and properties are not supported by the model. Direct extension of the model to concepts and properties may not be straightforward as it is designed around the existence of numeric values on one side of the matching.

From the requirements perspective of this work, A-TOPSS provides loose coupling in developing the rules, and the model behind it, but it is limited to numeric-to-strings approximation. Effectiveness and efficiency are relatively high, but with less than 100% for effectiveness. Pragmatic aspects are not addressed in this work.

### **3.6.3 Drosou et al. (PrefSIENA)**

#### ***Description***

Drosou et al. [32] address the problem that all the matched events to a subscription in traditional event-based systems are considered equally important. The authors propose an approach based on event ranking using user-defined preferences as well as diversity.

The subscription language allows the user to express the preference for some attributes over others. A preferential graph can be defined on subscriptions, with coverage relationships between them. Top- $k$  diverse events matching a subscription are ranked and returned. In a stream, a continuous periodic return of top- $k$  matchings is returned with a sliding window.

### ***Evaluation***

The proposed approach extends SIENA [22]. The authors use a real movie-dataset from the Internet Movie Database (IMDB) <sup>1</sup> of 58,788 movies. Events and subscriptions are generated from this set. For efficiency evaluation, authors compare a brute-force delivery algorithm with their heuristic algorithm. The metrics of interest are time and diversity. For 30 events, and  $k = 8$ , the brute-force requires 0.5 hours, compared to just 38 ms for the heuristic algorithm. The difference in time efficiency comes at the cost of decreased diversity of returned events, but the drop is below 1% in all cases.

For effectiveness evaluation, the authors measure the number, average rank, and diversity of the returned events. Ideally, the former one should be lowered while the latter two should be improved. Scenarios consist of 2,000 events, 930 of which match the subscriptions. For 400 events, and a window length  $k = 20$ , only 100 are returned, i.e. around 11% of matching events are ranked and returned, with an average rank up to 90% and a diversity between 80% and 90%.

### ***Critique***

This approach is not directly associated with semantics or pragmatics. However, it is relevant from the point of view that it extends current event systems with the ranking and best-effort paradigm that is crucial for large scales. The work builds a case for approximate matching in the delivery of events, where even matching events can be missed and some only are prioritized.

With respect to requirements, the approach is flexible regarding the organization and expectations of event publishers and consumers, and thus can be seen to address the loose coupling requirements. Efficiency and effectiveness are high with less than 100% quality of matching.

---

<sup>1</sup><http://www.imdb.com/>

### 3.6.4 Wasserkrug et al.

#### *Description*

Wasserkrug et al. [33] tackle the problem of evaluating event processing rules over uncertain events. The uncertainty they address could come from unreliable sources such as an inaccurate sensor reading, or an unreliable network such as packet loss. Also uncertainty is generated by an inability to determine if a phenomenon occurred given the available sources. The authors address two problems: the scalable derivation of complex events under high volume sources, and the correct derivation and propagation of uncertainties to conclude a complex event uncertainty.

The authors extend the concept of selectability, or early filtering, of events to exclude irrelevant uncertain events to some uncertain event rule derivation. They also devise a probabilistic derivation by translating CEP rules into Bayesian networks. To enhance the performance of derivation, the authors approximate the outcome of a network using a set of samples. However, instead of sampling the network, which would be inefficient due to its construction cost, they sample the event processing rules that correspond to the networks.

#### *Evaluation*

The authors evaluate their approach using a synthetic events and rules set. They use 20,000 explicit events and a set of rules of two levels of hierarchy, on a single machine. The main metrics of interest are the accuracy and the performance measured by event processing rate per second. For accuracy, the baseline is the theoretical expectation. For performance, they compare with a deterministic engine which is Amit [116].

The accuracy results show that actual probabilities, which are based on specially built Bayesian networks, always lie within 95% of the probabilities derived by the sampling approximation. This means that for a confidence interval of 5%, the sampling approximation leads to equivalent results to a none-approximated derivation.

Performance results show that the event rate decreases sub-linearly when the number of possible worlds, corresponding to the number of samples, increases. These results prove a scalable method for uncertain event derivation. In [164] the authors show that for a relatively small number of samples, corresponding to 0.2 approximation, the performance

is at the same number of magnitude with the Amit [116] deterministic engine, which is a magnitude of *hundreds* of events/sec. These results show that an uncertain engine can compete with a deterministic engine in terms of time performance.

### *Critique*

This work's strength comes from its comprehensive tackling of uncertainty within event engines. The work proves empirically that the extended selectability and sampling can deliver event engines capable of dealing with uncertain events, which the authors argue are inevitable in modern applications. The work also does not limit itself to single events, but shows how a full model can handle complex event processing too.

I agree with the authors on the importance of handling uncertainty natively in event engines. The work does not address semantics or pragmatics in event engines, but if loose coupling can lead to an uncertainty of events and their derivation, the work becomes related.

For requirements, Wasserkrug et al. do not focus on loosening semantic or pragmatic coupling. However, the approach, if considered related from the uncertain loose coupling perspective, shows effective and efficient results.

## **3.7 Dedicated Event Enrichers**

This category is mainly concerned with event enrichment via ad-hoc dedicated agents that are tailored specifically to particular situations. Such approaches are non-native to the event processing paradigm where the enrichment behaviour is pushed to the end user and less integrated with the rest of the features of event processing engines.

### **3.7.1 Schilling et al. (DHEP)**

#### *Description*

Schilling et al. [34] tackle the problem of scaling event processing engines to large-scale environments, such as the Smart Grid, through distribution. They particularly address the problem of distribution over heterogeneous event engines. They define a high-level

system, DHEP, that sits on top of heterogeneous distributed nodes. DHEP uses a meta-language that serves as a high-level unified language for defining contextual state objects, events, and rules.

DHEP has a runtime environment, which is a middleware that can consume events from an event bus, decode them, enrich them, and produce new ones back to the event bus. Encoding and decoding are common tasks to address heterogeneity at the syntax level, such as with protocols. A wrapper wraps existing event engines and adapts them to DHEP. Some rules have the ability to query state databases and enrich events with context before these events can cross system boundaries. DHEP estimates the cost of rules' evaluation, and thus decides on the distributed placement of rules over the nodes.

### ***Evaluation***

The system has been evaluated empirically using synthetic benchmarks. The main metrics of interest are the latency introduced by the system on top of the typical event engines that are wrapped, and the latency of enrichment. For the system latency, a smart meter sending *power request* events was simulated with one filtering rule. Results show that the system adds almost 150% of latency over the CEP engine that is wrapped (IBM's AMiT).

For enrichment, the system is evaluated by one rule that retrieves an integer value from a MySQL database and adds it to input events. Results show that enrichment adds a significant overhead to the rule latency. They also show that enrichment enabled with caching can reduce the added latency from around 1,000 – 1,400 to 50 units of time.

### ***Critique***

The strength of DHEP is that it recognizes the heterogeneity of event processing nodes, and tries to provide a solution on top of that. Another strength comes from the identification of enrichment as a task that needs to be reflected in distributed event processing solutions. The DHEP meta-language can be seen as an approach that tries to address both semantic and contextual pragmatic boundary crossing in distributed event engines.

Nonetheless, a unified language as proposed is an added layer that still needs to be maintained by administrators. Those need to establish semantic and contextual agreements that can scale to limit only. Furthermore, the evaluation focuses on performance and

latency while the proposed unified meta-language and its usability and ability to address heterogeneity is not investigated.

Concerning the requirements, this work attempts to address semantic and pragmatic coupling but the results are not conclusive. Efficiency is slightly worse than the wrapped CEP engine due to added overhead. Effectiveness in matching and completeness of events can be assumed 100% given that agreements on semantic and context modelling is established, as the system follows the exact model of the underlying CEP engines.

### 3.7.2 Hohpe and Woolf

#### *Description*

Hohpe and Woolf [18] abstract good practices in Enterprise Integration (EI) into a set of patterns of Message-oriented Middleware (MoM). Among the categories they define for patterns, there is the category of *Message Transformation*. Under this category, there is a set of patterns that are: *Envelope Wrapper*, *Content Enricher*, *Content Filter*, *Claim Check*, *Normalizer*, and *Canonical Data Model*. Those patterns can be seen as the actual patterns usually followed to address the requirements of this work.

#### *Evaluation*

The work of Hohpe and Woolf does not provide particular instantiations, or evaluation, of patterns but rather abstract guidelines. However, the authors claim to have designed the patterns based on a shared practice within MoM-based enterprise integration.

#### *Critique*

The main strength of this work is its abstraction of patterns that could enhance the design of event-based integration solutions. However, this work dedicates nodes that specialize in specific tasks such as enrichment or semantic normalization. An event engine should be able to address these tasks at once to scale into highly heterogeneous and open environments, which go beyond enterprise integration systems.

Patterns such as *Normalizer* and *Content Enricher* assume an exact model and a full control and agreements of semantics and contexts in the environment. These assumptions are only valid within small environments such as enterprise systems, the motivation

for the work. Besides, there is no open disclosure of how the patterns can be verified in real settings, through surveying the experts' opinions for instance.

With respect to requirements, patterns mainly abstract content-based messaging. Semantic and pragmatic coupling is thus comparable to content-based approaches that are coupled. Effectiveness and efficiency are high given that agreements are established, but that assumption is limited to the scale of enterprise integration closed systems.

## 3.8 Query-Based Event Fusion

Approaches in this category adopt declarative languages similar to SQL. Such languages support operators of semantics similar to relational *join*, enabling the fusion of streams of events with background context data.

### 3.8.1 Arasu et al. (CQL)

#### *Description*

The authors examine in [35] the problem of defining abstract semantics for a declarative language that can query relations and streams at the same time. The problem arises with non-monotonic and complex queries that include aggregations, subqueries, windowing, relations, and streams, etc. The authors define exact semantics for continuous queries based on two data types: relations and streams. They also define three black-box operators that are only characterized by their input/output rather than implementation; Those are stream-to-relation, relation-to-relation, and relation-to-stream operators.

Instantiating these black-boxes vary. For instance, in CQL the SQL language can be used to instantiate relation-to-relation operators. The authors use sliding windows to instantiate stream-to-relation operators. They define three operators to create streams from relations: *Istream*, *Dstream*, and *Rstream*. Semantics for these operators are defined, with an implementation in the *STREAM* data stream management system. Optimizations such as query planning are also investigated.

### *Evaluation*

Arasu et al. provide in [35] an analytical discussion to show the relative ease of capturing stream processing requirements. In [165] Babu et al. examine the adaptive optimization of the STREAM management system. They use synthetic data with generated streams of 32 bytes tuples on a single machine. Tuples are fed into the system, and the metric of evaluation is mainly the number of tuples per second that the system can process.

Results vary based on the variable parameters that reflect the actual adaptive and caching behaviour. However, the main result of interest within this context is that the system can join 3 relations with up to 40,000 tuples/sec and that number drops with more joining relations, and reaches around 5,000 tuples/sec for 9 relations.

### *Critique*

The main strength of this work lies in its abstraction of the semantics of stream querying to relations using a generic model. Thus, it unifies streams of events with contextual data and brings the task of joining into the core of the processing engine. The adoption of a declarative query language to achieve this also eases the fusion between events and their context, although this aspect is not evaluated quantitatively.

Nonetheless, the work adopts a semantic for the CQL operators that assume full control and understanding by the engine over the schema of both streams and relations. It thus serves as a fusion approach given that all events and contexts are known. From a requirements perspective, this work does not tackle the semantic heterogeneity or coupling problem. It also does not provide a solution to the loose pragmatic coupling requirement. Its efficiency and effectiveness are high given that all the assumptions made are valid, which may be only the case in small and controlled environments.

## **3.8.2 Teymourian et al.**

### *Description*

Teymourian et al. [36] tackle the problem of fusing complex event processing with knowledge bases. They extend semantic complex event processing, i.e. systems that use RDF and ontologies to describe events and declarative rules to represents patterns. They add the querying of external knowledge bases, which are described using RDF,

ontologies, and rules. The authors formalize complex event patterns based on a logical knowledge representation (KR) and interval-based event calculus.

The purpose of their work is to make the complex event engine aware of more information about the events, through taking into account their background knowledge. Thus, events can be detected based on their type hierarchy, temporal/spatial relationships, and also depending on their relationship to other non-event objects. Consequently, expressiveness and flexibility of the event engine improves.

The authors categorize event rules into 5 categories based on the role of the query concerning the background knowledge base and its relation to the event detection part in the event rule. Based on this categorization, the authors propose a set of query planning strategies that decrease the latency for event rule execution. The authors implement their approach with SPARQL and RDF knowledge bases.

### ***Evaluation***

The authors use two real-world datasets: live stock market event stream from Yahoo! finance, and background knowledge about companies from DBpedia. They link both sets manually by linking on the company stock market symbol and its corresponding DBpedia URI. They use two machines, one for each set. The DBpedia dataset has 288 million RDF triples, hosted in a Virtuoso triple store. The Prova rule engine has been used for event processing.

The metric of interest is throughput, i.e. the number of events per second which are fused with the background knowledge base according to a variety of queries. Results show that throughput reaches 280,000 events/sec for simple event rules, and gets to 500 – 4,000 events/sec with complex ones in terms of the dependency between events and the background knowledge queries. Results also show that throughput decreases drastically when the size of background knowledge, e.g. number of RDF triples, returned by the query increases. For instance, for simple event rules this ranges from 280,000 events/sec for a few triples to around 1,000 events/sec for 1,400 returned triples.

### ***Critique***

The strength of this work lies in its fusion approach of events before they can be considered for complex event detection. This fusion can increase the ability of the event engine to detect more situations given the available background knowledge. The work

also adopts a unified model for both events and background knowledge and provides a principled way to improve the performance of event enrichment.

However, this work tackles the problem of event completeness with context, but users are still assumed to know fully how the background queries shall be used and how data shall be fused with events. Thus, regarding requirements, loose pragmatic coupling is not supported although it is made easier with a declarative and unified model. The semantic aspect is equivalent to that of conceptual event processing, as they both rely on a top-down semantic model represented by the ontologies. Effectiveness and efficiency of the approach are high, but given that the assumptions on full semantic and pragmatic agreements exist, which may be only the case in small and controlled environments.

### 3.8.3 Le-Phuoc et al. (CQELS)

#### *Description*

Le-Phuoc et al. [37] address the problem of scalable integration between Linked Data streams, and background Linked Data. The authors propose a white box approach in which operators such as windowing, relational, and streaming operators are given semantics within a Linked Data framework. Thus, optimization through an adaptive planning of query execution can be done natively by the query engine. Intermediate transformation of data and queries into corresponding black box query engines and stream engines is not further required. As a result, query execution delay can be lowered.

#### *Evaluation*

Evaluation is conducted with 5 query templates selected to cover most operators. For stream data, the authors use RFID-based tracking data from the Open Beacon community<sup>2</sup>. The data represents the movement of the attendees at a research conference. For static data, simulated DBLP records have been generated.

The main metric of interest is the query execution time. The system has been compared with two similar engines: C-SPARQL [166] and ETALIS [167]. Results show that the average execution time for single queries ranges from 0.47 – 21.83 milliseconds, compared to 99.84 – 395.18 milliseconds for C-SPARQL, and 0.06 – 469.23 for ETALIS. Results

---

<sup>2</sup><http://www.openbeacon.org/>

also show that execution time stabilizes when the number of contextual triples increases, compared to a linear increase for the other systems. It also increases linearly with the number of queries. For instance, the execution time for query  $Q_1$  increases from 1 millisecond for 1 query, to 100 millisecond for 1,000 query.

### *Critique*

The strength of this work comes from its adoption of a native white box model to handle Linked Data streams and background data. Thus, execution time can be saved by avoiding intermediate transformations. Linked Data is a generic representation of data in terms of syntax, and uses vocabularies for semantics. Such an approach has the potential to provide a generic model for stream and background data.

This work assumes a full knowledge by the user of how background information can be found and fused. Thus, for requirements, loose pragmatic coupling is not enabled. The semantic aspect is equivalent to that of conceptual event processing. The approach does not support reasoning, e.g. through a type hierarchy. Thus, effectiveness of semantic matching is limited. Effectiveness and efficiency of the background data fusion are high if the assumptions on full semantic and pragmatic agreements exist, which might not be valid at large scales such as the Internet of Things.

#### **3.8.4 Anicic et al. (EP-SPARQL)**

##### *Description*

Anicic et al. [38] propose an approach to bridge the gap between event processing systems that lack integration with background knowledge, and reasoning over background knowledge that lacks dealing with rapidly changing data. The authors propose EP-SPARQL which is a language for which they define syntax and formal semantics. Syntax and semantics are an extension of SPARQL, where queries are translated into algebraic expressions. New operators such as SEQ, which expresses that a graph pattern strictly comes after another graph pattern in a stream, are given formal semantics.

The language's execution model is founded in logic programming and is capable of inferencing over temporal and static data. The execution model is based on Event-Driven Backward Chaining (EDBC) of event rules. That enables incremental detection

of event complex event situations. EDBC rules are logic rules, and that is how they can be integrated with background logic-based knowledge bases.

### ***Evaluation***

For evaluation, the authors have implemented the ETALIS system in Prolog. Two main tests have been conducted: event processing capability, and stream reasoning capabilities. For event processing, Esper 3.3.0 [48] has been used as a baseline. The main metric of interest is throughput, measured by the number of events processed per second. The systems were tested for two queries from stock market scenarios. Results show a throughput of 8,200 – 9,400 events/sec compared to 7,000 – 9,000 events/sec for Esper, depending on the temporal window size. When the window size is defined by the number of events, ETALIS scores steadily around 25,000 events/sec, compared to 10,200 events/sec for Esper.

For stream reasoning, only results of ETALIS are reported. A set of 40,080 sub-classes have been used, with a maximum class-hierarchy depth of 8. The purpose is to detect if a subject in a stream is of a particular class. The main metric of interest is the delay caused by the inference. The results show that inference causes delays that increase linearly with the number of triples, from 500 milliseconds for 5,000 triples, to 2,000 milliseconds for 20,000 triples.

### ***Critique***

One strength of this work comes from its unification of background knowledge and event processing by logic programming. In principle events are not seen as different objects from their contextual information, but rather as information that is complemented with the background knowledge. Another strength comes from the embedding of semantic reasoning in the process as well. Thus, this work, in fact, acknowledges both semantics and pragmatics as two related problems on a logical basis.

However, the use of logic programming requires an effort by the logician to define semantic and pragmatic assumptions explicitly. This is a labour-intensive task and may not be scalable. The semantic reasoning is only evaluated with a type hierarchy, but that is generic given the existence of sufficient logic rules.

Regarding requirements, the work does address effective and efficient semantic matching and contextual background knowledge integration. However, loosening semantic

and pragmatic coupling is not effectively addressed due to basing the model on logic programming. Logic programming is a pure symbolic model that requires an explicit definition and maintenance. This may not be scalable in a loosely coupled environment.

## 3.9 Semantic and Context Event Transformation

Approaches in this category handle events individually and perform a set of transformations on them to move from one semantic model to another. The transformation is typically more specific to the approaches, so I do not put them in the previous categories but dedicate the last category to them.

### 3.9.1 Freudenreich et al. (ACTrESS)

#### *Description*

Freudenreich et al. [39] address the need for event interpretation in event-based systems based on their corresponding contexts. For instance, an event producer may use *meters* to measure *distance*, while a consumer could be interested in the measure in *yards*. The authors propose an approach that could be built upon existing publish/subscribe systems. The approach is based on a context handler component that transforms events before they go into a broker network, or before they get received by a consumer.

For transformation to happen, a context repository is stored on the level of a broker network. The context repository contains agreed-upon hierarchy of contexts, such as *root*, *European*, and *German*, and contexts for time and measurement units. Contexts also have transformation functions between various nodes in the context hierarchy. The producers indicate the context of its event, and the same for consumers. Context handlers drive the conversions according to the source and destination contexts.

#### *Evaluation*

The work has been empirically evaluated with synthetic messages over a distributed setting. The approach was implemented over ActiveMQ [168], a publish/subscribe message broker, and compared with a baseline of pure content-based publish/subscribe with no transformations, and with a transformation based on reflection. A producer-to-consumer

ratio of up to 1 : 10 has mainly been used. The main metric of interest is latency in processing an event.

Results show that in a distributed setting, the baseline scored 850 – 1,000 microseconds for 1 – 10 consumers respectively. The proposed approach scored 20% – 60% more than the baseline’s latency, outperforming the reflection-based approach that was 1,300 – 2,850 microseconds. Another interesting measure provided by the authors is the implementation effort. They show that, for the proposed approach, a total of 15 extra lines of code is required, compared to 29 for the reflection-based approach.

### *Critique*

This work targets the same problem of this thesis, taking into consideration specifics of event-based systems especially decoupling. The strengths of this work come from acknowledging contextual transformations as native components in event processing. The context handler component is made central to handling messages. The evaluation of the implementation effort provided by the authors is valuable as it makes it explicit how an approach to the semantic and context transformation problem should be reflected in usability to loosen the coupling.

Nonetheless, the downside of this work is that the context model is labour-intensive to build as it is formed of conversion functions. Context in this work is defined in a way that makes it closer to semantics rather than to background knowledge. From a requirements perspective, this work loosens semantic coupling to a degree, but results are not conclusive. The model is effective and efficient, given that context management and conversion functions are all defined, which may be unfeasible at large scales. Pragmatic aspects are not covered.

### **3.9.2 Cilia et al. (CREAM)**

#### *Description*

Cilia et al. tackle in [40] the joint problem of semantic heterogeneity and contextual dependencies of event interpretation. The authors propose an approach that tackles these problems at the notification service level, rather than the application level. The approach is comprised of three components: the shared conceptual/contextual model,

the local conceptual/contextual models at the publishers and consumers sides, and the mediators for semantic and contextual transformation. Producers and consumers represent their events and subscriptions in their vocabularies and contexts. Such local models are called the *matching models*, and they are subsets of the agreed-upon model.

The mediators, or adapters, convert events and subscriptions into the shared models. They also enrich events with contextual information from the producer side. The notification service operates with the newly formed data items. At the receivers' side, events are turned into suitable models for the subscriber according to its local adapters. The authors implement the conceptual part of this approach in the CREAM middleware [41]. CREAM also features composite event detection over the single event semantic approach.

### ***Evaluation***

The authors provide an analytical discussion of the proposed approach, comparing it with two other conceptual groups that are: the implicit agreements outside of the middleware, and the explicit handling of heterogeneity at the application level. The analysis leads to the superiority of the proposed approach as it enables a loosely coupled mode for tackling the issue of heterogeneity rather than completely depending on agreements. An empirical account for the proposed approach in terms of effort, effectiveness, or efficiency is not discussed.

### ***Critique***

There are two main strengths of this work: the first one lies in its adoption of an approach that targets loosening the coupling in the notification service use, and thus enabling scalability, and the other one lies in its unification of the semantics and context heterogeneity problems. That reflects the unified nature of the two as they both serve a better interpretation of events, and enable them to cross semantic and pragmatic boundaries.

Nonetheless, the approach followed in this work assumes a shared semantic and contextual model. Achieving such models may not be feasible at high scales of distribution [43]. Besides, a significant effort is assumed at the adapters/mediators sides to implement conversion and enrichment functions. While this can be practical at enterprise and

small scales, it may be challenging when semantic and contextual boundaries become harder to cross at large scales, the thing that is not evaluated in the work.

From a requirements perspective, there is an attempt to address semantic and contextual loose coupling. Effectiveness and efficiency are not articulated and measured.

## 3.10 Discussion and Gap Analysis

The previous discussion can serve as the basis for a gap analysis, which can happen at two levels: requirements and features. The requirements level provides indications at the problem space while the features level provides indications for the solution space.

### 3.10.1 Gap Analysis at the Requirements Level

Table 3.2 summarizes the discussion made above on the addressing of requirements by each category and specific approaches. From this review of the literature, the following conclusions and gap analysis can be done:

- *Content-Based Event Processing* approaches are mainly efficient with the timely matching and routing of events. They assume an implicit agreement on the semantics of events outside of the event engine and express agreements in terms of rules and subscriptions. This class of approaches matches events on the basis that they are complete, leaving all issues with pragmatic agreements outside of the event engine.
- *Concept-Based Event Processing* differs from the content-based approaches in that they make semantics explicit through knowledge representation models such as ontologies and class hierarchies. Given agreements on these explicit models, efficient and effective detection of positive and negative matching can be achieved. Nonetheless, the agreement on explicit models may become by itself an infeasible task to achieve due to heterogeneity. This class of approaches also leaves pragmatics out of its scope.

TABLE 3.2: Requirements as Addressed by Previous Work

		R1. Loose semantic coupling		R2. Loose pragmatic coupling		R3. Efficiency		R4. Effectiveness	
		TR1.1. Low cost to define and maintain rules' terms	TR1.2. Low cost to build and agree on the event semantic model	TR2.1. Low cost to define and maintain rules' context parts	TR2.2. Low cost of agreement on contextual data	TR3.1. Timeliness in matching event semantics	TR3.2. Precision in integrating contextual data	TR4.1. Effectiveness in matching event semantics	TR4.2. Completeness of events with contextual data
Content-based event processing	Carzaniga et al. [22] (SIENA)	-	-	NA	NA	++	NA	++	NA
	Eugster et al. [23]	-	-	NA	NA	++	NA	++	NA
	Fiege et al. [24] (Rebeca)	-+	-	-+	-	++	++	NE	NE
Concept-based event processing	Petrovic et al. [25] (S-ToPSS)	+	-	NA	NA	+	NA	++	NA
	Wang et al. [26] (OPS)	+	-	NA	NA	++	NA	++	NA
	Zeng and Lei [27]	+	-	-+	-+	++	NE	++	NE
	Blair et al. [28] (CONNECT)	++	-+	NA	NA	++	NA	++	NA
Approximate event processing	Zhang and Ye [29] (FOMatch)	+	-+	NA	NA	++	NA	+	NA
	Liu and Jacobsen [30, 31] (A-TOPSS)	+	-+	NA	NA	++	NA	+	NA
	Drosou et al. [32] (PrefSIENA)	-+	-	-+	NA	++	NA	+	NA
	Wasserkrug et al. [33]	-+	NA	NA	NA	+	NA	+	NA
Dedicated event enrichers	Schilling et al. [34] (DHEP)	-+	-	-+	-	+	+	++	++
	Hohpe and Woolf [18]	-	-	-	-	++	++	++	++
Query-based event fusion	Arasu et al. [35] (CQL)	NA	NA	-+	-	++	++	++	++
	Teymourian et al. [36]	+	-	+	-	NE	++	++	++
	Le-Phuoc et al. [37] (CQELS)	+	-	+	-	NE	++	+	++
	Anicic et al. [38] (EP-SPARQL)	+	-	+	-	++	++	+	+
Semantic & context event transformation	Freudenreich et al. [39] (ACTRESS)	+	-+	NA	NA	++	NA	++	NA
	Cilia et al. [40, 41] (CREAM)	+	-+	+	-+	NE	NE	NE	NE

++ the requirement dimension is well covered  
 + the requirement dimension is partially covered with positive results  
 -+ there is an attempt to address the requirement dimension but the solution is not effective  
 Legend - the requirement dimension is poorly covered  
 -- the requirement dimension is very poorly covered  
 NA the requirement dimension is not addressed or the focus of the research  
 NE the requirement dimension is not evaluated

- *Approximate Event Processing* uses explicit models of semantics along with approximate matching of events. This approximate matching allows a loose semantic coupling due to their ability to deal with uncertainties of users on semantics. Time efficiency is high, but less effectiveness is achieved due to the approximate model that allows some false positive/negatives to occur. These approaches do not deal with pragmatic and contextual coupling and scalability.
- *Dedicated Event Enrichers* are orthogonal approaches to classes that deal with semantics. They mainly focus on integrating events with their contexts. They depend on an implicit understanding of the pragmatics around events that are implemented by developers through a set of ad-hoc enrichment logic. This keeps context handling out of the event engine and represents a level of coupling that limits scalability where significant contextual boundaries exist.
- *Query-Based Event Fusion* approaches focus on the integration of events with their context using an approach similar to joins in databases. This approach is effective and efficient. However, the fact that a full understanding of event contexts and their need for matching is assumed and encoded by developers as join statements causes a pragmatic coupling that limits scalability with contextual boundaries.
- *Semantic and Context Event Transformation* represents a set of approaches that consider semantic and contexts to have one nature and impact on event matching. They are effective and efficient in matching and completing the events. Thus, they handle semantic and pragmatic interoperability. Nonetheless, semantic and pragmatic models are explicit, e.g. granular conversion functions, requiring agreements that form a coupled mode that is not scalable in heterogeneous, open and distributed environments.

As a conclusion, the following gap in the literature at the requirements level can be detected:

**Gap Analysis- The Requirements Level.** *The event processing literature lacks approaches that unify the problems of semantic and contextual pragmatic interoperability despite their close nature and importance for event interpretation, and at the same time keep loose coupling on these dimensions for the purpose of scalability for semantic and pragmatic boundaries.*

### 3.10.2 Gap Analysis at the Features Level

Based on the discussion of each approach in this chapter, a set of features can be extracted which gives an idea of the main building blocks and assumptions made by related work. The gap analysis of features helps in Chapter 4 proposing the main hypotheses of this work's approach to answer the research questions. The features are:

1. *Matching Model*, which details if the model uses exact string matching, includes semantic matching such as synonyms and hyponyms, or if it is approximate.
2. *Semantic Model*, which details the type of semantics used, whether it is implicit or explicit, top-down symbolic, or bottom-up statistical.
3. *Domain Specificity Cost*, which details the cost to make the semantic model suitable for a specific domain or situation.
4. *Semantic Interoperability Cost*, which details the effort needed to make the class of approaches work within an open, heterogeneous environment.
5. *Context*, which details the assumptions about the context of events whether it is ignored and events are assumed to be complete, fully known by the developers, or partially left to the engine.
6. *Context Retrieval*, which details how contextual data is retrieved from the source.
7. *Context Search*, which concerns whether the engine is capable of searching for contextual data or if it has to be specified all externally to it.
8. *Context Fusion*, which concerns how contextual data is added in the events.
9. *Enrichment Cost*, which concerns the effort needed to make the class of approach work within an open environment with significant pragmatic boundaries due to dependencies on contextual data of events.

Table 3.3 summarizes the features of each class of approaches.

TABLE 3.3: Features as Addressed by Previous Work

	Matching	Semantics	Domain Specificity Cost	Semantic Interoperability Cost	Context	Context Retrieval	Context Search	Context Fusion	Enrichment Cost
<b>Content-based event processing</b> [22–24]	exact string matching	not explicit	defining many domain rules	defining many rules	event assumed complete	NA	NA	NA	NA
<b>Concept-based event processing</b> [25–28]	Boolean semantic matching	top-down symbolic	defining domain-specific ontology	granular shared agreements	event assumed complete	NA	NA	NA	NA
<b>Approximate event processing</b> [29–33]	approximate matching	top-down symbolic	defining domain-specific ontology	granular shared agreements	event assumed complete	NA	NA	NA	NA
<b>Dedicated event enrichers</b> [18, 34]	NA	top-down symbolic	NA	NA	fully known	ad-hoc	NA	ad-hoc	defining many enrichers
<b>Query-based event fusion</b> [35–38]	Boolean semantic matching	top-down symbolic	defining domain-specific ontology	granular shared agreements	fully known	query	NA	join	defining much join logic
<b>Semantic &amp; context event transformation</b> [39–41]	Boolean semantic matching	top-down symbolic	defining domain-specific ontology	granular shared agreements	fully known	encoded in events by developer	NA	transformation	defining many conversion functions

Legend: NA means that the feature is out of the scope of the research

From Table 3.3, the following observations are made:

- Most approaches use an exact Boolean model of matching. Such a model can be sensitive to errors and less tolerable with missing data, and thus assumes a significant level of agreements that can limit scalability.
- Most approaches depend on top-down symbolic models of semantics to achieve semantic interoperability. These symbolic models can restrict the scalability of event systems due to the semantic coupling associated with these granular semantic models.
- Most approaches require a significant effort to adapt into domain-specific semantics, limiting the ease of crossing system boundaries.
- Most approaches either assume events are complete and ignore contexts or handle context externally to the event engine assuming it is fully known. This can restrict the scalability of event systems due to the pragmatic coupling associated with the models.
- There is no uniform and native way in event engines to retrieve, search, and fuse context with events. This leaves handling the pragmatic boundaries external to the event engines, leading to pragmatically coupled environments and thus limits scalability.

As a conclusion, the following gap in the literature at the level of features can be detected:

**Gap Analysis- The Features Level.** *The event processing literature lacks approaches that unify the problems of semantics and contextual pragmatics uniformly and natively to the event engine. The literature is mainly based on symbolic semantics, exact matching, ad-hoc domain specificity, and ad-hoc enrichment.*

### 3.11 Other Relevant Approaches

I herein discuss two relevant approaches that can be argued to address some aspects of the problem of semantic and pragmatic coupling.

### 3.11.1 Schema Matching

Schema matching approaches have been used in the database community to address the problem of semantic heterogeneity in database schema [169]. It can be argued that several schema are mapped before a traditional event system is put in use. The event system would use the schema mapping to translate the events or subscriptions.

I argue that there be three main limitations to this approach:

- The assumption that existing schemas are in place can be invalid in event systems as the users, who compose events and subscriptions, may not comply to a predefined schema. That leaves open the use of many possibilities of the terms that can be used. In fact, the ‘*schema-less*’ assumption about data has been regarded as a more realistic assumption at large scales in the database community [170, 171].
- It is known that schema matching is intrinsically uncertain [172, 173]. That is, each mapping between two schemas is associated with a probability distribution that quantifies the mapping. Schemas are typically large and concern a domain, while events and subscriptions are small items that concern atomic concurring of a subset of a domain. Thus, an a-priori mapping may not be useful for the event-to-subscription mapping. Technically, they would have different probability distributions.
- Using a-priori mappings to translate events or subscriptions to all possible variations to cover heterogeneity may lead to an exponential number of newly generated events or subscriptions. This can put a burden on the traditional event engine underneath limiting its scalability.
- A-priori mappings limit users control over the meanings of events and subscriptions that they may want to adapt to suit a particular situation.
- This approach does not address the aspect of pragmatics and event enrichment.

All in all, the limitation of this approach lies in its externalization of the semantic handling out of the event engine. I argue that such an approach shall be adapted and reduced to become native to the event engine, and that is one of the lines followed in this work as discussed in Chapter 5.

### 3.11.2 Approximate Query Answering Over Databases

In [174] Freitas et al. propose an approximate query processing approach for databases based on distributional semantics. They analyse natural language queries and build a query plan for Linked Data databases. To bridge the semantic gap between the query and the database, they employ a statistical model of semantics that is built automatically from textual corpora [175]. Freitas et al. further devise a vector-space index of the database named the  $\tau$  - *space*.

The strengths of this approach are the ease in building the semantic model and its ability to tackle natural language sentences. The approach is limited within an event processing context due to the following:

- The paradigm in [174] is a query paradigm suitable for databases. Event processing, on the other hand, is an active, on-the-fly, and timely paradigm. The timeliness in event processing needs different types of optimizations such as commonalities between subscriptions. That has an effect on the actual matching model.
- Freitas et al. analyse a query as it is input by the user, and the database is indexed beforehand. Events and subscriptions, on the other hand, may have various interpretations that are defined by users according to particular situations. Thus, the event engine shall allow the adaptation of interpretations on-the-fly.
- Adaptive thresholds are used to cut off relevant answers to a query, which can then be returned to the user. However, single event processing should keep probabilistic scores of the matching to be used later for complex event processing, a case not relevant within a database querying context.
- Query answering over databases typically uses a closed world assumption about the database. Thus, considerations regarding pragmatics and enrichment are out of the scope.

I argue that some aspects of this approach shall be adapted to the event processing paradigm, and that is one of the lines followed in this work such as extending the query model with probabilistic and top- $k$  matching, and common-predicate optimization in Chapter 5; equipping statistical semantics with tags to adapt it to particular situations in Chapter 6; and dynamic event enrichment in Chapter 7.

### 3.12 Chapter Summary

In this chapter, the requirement of loose semantic coupling has been elaborated into the technical requirements of low cost to define rules with respect to the use of terms, and low cost to build and agree on the event semantic model. The requirement of loose pragmatic coupling has been elaborated into the technical requirement of low cost to define context parts of rules, and low cost of agreement on contextual data that is needed in events. The requirement of efficiency has been elaborated into the technical requirements of timeliness in matching and precision in integration with contextual data while the requirement of effectiveness has been elaborated into the technical requirements of effective matching and completeness of events with contextual data. Requirements are backed up by similar requirements from the literature.

The literature has been analysed to project related work with respect to the identified requirements. Related work has been classified into six categories. The first three: content-based, concept-based, and approximate event processing mainly address semantic interoperability. The last three: dedicated event enrichers, query-based event fusion, and semantic and context event transformation mainly tackle pragmatic interoperability.

Related work analysis revealed that the event processing literature lacks approaches that unify the problems of semantic and contextual pragmatic interoperability, which keep loose coupling on these dimensions for the purpose of scalability. It also showed that the event processing literature lacks approaches that unify these problems uniformly and natively with the event engine. The literature is mainly based on symbolic semantics, exact matching, ad-hoc domain specificity, and ad-hoc enrichment of events.

## Chapter 4

# Approximate Semantic Event Matching and Dynamic Enrichment

“An idea can be tested, whereas if you have no idea, nothing can be tested and you don’t understand anything.”

— James D. Watson

### 4.1 Introduction

To tackle the main requirements of loose semantic and pragmatic coupling in event processing efficiently and effectively, I propose an approach to event processing that is based on three main models: the approximate semantic event matching model, the thematic event processing model, and the dynamic native event enrichment model. This chapter gives an overview of these models. It also aims at developing a set of hypotheses to address the research questions of this work. In order to develop the hypotheses, the approach is decomposed analytically into four conceptual elements: subsymbolic distributional event semantics, free event tagging, dynamic native event enrichment, and approximation. More discussion and evaluation of the models, the elements, and the rationale behind the derived hypotheses is the focus of Chapter 5, Chapter 6, and Chapter 7.

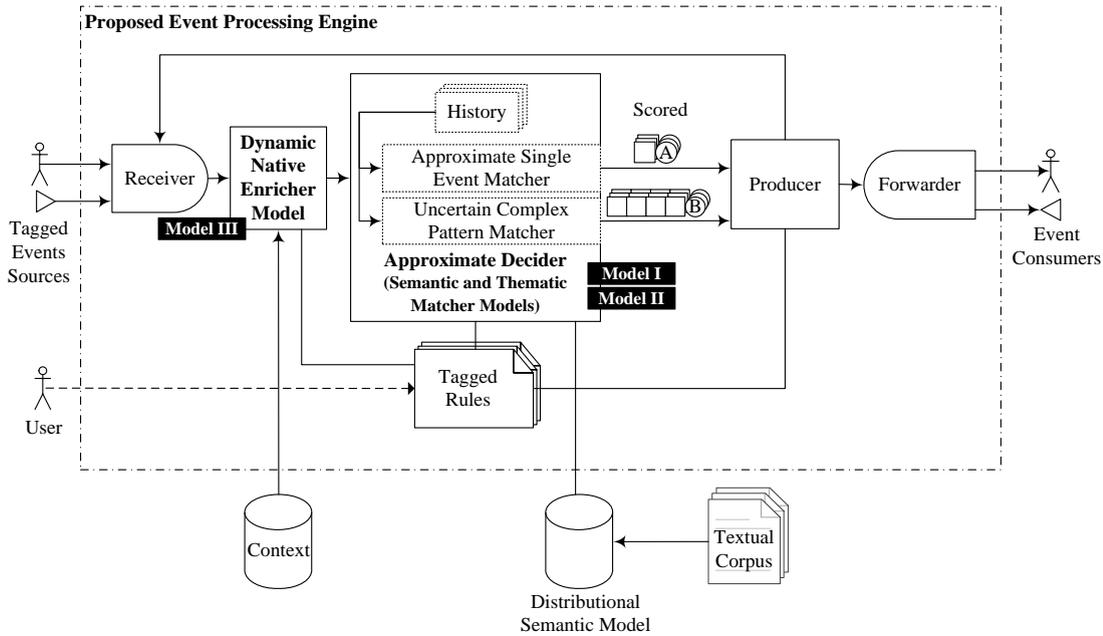


FIGURE 4.1: The main models of the proposed approach

An overview of the main models is discussed in Section 4.2. Section 4.3 provides an overview of the main elements and their relationships to the requirements and research questions and the main models. The scope of the proposed model is discussed in Section 4.4. A summary of the chapter is presented in Section 4.5.

## 4.2 Main Models

The proposed approach is constructed from three main models as shown in Figure 4.1 and outlined in the following sections.

### 4.2.1 The Approximate Semantic Event Matching Model

This model tackles the requirements of efficient and effective loose semantic coupling. It is illustrated by Model I in Figure 4.1. It extends the current event processing paradigm through the following:

- Rules are equipped with the *tilde*  $\sim$  semantic approximation operator so users can express their delegation to the event engine to match similar or related event terms to the term used in a subscription. The background semantic model for

approximation is a statistical model built from co-occurrences of terms in a large corpus of plain text documents. For instance, the following subscription tells the event engine to match it to events generated from a ‘*laptop*’ or a similar device, with the term ‘*office*’ used or related terms such as ‘*room*’ or ‘*zone*’.

```
{type= increased energy usage event,
device= laptop~,
office~= room 112}
```

- The single event matcher is equipped with matching and mapping algorithms to detect events semantically relevant to approximate subscriptions. For instance, let an event of *increased energy consumption* be represented as follows:

```
{type: increased energy consumption event,
measurement unit: kilowatt-hour,
device: computer,
office: room 112}
```

The most probable mapping, or the top-1 mapping, of this event to the previous subscription is generated as a probable scored result. It can be described as follows:

$$\sigma^* = \{(\mathbf{type} = \mathbf{increased\ energy\ consumption\ event} \leftrightarrow \text{type:increased energy usage event}),$$

$$(\mathbf{device} \sim = \mathbf{laptop} \sim \leftrightarrow \text{device:computer}),$$

$$(\mathbf{office} = \mathbf{room\ 112} \leftrightarrow \text{office: room 112})\}$$

- The complex pattern matcher can then perform a probabilistic reasoning to deduce the probabilities of occurrences of the derived events in the action parts of the complex rules.

This model has been presented in the ACM Transactions on Internet Technology Journal (ToIT 2014) [153], and the ACM International Conference on Distributed Event-Based Systems (DEBS 2012) [155]. It will be discussed in detail in Chapter 5.

#### 4.2.2 The Thematic Event Matching Model

This model tackles the requirements of efficient and effective loose semantic coupling. It is illustrated by Model II in Figure 4.1. It suggests associating free tags, called themes

or thingsonomies, that describe the themes of types, attributes and values in events and subscriptions, and clarify their meanings. For instance, the previous *increased energy consumption* event is associated with tags as follows:

{appliances, building}

These tags help to disambiguate the meaning of terms in the event such as ‘*energy*’ and ‘*office*’ and get them closer to the energy management domain in smart buildings. Thematic events can more easily cross semantic boundaries as: (1) they free users from needing a prior semantic top-down agreements, and (2) they carry approximations of events meanings composed of payloads and theme tags which, when combined, carry less semantic ambiguities. An approximate matcher exploits the associated thematic tags to improve the quality of its uncertain matching of events and subscriptions.

This model has been presented in the IEEE Internet Computing (2015) [151], and the International ACM/IFIP/USENIX Middleware Conference (Middleware 2014) [152]. It will be discussed in detail in Chapter 6.

### 4.2.3 The Dynamic Native Event Enrichment Model

This model tackles the requirements of efficient and effective loose pragmatic coupling. It is illustrated by Model III in Figure 4.1. In this model, events are assumed incomplete under an open world assumption. Enrichment is the process of complementing events from background knowledge. The model uses four aspects for event enrichment: determination of the enrichment source, retrieval of information items from the enrichment source, finding complementary information for an event in the enrichment source, and fusion of complementary information with the event.

The model proposes that the enrichment logic is described using a set of declarative language constructs similar to the ones used currently for matching purposes. Four language clauses that are mapped to the four enrichment aspects are proposed: ENRICH FROM, RETRIEVE BY, FIND BY, and FUSE BY. All the enrichment clauses are described by the event consumer. The resulting subscription, which contains enrichment and matching elements, is called a unified subscription. For instance, the following unified subscription tells the engine to explore a Linked Data graph by a method called

Spreading Activation to enrich an RDF event with triples that can be missing such as the *‘floor’* in the building where it was generated.

```
ENRICH FROM <www.myenterprise.org>
RETRIEVE BY ‘DEREF’
FIND BY ‘Spreading Activation’
FUZE BY ‘UNION’
{?event rdf:type ont:EnergyConsumption.
?event (?p){3} building:SecondFloor.}
```

This model has been presented in the ACM International Conference on Distributed Event-Based Systems (DEBS 2013) [154], and the International Workshop on Semantic Sensor Networks (SSN 2011) at the International Semantic Web Conference (ISWC 2011)[156]. It will be discussed in detail in Chapter 7.

### 4.3 Main Elements

The approach I propose in this work can be conceptually decomposed into four main elements. Elements are jointly used throughout the concrete models of the previous section and experiments discussed in Chapters 5, 6, and 7. Elements form the basis for formulating the hypotheses of this thesis. These elements are outlined in the following sections:

#### 4.3.1 Subsymbolic Distributional Event Semantics

This element stems from the need for loosening the semantic coupling between event producers and consumers. Assuming that semantic coupling can be quantified by the number of mappings between symbols, i.e. terms, and meanings, then a semantic model that condenses these mappings can be very useful. Ontological models require granular agreements on the symbol-meaning mappings, that is proportional to the number of symbols. However, distributional vector space semantics leverage the statistics of terms co-occurrence in a large corpus to establish semantics [42]. For instance, the terms *‘power’* and *‘electricity’* would frequently co-appear in an energy management domain

corpus. Thus, they can be assumed related or similar, and this can be leveraged for energy event matching. Using such a model leaves event producers and consumers to loosely agree on the corpus as a representative of their common knowledge and decrease the need for granular agreements on every individual term of the domain.

The research question mainly addressed by the subsymbolic distributional event semantics element is research question *Q1*:

*Q1.* The first research question is concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rules and their meanings but rather a form of statistical loose agreements on the meanings. The question is how to achieve timely event matching with high true positives and negatives in such a loosely semantically coupled environment?

The formulated hypothesis that underlies the proposal of this element and its use within the proposed models is as follows:

*H1.* Subsymbolic distributional event semantics decreases the cost needed to define and maintain rules with respect to the use of terms, and to build and agree on an event semantic model more than symbolic semantic models; and at the same time it can achieve timely event matching with high true positives and negatives of magnitudes comparable to that of event processing based on semantic models.

More discussion about this element and the rationale behind this hypothesis will be detailed in Section 5.4. This hypothesis is the subject of investigation with the approximate semantic event matching model in Chapter 5, and the thematic event matching model in Chapter 6.

### 4.3.2 Free Event Tagging

This element stems from the need to enable event processing within a loosely coupled model in an effective and efficient way, and allow users to adapt the conveyed events' meanings in different domains and situations. Free tagging of events and subscriptions

do not introduce any coupling components between participants as suggested by top-down fixed taxonomies. This element builds on the success of free tagging, known as folksonomies, within social media research [43]. For instance, the term ‘energy’ when used in an event tagged by the tags {‘building’, ‘appliance’} helps the matcher distinguish the meaning of ‘energy’ and associate it with the domain of power management, rather than associating it with the domain of sport or diet for example.

The research question mainly addressed by the free event tagging element is research question *Q1*:

*Q1.* The first research question is concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rules and their meanings but rather a form of statistical loose agreements on the meanings. The question is how to achieve timely event matching with high true positives and negatives in such a loosely semantically coupled environment?

The formulated hypothesis that underlies the proposal of this element and its use within the proposed models is as follows:

*H2.* Free tagging of events and subscriptions does not add to the cost of defining and maintaining rules with respect to the use of terms, and the cost of building and agreeing on an event semantic model required by subsymbolic event semantics; and at the same time it can achieve timely event matching with high true positives and negatives more than event processing based on non-tagged subsymbolic event semantics.

More discussion about this element and the rationale behind this hypothesis will be detailed in Section 6.3. This hypothesis is the subject of investigation with the thematic event matching model in Chapter 6.

### 4.3.3 Dynamic Native Event Enrichment

This element stems from the need for loosening the pragmatic coupling between event producers and consumers in an effective and efficient way. Such a coupling is caused by

mutual agreements on contextual information as in dedicated enrichers. It can be reduced by allowing the event processing systems to discover contextual data dynamically. For instance, an *energy consumption event* could include information about the consuming ‘*device*’ and its ‘*power consumption*’. The event engine shall be able to dynamically look up the device in a building management system database to get information about the ‘*room*’ and ‘*floor*’ where the device exists. Thus, events are assumed to be incomplete, and contextual data is dynamically added through an enrichment process that is moved to the core of the event engine.

The research question mainly addressed by the dynamic native event enrichment element is research question *Q2*:

*Q2.* The second research question is concerned with the case when event producers and consumers do not have equal assumptions on the amount of contextual information included in events and how much they are complete with respect to evaluating some consumers’ rules. The question is how to complement events with context at high precision and completeness needed to meet consumers expectations in such a loosely contextually coupled environment?

The formulated hypothesis that underlies the proposal of this element and its use within the proposed models is as follows:

*H3.* Dynamic native event enrichment decreases the cost needed to define and maintain the context parts of rules, and to agree on contextual data that is needed in events more than dedicated enrichers; and at the same time it can achieve high precision integration of event context with high completeness of events comparable to that of event processing based on dedicated enrichers.

More discussion about this element and the rationale behind this hypothesis will be detailed in Section 7.3. This hypothesis is the subject of investigation with the dynamic native event enrichment model in Chapter 7.

#### 4.3.4 Approximation

This element stems from the realization that loosening the coupling between event producers and consumers at the semantic and pragmatic levels introduces uncertainties to the engine. Uncertainty results from not exactly knowing which event's tuples shall be mapped to which subscription's tuples, and which information can be assumed in an event that is incomplete. For instance, with the loose agreements on terms semantics, there are various possible mappings between an event and a subscription such as:

$$\sigma_1 = \{(\mathbf{device} = \mathbf{laptop} \leftrightarrow \text{device:computer}),$$

$$(\mathbf{room} = \mathbf{room 112} \leftrightarrow \text{office: room 112})\}$$

$$\sigma_2 = \{(\mathbf{device} = \mathbf{laptop} \leftrightarrow \text{office: room 112}),$$

$$(\mathbf{room} = \mathbf{room 112} \leftrightarrow \text{device:computer})\}$$

Each mapping has a different probability that reflects the uncertainty of the matching. The same applies to the uncertainty about which tuples complement an event. Approximation at the core of the event processing engine can tackle uncertainties and complement the elements mentioned earlier.

The research questions mainly addressed by the approximation element are both research questions *Q1* and *Q2*:

*Q1.* The first research question is concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rules and their meanings but rather a form of statistical loose agreements on the meanings. The question is how to achieve timely event matching with high true positives and negatives in such a loosely semantically coupled environment?

*Q2.* The second research question is concerned with the case when event producers and consumers do not have equal assumptions on the amount of contextual information included in events and how much they are complete with respect to evaluating some consumers' rules. The question is how to complement events with context at high precision and completeness needed

to meet consumers expectations in such a loosely contextually coupled environment?

The formulated hypothesis that underlies the proposal of this element and its use within the proposed models is as follows:

*H4.* Approximate event processing can operate in event environments with low-cost agreements on event semantics and pragmatics more than exact event processing; and at the same time achieve timely event matching with high true positives and negatives, and high precision integration of event context with high completeness of events, comparable to that of event processing based on exact models.

More discussion about this element and the rationale behind this hypothesis will be detailed in Sections 5.5 and 7.4. This hypothesis is the subject of investigation mainly with the approximate semantic event matching model in Chapter 5, along with further tests with the thematic event matching model in Chapter 6 and the dynamic native enrichment model in Chapter 7.

#### 4.3.5 Elements within the Event Flow Functional Model

The main elements of the proposed approach can be unified and fit into the event processing functional model discussed in Section 2.5.2. The elements work together, along with non-impacted components, to fulfil the role of an event processing engine along with the requirements tackled in this work. Figure 4.2 contrasts the changes introduced by the proposed approach, with a typical event processing model as discussed previously in Section 2.5.2.

Figure 4.2 shows how each element fits into the model as follows:

- **Elm1** *Subsymbolic Distributional Event Semantics*. The actual distributional semantic model could be built outside of the event processing engine by indexing a textual corpus. The resulting model forms the basis to compare any two strings in events and subscriptions, as they get decoded into their vector representations. Vectors form the basis for distance and similarity measure.

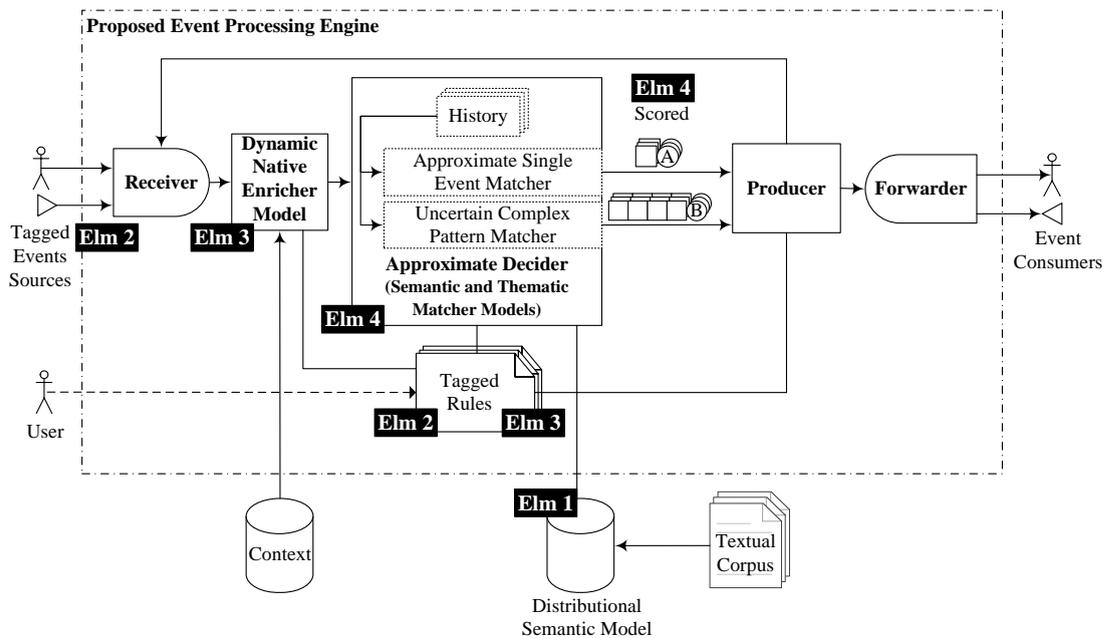


FIGURE 4.2: The main elements within the proposed event processing approach

- **Elm2** *Free Event Tagging*. Events flowing from event sources, and subscriptions get tagged by users before they are considered for matching. Users use free tags to enhance events and subscriptions and improve their interpretation by the matcher.
- **Elm3** *Dynamic Native Event Enrichment*. A new functional component, the *Dynamic Enricher*, is added to the model. Enrichment guidance elements are added to the subscriptions to identify parts like the enrichment source, the retrieval, search, and fusion mechanisms of contextual information with events. The enricher is also guided by the matching parts of subscriptions. Events get enriched before being passed to the decider.
- **Elm4** *Approximation*. Events are matched in the decider against subscriptions. The decider is now approximate, and the result of matching are scored events that signify their relevance to each subscription. The decider makes use of the semantic model, the tags, and the enriched complemented versions of the events.

Table 4.1 summarizes the mapping between the elements, requirements, and research questions, with the proposed models of approximate semantic matching, thematic event matching, and dynamic native event enrichment that constitute the event flow model.

TABLE 4.1: Elements and Models of the Proposed Approach with Respect to Requirements and Research Questions

		<b>The approximate semantic event matching model</b>	<b>The thematic event matching model</b>	<b>The dynamic native event enrichment model</b>
<b>Elements</b>	Subsymbolic distributional event semantics	×	×	×
	Free event tagging		×	
	Dynamic native event enrichment			×
	Approximation	×	×	×
<b>Requirements</b>	<i>R1.</i> Loose semantic coupling	×	×	
	<i>R2.</i> Loose pragmatic coupling			×
	<i>R3.</i> Efficiency	×	×	×
	<i>R4.</i> Effectiveness	×	×	×
<b>Research Questions</b>	<i>Q1.</i> The first research question is concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rules and their meanings but rather a form of statistical loose agreements on the meanings. The question is how to achieve timely event matching with high true positives and negatives in such a loosely semantically coupled environment?	×	×	
	<i>Q2.</i> The second research question is concerned with the case when event producers and consumers do not have equal assumptions on the amount of contextual information included in events and how much they are complete with respect to evaluating some consumers' rules. The question is how to complement events with context at high precision and completeness needed to meet consumers expectations in such a loosely contextually coupled environment?			×

## 4.4 Scope

Beside the functional model, the proposed elements can be localized with respect to the other models of Cugola and Margara [8] for the purpose of scoping. The following discussion complements that on the scope of the work outlined in Section 2.11 as follows:

- *The processing model.* This work follows a single selection policy and a selected consumption policy. Single events get enriched between the receiver and the decider before they are considered for matching.
- *The deployment model.* This work assumes a distribution of the participants of an event processing environment with a centralized deployment model of the engine. The subsymbolic distributional semantic model is established within the distributed environment. Low requirements regarding the effort to establish such a model include the adoption of a mediator distributed comprehensive corpus, such as Wikipedia, and the use of free tagging.
- *The interaction model.* This work follows a push-based model of interaction. The introduction of explicit context and semantic relatedness services lead to some forms of *pull-based* interaction between the engine from one side, and the enrichment source and the semantic measure on the other side. Nonetheless, this pull-based behaviour is secondary, and the prime interaction between event producers and consumers is push-based.
- *The data model.* This work can be generalized to include various data models. Nonetheless the concrete model of attribute-value records have been used for experimentation in Chapters 5 and 6, while a graph model is used in Chapter 7. The use of free tagging introduces changes to the event model, where thematic tags are associated with the event payload that conforms to a classical model such as attribute-value records as shown in Chapter 6.
- *The time model.* As this work is scoped to single event matching, no partial or total temporal order such as happened-before relationships are considered. Nonetheless, as single event matching is probabilistic due to approximation, there is a need to handle probabilities propagation during complex event pattern matching. This is partially addressed in this work but is kept out of scope.

- *The rule model.* Rules considered in this work are detection rules. The adoption of an approximate model leads to the awareness of uncertainty in semantics and pragmatics and the possibility to describe rules as supporting uncertainty.
- *The language model.* The language considered in this work is a detection language, with a single-item selection operator. The use of unified enrichment subscriptions introduces changes to the subscription model, where enrichment language elements are associated with the subscription expression that conforms to a classical model such as attribute-value records as shown in Chapter 7.

## 4.5 Chapter Summary

This chapter proposed an approach to loosen semantic and pragmatic coupling based on three main models: approximate semantic event matching, thematic event matching, and dynamic native event enrichment. The models have been conceptualized in four elements that constitute the hypotheses underlying the models: subsymbolic distributional event semantics, free event tagging, dynamic native event enrichment, and approximation.

The rationale for subsymbolic distributional event semantics is that symbolic models require granular agreements on the symbol-meaning mappings which imply coupling, while subsymbolic distributional semantics leverages the statistics of terms co-occurrence in a large corpus, leading to relaxed semantic agreements. The grounds for free event tagging stems from the fact that it does not introduce any coupling components between participants as opposed to top-down imposed fixed taxonomies.

The rationale for dynamic native event enrichment stems from the acknowledgement that events are incomplete, and contextual data can be dynamically added through an enrichment process that is moved to the core of the event engine. Finally, the rationale for approximation stems from the realization that loosening semantic and pragmatic coupling introduces uncertainties to the event processing engine, thus approximation is needed to enable the other elements of the proposed approach.

The chapter mapped the proposed models to the elements, the requirements, and the research questions. The models, elements, and hypotheses are discussed in more detail and evaluated within Chapter 5, Chapter 6, and Chapter 7.

## Chapter 5

# The Approximate Semantic Event Matching Model

“An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.”

— John Tukey

### 5.1 Introduction

In Chapter 4 a set of hypotheses has been formulated to answer the research questions based on a set of four elements that are combined to form the proposed approach. This Chapter 5 tackles mainly research question *Q1* that states the following:

*Q1.* The first research question is concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rules and their meanings but rather a form of statistical loose agreements on the meanings. The question is how to achieve timely event matching with high true positives and negatives in such a loosely semantically coupled environment?

This chapter tests the hypothesis  $H1$  and the semantic part of hypothesis  $H4$  which are formulated as follows:

- $H1$ . Subsymbolic distributional event semantics decreases the cost needed to define and maintain rules with respect to the use of terms, and to build and agree on an event semantic model more than symbolic semantic models; and at the same time it can achieve timely event matching with high true positives and negatives of magnitudes comparable to that of event processing based on semantic models.
- $H4$ . Approximate event processing can operate in event environments with low-cost agreements on event semantics and pragmatics more than exact event processing; and at the same time achieve timely event matching with high true positives and negatives, and high precision integration of event context with high completeness of events, comparable to that of event processing based on exact models.

To test the hypotheses, this chapter constructs a model that realizes the elements of subsymbolic distributional semantics and approximation. Section 5.2 outlines the proposed model. This model has been mainly presented in the ACM Transactions on Internet Technology Journal (ToIT 2014) [153], and the ACM International Conference on Distributed Event-Based Systems (DEBS 2012) [155].

Section 5.3 presents an overview of the distributional semantic model that is used. The background of semantics and approximation and a discussion of the rationale behind the hypotheses about the elements of subsymbolic distributional semantics and approximation are detailed in Section 5.4 and Section 5.5 respectively. The concrete approximate semantic event matching model is discussed afterwards. Section 5.6 describes the event flow model and the changes which affect it. The event, language, and matching models are discussed in Section 5.7, Section 5.8, and Section 5.9 respectively.

The constructed model of event matching is empirically validated where Section 5.10 details the evaluation methodology and results. This chapter shows that the formulated hypotheses  $H1$  and  $H4$  are valid. Thus, the elements of subsymbolic distributional semantics and approximation can answer the research question and consequently can address the requirements of effective and efficient event processing that is loosely coupled in semantics. This chapter is summarized in Section 5.11.

## 5.2 Overview

The proposed model loosens semantic coupling by making participants agree on a topic or set of topics that are represented as a large corpus of text. The corpus is then used to build a distributional semantic model to derive semantic similarity and relatedness. The model also introduces the *tilde*  $\sim$  semantic approximation operator to the event processing language. For example, a subscription to energy events such as the one required in Section 2.2 can be expressed as  $\{\mathbf{type}=\text{heater energy consumption increased } \sim\}$  to let the engine match events of the mentioned type or any other type semantically related to it. The proposed model is realized based on:

- The use of distributional semantics relatedness measures, such as the Wikipedia-based Explicit Semantic Analysis (ESA) to parametrize the *tilde*  $\sim$  operator.
- A matching model rooted in uncertain schema matching related work [173].
- A probability model for uncertainty management.

## 5.3 Distributional Semantics as a Loosely Coupled Event Semantic Model

Distributional semantics is based on the hypothesis that similar and related words appear in similar contexts as discussed in Section 5.4. Distributional models are useful for the task of assessing semantic similarity and relatedness between terms. A semantic measure is a function  $sm$  that quantifies the similarity/relatedness between two terms. Typically  $sm$  has its values in  $\mathbb{R}$ . Distributional models can be constructed automatically from statistical co-occurrence of words in a corpus of documents, e.g. the measure based on the Explicit Semantic Analysis (ESA) of the Wikipedia corpus.

The IoT event cloud would include events from various domains which suggests that domain-agnostic measures have a potential for IoT. The discussion in this chapter is scoped to the domain-agnostic distributional semantic measure  $esa$  [42] constructed from the Wikipedia corpus as of 2013 <sup>1</sup>. That is due to its relative ease of construction as it is based on statistical analysis of unstructured document corpus. However the model

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Wikipedia:Database\\_download](http://en.wikipedia.org/wiki/Wikipedia:Database_download)

is generic and suitable for other measures as well. Semantic measures are assumed to be external services to the event engine, and they are constructed independently. This assumption simplifies the interface between the event engine and the service and makes the embedding of different services relatively easy.

In the next two sections, I discuss in detail the elements of subsymbolic distributional event semantics and approximation. The discussion gives a background of what semantics is and the various models of semantics. It also gives a background of approximation and its role within computing systems. The discussion in Section 5.4 and Section 5.5 motivates the rationale behind hypothesising that these elements can answer the research question and meet the requirements of efficient and effective loose semantic coupling in event systems. The approximate semantic event processing model is discussed afterwards from Section 5.6 to Section 5.10.

## 5.4 Subsymbolic Distributional Event Semantics

As this work addresses the problem of coupling in semantics, a semantic model is at the core of the approach. Thus, the first element of the proposed approach tackles the aspect of semantics.

### 5.4.1 Semiotic Systems for Symbols and Meanings

Semantics generally refers to a relationship between two spaces (or worlds or sets): the meanings, and the symbols. As put by Gärdenfors [139, p. 151]:

“Semantics concerns the relation between the words or expressions of a language and their meaning.”

This view is mostly visible in the field of semiotics [143] where the focus is on signs and sign systems. Two main models traditionally frame the science of semiotics: the Saussurean model [176], and the Peircean model [177]. In semiotics, the sign is the whole of the symbol and the meaning. Saussure divides a sign into two parts: the signifier, or the symbol, and the signified. For example, Figure 5.1 illustrates a sign that is composed

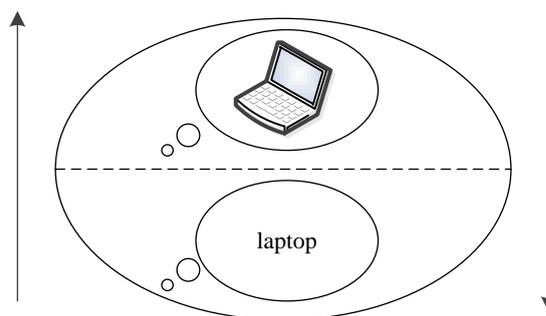


FIGURE 5.1: A sign of two parts: a signifier and a signified

of the concept laptop, represented by the top image, and its signifier, the English word ‘*laptop*’ at the bottom.

Peirce [177] on the other hand provides a triangular model of a sign. For Peirce the sign consists of three components: the representation, the interpretant, and the object. The representation is the form that the sign takes, which is analogous to the signifier in the Saussurean model. The interpretant is the sense, which is similar to the signified in the Saussurean model. The interpretant can also refer by itself to an object that could be a material aspect that exists in reality.

Despite the fact that Saussure suggests a dyadic model, and Peirce suggests a triadic model, they both recognize the signifier or the representation, that is the symbol, and the signified, that is the meaning. For Saussure, the signified is, in fact, a mental representation, which is the same meant by Peirce’s interpretant. Saussure recognizes that the mental representation could stand for something in reality, but does not dedicate a part of the sign for it explicitly. On the other hand, Peirce dedicates a part of his sign model to that as represented with the object part [143, p. 13–35].

What matters in this discussion is the distinction between two sets: the symbols and the meanings. The two sets only meet within signs. Signs are then the realization of semantics. They are the mapping relations mentioned by Gärdenfors. This leads to a conclusion that any model of semantics needs to provide or adopt a model for meaning representation, as well as a model for mapping.

Various proposals have been put towards representing meanings. I adopt here a classification of these proposals based on the three-level framework described by Gärdenfors [139, p. 33–58]. The purpose of this classification is to draw fair comparisons between meaning models. Gärdenfors bases this categorization on previous similar works by

Harnad [178], Mandler [179], and Radermacher [180]. He recognizes three main levels of representation: symbolic, conceptual, and subconceptual.

The domain of meanings can be classified into objects, properties, and concepts. Objects are individuals like a specific laptop used by Alice. Properties are a “way of abstracting away redundant information about objects” [139, p. 59]. For instance, Alice’s laptop is ‘*black*’ which is a property. Concepts are the most generic form of objects and properties. A concept clusters similar properties and objects such as the concept ‘*Laptop*’. I discuss those notions with respect to the three levels of meaning representation in the following sections.

### 5.4.2 Symbolic Representation of Meaning

The key principle of symbolism is that information is represented by *symbols*, and the processing of information is by definition a *manipulation of symbols* through *rules* [139, p. 35–36]. This symbolic paradigm sticks to a symbolic level syntax and semantics of a combinatorial nature as discussed for example by Foder and Pylyshyn [181]:

“While both Connectionist and Classical architectures postulate representational mental states, the latter but not the former are committed to a symbol-level of representation, or to a ‘language of thought’: i.e., to representational states that have combinatorial syntactic and semantic structure.”

The symbolic approach to meaning is widely adopted by computing communities such as Artificial Intelligence (AI).

#### 5.4.2.1 Computationalism and the Symbolic Paradigm

A key testimony to the dominant role of the symbolic paradigm in computing systems comes from Newell and Simon [182]:

“One of the fundamental contributions to knowledge of computer science has been to explain, at a rather basic level, what symbols are. This explanation is a scientific proposition about Nature. It is empirically derived, with

a long and gradual development. Symbols lie at the root of intelligent action, which is, of course, the primary topic of artificial intelligence. For that matter, it is a primary question for all of computer science.”

Symbols can be gathered into sentences of a *language of thought*. What a sentence means is a *belief* of an agent. Various beliefs are connected by logical or inferential relations such as first-order logic in AI. Thus, meanings are purely the result of logical, syntactic relations of symbols, rather than the states to which they refer.

The tradition of the symbolic paradigm of meaning representation is the signature of computationalism in general, not only of AI. It also extends to other areas such as databases and event-based systems where semantic assumptions are derived from those of databases. For instance, in the database community, the relational model has been widely adopted [183]. Codd [183] proposes “a relational view of data” such that:

“The term relation is used here in its accepted mathematical sense. Given sets  $S_1, S_2, \dots, S_n$  (not necessarily distinct),  $R$  is a relation on these  $n$  sets if it is a set of  $n$ -tuples each of which has its first element from  $S_1$ , its second element from  $S_2$ , and so on.”

Elements of  $S_1, S_2, \dots, S_n$  are referred to as constants, which is the database name for symbols.

Schema is handled similarly, and here I quote Codd [183]:

“The significance of each column is partially conveyed by labelling it with the name of the corresponding domain.”

Where labels are the synonyms of symbols for a schema. I argue that the definition of relation be none but a logical predicate, a common tenet of symbolism.

Another indication of the symbolic paradigm in databases comes from the Unique Name Assumption (UNA) which is adopted in databases [184]:

“The unique-name assumption which says that two distinct constants (either atomic values or objects) necessarily designate two different objects in the universe.”

I argue that by projecting this definition on the semiotics model, the result is an isomorphic mapping between the set of signifiers and the set of signifieds. Thus, meanings are explicitly mapped one-by-one to symbols, which reflects a symbolic paradigm.

#### 5.4.2.2 Symbolic Semantics

Due to the tight relationship between symbols and meanings in the symbolic paradigm, it is difficult to separate a meaning model from a model of semantics. Thus, discussion of a representation of meanings in this paradigm is typically a part of an overall semantics program. There are mainly three directions to tackling the fundamental question of what a property is [139, p. 60–62]:

- *Extensional Semantics* where a property is identified by the set of objects that have the property. For instance, ‘*black*’ is the set of all objects of the colour *black*. In the model theory of Tarski [185], this is done through a mapping between a language and a model structure that is said to represent *the world*.
- *Intensional Semantics* alters the concept of one world to the case of multiple *possible worlds*. This is done to handle the case of the so-called intentional properties such as *small*, which can not be thought of simply as a set of small objects. Thus, the basic elements of semantics become the objects in a set of possible worlds. Truth functions map a language to a subset of these worlds to provide an interpretation of the language. This model has been developed by Kanger [186], Kripke [187], Montague [188], and others.
- *Situation Semantics* uses a one world model, but instead of truth functions from symbols or sentences to possible worlds, it uses a polarity function from symbols or sentences to a subset of the world, called *situation*. This model has been largely developed by Barwise [189].

Properties and concepts are not distinct in symbolism. Thus a discussion on properties also applies to concepts. In ontologies, for instance, properties and concepts are described using *TBox* statements. Objects, on the other hand, are described using *ABox* statements that are compliant with the TBox terminological description.

### 5.4.2.3 Limitations of the Symbolic Paradigm

The classical symbolic approaches to meaning and semantics have been criticized from various aspects. For instance, Gärdenfors [139, p. 37–40, 62–66] provides the following critics:

1. The definition of properties and concepts as functions is highly counterintuitive.
2. The definition does not explain how a person can perceive two objects to have the same property or two properties to be similar.
3. The traditional symbolic approaches to semantics and meaning cannot account for the problem of inductive reasoning.
4. The model-theoretic definition of properties and concepts does not, in fact, work as a theory of meaning as investigated by Putnam [190, p. 22–48].
5. The *frame problem* that states that representing all necessary knowledge about the world in a symbolic way requires a combinatorial explosion of logical axioms and inferences.
6. The *development problem* that results from the unnatural or simple way to change predicates at the symbolic level as a cognitive system evolves in time.
7. The *symbol grounding problem* that states that, in the symbolic paradigm, the meanings of symbols are actually grounded in the symbols themselves [191].

I add to this critique a perspective derived from the requirement of loose semantic coupling in event-based systems. The problem as I see with the symbolic approach to meaning is that it does not largely separate the symbolic level from the meaning level, as also manifested by the symbol grounding problem. Let us assume that the agents who exchange information are symbolic agents, such as event agents programmed by humans who are symbolic too. When agents need to agree on the meanings, which are the essence of information exchange, they have to agree on symbols due to the tight relationship between meanings and symbols.

Agreeing on symbols is granular and an extremely costly process, thus it does not qualify for the loose semantic coupling requirement. Fundamentally this is a result of the lack of

a natural account for similarity in the symbolic level models of meanings. The existence of similarity, and more generally topological relationships, between meanings can lower the amount of information that two parties need to agree on. This issue is well tackled at the conceptual level of meaning models as discussed in Section 5.4.3.

Furthermore, the combinatorial nature of the frame problem is an inefficient way to loosen semantic coupling or to create an event system efficiently, contradicting the Requirements *R1* of loose semantic coupling and *R3* of efficiency. The development problem can also be manifested in an unsuitable manner to reflect meanings in a changing event environment, contradicting the Requirement *R4* of effectiveness.

### 5.4.3 Conceptual Representation of Meaning

At this level comes various alternative meaning models that fundamentally leverage the topological features of meanings. Here lies a class of approaches which depart from the symbolic tenets, but at the same time do not go very deep to a neural level where explicit explanations of the models can be lost. What is defining in these approaches is a geometrical nature of the meaning space. In such a geometry, distances and closeness between meanings can be established.

#### 5.4.3.1 Conceptual Spaces

An example of conceptual representations of meaning is the Conceptual Spaces proposed by Gärdenfors [139]. He states the central principle behind his proposal as follows:

“The epistemological role of the theory of conceptual spaces to be presented here is to serve as a tool in modelling various *relations* among our experience. ” [139, p. 5]

So, for Gärdenfors, the notion of *similarity* is a central motivation behind geometrical models of meanings.

Gärdenfors’ conceptual spaces start from the observation that concepts are not independent from each others, but rather are structured into *domains*, e.g. the domain of colours, the spatial domain, etc. Conceptual spaces are then built up from *quality*

*dimensions* that serve the purpose of building the domains. For instance, the colours domain can be built up from three dimensions: *hue*, *chromaticness* or saturation, and *brightness*.

Dimensions do not have to follow a classical Euclidean geometry. They could rather organize in a form that ideally fits better with human cognition, e.g. perception in the case of colours. Spaces formed by quality dimensions shall still have some basic topological features such as betweenness, and the existence of a distance function in the space (i.e. being a metric space). For example, the *hue* dimension of colours is, in fact, a circle. *Yellow* is closer to *Green* on that circle, rather than to *Violet*. The overall domain of colours becomes a spindle, rather than a classical three-dimensional Euclidean space [192].

A *natural* property in conceptual spaces is a convex region in a domain. For instance, the property *Green* is a three dimensional region in the colour spindle. Given two points within this *Green* area *a* and *b*, any point that lies *between a* and *b* is in the *Green* area too. Betweenness here does not have to follow a straight line according to the Euclidean geometry, but rather can be a “*curved*” line according to the defined domain geometry.

A natural concept in conceptual spaces is a set of regions in a number of domains, with an assignment of salience weights to the domains. For instance, the concept of a *Laptop* is a collection of possible regions from the colour domain such as *Black*, *White*, and *Silver*; a region of the space domain possibly of possible sizes of laptops, etc. Some domains can be weighted higher than others. Given this definition, some concepts are closer to each other than others. For example, the concept of a *Laptop* is closer to the concept of a *Mobile Phone* than to the concept of a *Car*. That can be derived geometrically from the closeness between regions in each domain.

Objects are points in the conceptual space. For instance, *Alice’s laptop* is one with a specific colour like black, a specific size, etc. This point lies within the region of the concept *Laptop*. Thus, some objects can be closer to others based on a geometrical basis. For example, Alice’s laptop and Bob’s laptop have the same screen size, are yellow and green, have same the CPU and memory size, and have a power consumption of 30 and 35 watts respectively. Geometrically they can be closer than Alice’s laptop and Dan’s laptop, which is black, with higher CPU power, and consumes 60 watts of power.

Some objects can be more central within a concept region, i.e. more *typical*. This provides a very close connection with the prototype theory [193] widely used in cognitive science. Nonetheless, conceptual spaces provide a more natural way to represent closeness and similarity between meanings.

Given this theory of meaning, the question arises of how one can build a theory of semantics based on that. In semantics, the concern is to provide a mapping with symbols and expressions of a language  $\mathcal{L}$ . Gärdenfors [139, p. 167–176] suggests a mapping between various types of a language’s expressions and the conceptual spaces elements. His main thesis is that:

“Basic lexical expressions in a language are represented semantically as natural concepts. ” [139, p. 167]

A key point that I raise here is that geometrical models of semantics, such as conceptual spaces, emphasizes a *lexical semantics*. That is, the main target of semantic mapping is the lexicon or the symbols of a language. More complex syntactic structures such as sentences have been the focus of truth conditions types of semantics, manifested by the symbolic approaches to meaning shown in Section 5.4.2.

I argue that in events models, the syntax is more controlled and of less importance rather than when dealing with natural languages. For instance, in an event model such as the attribute-value maps, the focus in this work, the actual language of the event is reduced to the set of lexicons or terms used as attributes and values. Thus, geometrical spaces of meanings are suitable as no more compositionality requirements are assumed.

Conceptual models of semantics that leverage topology and similarity of meanings, and thus of terms, can lower the dimensionality of the agreement problem. Let us assume that two cognitive agents, such as event publishers and consumers, agree on a conceptual space. If the publisher uses the term ‘*laptop*’ in an event, while the consumer expects the term ‘*device*’ then the consumer can leverage the similarity between both terms due to the existence of a mediator space and still be able to establish matching between the event and the subscription. Thus, the agreement problem has been considerably lowered to a loose agreement on the space rather than a granular agreement on every term that could be used. Within this sense, the conceptual models of meanings meet the Requirement *R1* of loose semantic coupling.

I redefine event matching as an estimation of closeness, or similarity, between the event and the subscription. This similarity is broken down to a similarity between two attributes or two values, which are points in a conceptual space. Thus, a model of meaning derived from geometrical spaces, which support similarity naturally, makes the best fit for the event matching problem. Measuring closeness can be reduced to measuring distance in the space, which is efficient computationally on mathematical grounds. From this perspective, the conceptual models of meanings meet the requirements  $R3$  and  $R4$  of efficiency and effectiveness.

#### 5.4.3.2 Statistical Distributional Semantics

While I agree on the importance of similarity as the basis for meaning and semantic models for event matching, I argue that the proposal of Gärdenfors [139] be computationally challenging. The main problem with Gärdenfors' conceptual spaces is that building quality dimensions, and agreeing on them can be hard to achieve at large scales. It can be reduced to the problem of agreement on symbolic concepts. Thus, the conceptual spaces as proposed in [139] can be understood as a generic model to conceptual level approaches. Nonetheless, more computationally suitable models are needed to tackle the semantic coupling problem.

What is required is an instantiation of a conceptual space as defined by Gärdenfors, such that it builds a geometrical space that supports the basic notions of distance and similarity. I argue that such an instantiation of the model can be built solely by operating at the symbolic level. To clarify this, let us assume a large number of textual documents. If two terms such as *'laptop'* and *'device'* frequently occur with each other, one can assume that they are close within the meaning space, and that is reflected in the text, which is a symbolic representation [194].

Thus, while the meaning space is not accessible, one can approach it by how the symbols are used, i.e. a usage-based approach. This particular observation is the tenet of a class of approaches within computational linguistics known as *Statistical Semantics* [195]. Statistical semantics is based on the distributional hypothesis that states according to Harris that words that occur in the same contexts tend to have similar meanings [194]. This class of approaches to semantics is also called *Distributional Semantics* which is

the name I adopt in this work, often along with the adjective *subsymbolic* to emphasize its relative relationship to the symbolic approach.

### 5.4.3.3 Vector Space Models

One of the widely used mathematical tools to formalize and deal with distributional semantics are Vector Space Models (VSM) [196]. The premise is that a multi-dimensional vector space is built out of some textual corpora that reflect the usage of terms in a domain-agnostic or domain-specific setting. A term or a meaning then becomes a vector in the space with coordinates for each component. The main motivation for using VSM for semantic modelling lies in its highly automatic nature to build knowledge as put by Turney and Pantel:

“VSMs extract knowledge automatically from a given corpus, thus they require much less labour than other approaches to semantics, such as hand-coded knowledge bases and ontologies. ” [196]

This supports loosening the semantic coupling as suggested by Requirement *R1*, which is not the case for models that require labour such as ontologies. Besides, VSMs serve as a direct link between hypotheses of cognitive backgrounds such as the distributional hypothesis, and computational models of mathematical feasibility. Vector space mathematics are efficient and proved useful in Information Retrieval (IR) settings [197, p. 100–122], which meets the Requirement *R3* of efficiency. Furthermore, VSMs are suitable for tasks that are concerned with measuring the similarity between words, which meets a matcher’s role in an event processing engine and the Requirement *R4* of effectiveness as put by Turney and Pantel:

“VSMs perform well on tasks that involve measuring the similarity of meaning between words, phrases, and documents. ” [196]

Matrices are the basic elements to encode term statistical occurrences. For instance a term–document matrix encodes the number of times a term occurs in a document of the corpus. Documents are a special case of contexts, which could be windows of terms around the target term or terms of a particular syntactic relation with the target

term [198]. Weighting schemes can be used to increase the importance of some terms or documents. For instance, the Term Frequency Inverse Document Frequency (TF/IDF) scheme gives more weight to a term if it appears more often in a document and less often in other documents.

#### 5.4.3.4 Latent Semantic Analysis

One example, which is widely used in cognitive science and information retrieval, is the Latent Semantic Analysis (LSA) [197, p. 369–383]. LSA builds upon a term-document matrix but targets the reduction of dimensionality of this matrix. Dimensionality reduction targets various objectives: the reduction of noise from terms such as ‘the’, sparsity reduction, computational efficiency, and handling synonymy [196]. Deerwester et al. [199] introduced a principled approach to smoothing the matrix and reducing the dimensionality with an algebraic approach named as Singular Value Decomposition (SVD).

The rationale behind SVD is to *compress* the information in the matrix and make use of potential term-to-term and document-to-document relationships purely on mathematical grounds. During SVD, some information is lost. Nonetheless many benefits are gained by getting a smaller matrix. A vital point here is that the resulting matrix represents a vector space of dimensions different from the original documents. The new dimensions are *latent meanings* according to Deerwester et al. [199] and they could enhance the precision-recall in information retrieval systems.

Here I see a very clear link with the conceptual space model of Gärdenfors [139] and a return to a semiotics system where symbols and meanings are represented. Nonetheless, what these hidden meanings are and if they are the adequate representation of the meanings space is out of the scope of the LSA but I recognize this as an important direction of future work within this area. In fact, this very point of latent meanings brought critics to the LSA model and motivated the research towards spaces with more interpretable indications as put by Gabrilovich and Markovitch:

“Latent semantic models are notoriously difficult to interpret, since the computed concepts cannot be readily mapped into natural concepts manipulated by humans. The Explicit Semantic Analysis method we proposed

circumvents this problem, as it represents meanings of text fragments using natural concepts defined by humans. ” [42]

#### 5.4.3.5 Explicit Semantic Analysis

Gabrilovich and Markovitch [42] introduced an Explicit Semantic Analysis (ESA) approach for computing semantic similarity and relatedness where the dimensions of the vector space are human-defined and easy to interpret. For instance, they applied their approach on Wikipedia and proved it to outperform LSA in computing words semantic relatedness with 75% vs. 56% of correlation with human judgement. In a nutshell, Wikipedia-based *esa* builds an index of words based on the Wikipedia articles they appear in. A word becomes a vector of articles and the more common articles between two words exist, the more related the words are. For example,  $esa('parking', 'garage') > esa('parking', 'energy')$  as the formers appear frequently in common articles. Typically, semantic relatedness between a pair of terms is measured using cosine distance between the two vectors representing the two terms.

I scope this work to the distributional explicit semantic analysis semantic measure constructed from the Wikipedia corpus as of 2013 <sup>2</sup>. This is because of its relative ease of construction as it is based on a statistical analysis of unstructured document corpus as shown by Carvalho et al. [175]. I use in this work the ESA index developed by Freitas et al. [174, 175, 200]. Wikipedia is also a very comprehensive corpus of human knowledge, created and curated by vastly decoupled users. That simulates the requirement of loosely coupled semantics in event exchange. However the proposed model is generic and suitable for other measures as well, as shown in Chapters 5, 6, and 7.

#### 5.4.3.6 Limitations of Distributional Semantics

Distributional semantics and similar models are criticized on three main aspects as discussed by Lenci [201]:

1. *Compositionality*, which states that distributional semantics mainly concerns lexical meanings, i.e. meanings of individual terms. This, in fact, extends to more generic models of cognitive semantics as stated by Gärdenfors:

---

<sup>2</sup>[http://en.wikipedia.org/wiki/Wikipedia:Database\\_download](http://en.wikipedia.org/wiki/Wikipedia:Database_download)

“The emphasis of the studies within cognitive semantics has been on *lexical meaning* rather than on the meaning of sentences.” [139, p. 157]

2. *Lexical inference*, which states that distributional models do not account for various types of lexical relationships and thus cannot validate or invalidate some types of inferences. For instance, the inference “I have a *laptop*  $\rightarrow$  I have a *device*” is valid but the opposite is not as the hyponymy relationship between ‘laptop’ and ‘device’ is asymmetric.
3. *Reference and grounding*, which states that distributional models lack the capacity to address aspects of linking a word’s meaning to the world. It is argued that distributional models be nothing but symbolic representations and thus fall under the symbol grounding problem [191].

I argue that the compositionality problem be not an issue for the event matching approach. That is because linguistic structures and syntax is not the kind of data model used in event processing systems to represent events and subscriptions. In fact, models such as the attribute-value data model reduces the meaning representation problem to the individual items of attributes and values, thus making lexical meaning enough for the problem in hand.

The lexical inference problem is irrelevant to the event processing case tackled in this work as it does not address inference. Besides, inferences in event processing systems are done on the level of events rather than lexis. Matching an event to a subscription is considered as a symmetric operation in this work, and thus, asymmetry is not an issue.

I argue that what matters for this work from the grounding perspective is relevant to the semantic coupling problem. In distributional semantics, there is a need for an agreement on distributions, i.e. on the corpus in general, which is coarse-grained compared to the granular agreement needed on each symbol in classical symbolic models.

Thus, I do not agree that distributional semantics models can be considered symbolic at the same level with classical symbolic models. They indeed perform on the symbolic level, but distributions can be regarded as approximations of the lower levels in the meaning space rather than the symbolic space as it is apparent in hidden dimensions of LSA for example. Thus, while I agree on the principle of describing such models as symbolic, I argue that they should not share that same adjective with classical symbolic

models. Thus, I draw upon the distinction made by Gärdenfors [139, p. 33–58], and I call them *subsymbolic* in this work.

#### 5.4.4 Subconceptual Representation of Meaning

At the subconceptual level lies a class of approaches to meaning representation that builds on the work of early cognitive philosophers such as David Hume who states that:

“It is evident, that there is a principle of connexion between the different thoughts or ideas of the mind, and that in their appearance to the memory or imagination, they introduce each other with a certain degree of method and regularity.” [202, p. 31]

This *associationism* view as called by Gärdenfors [139, p. 40-41] has been manifested with a model of cognition: the *connectionism*.

##### 5.4.4.1 Connectionism

Connectionist systems are Artificial Neural Networks (ANNs) and consist of a number of connected units: the neurons. A neuron can be activated and deactivated based on weighted excitatory and inhibitory input coming from his input connections. Activation spreads in the network, with units getting activated and deactivated in a parallel manner. The *state* of the network at a specific point in time could be thought of as a meaning or idea [203]. Thus, the ANN operates in fact as a dynamical system, which *resonates* in a *phase space*, a space of states, each of which is a point in this high dimensional space.

Consequently, similar to the conceptual spaces, a geometrical interpretation can be given to ANNs. Similarity and distance are as natural to such a space as to conceptual-level models. Thus, what applies to the conceptual spaces also applies to connectionist models in terms of geometrical properties and support for distances and similarity.

##### 5.4.4.2 Limitations of Connectionism

Critics of connectionist models of meaning representation come from two main aspects [139, p. 42-43]:

1. *Learnability*, which states that ANNs need a large training set to learn structure and adjust weights.
2. *Interpretation*, which states that it is hard to describe what an emerging network means.

The learnability problem can be the main drawback that puts vector space models ahead of ANNs when it comes to computational systems such as event processing. Building vector space models is highly automatic and can be done in an unsupervised way as shown by Carvalho et al. [175]. This meets Requirement *R1* of loose semantic coupling as less agreement is needed at the learning stage for distributional semantics.

Vector space models are of a lower dimensionality compared to ANNs, making computation in vector space systems more suitable for Requirement *R3* of efficiency. Interpreting the networks and their hidden dimensions is similar to the issue of latent semantics models. Thus explicit semantic analysis appears as the right fit to overcome this secondary problem.

#### **5.4.5 How Subsymbolic Distributional Event Semantics Meets the Requirements**

As discussed throughout Section 5.4 various models of meanings and semantics exist, but the vector space distributional model based on explicit semantic analysis appears to meet the requirements. It also has the favourable characteristics of support of similarity, and ease of building and interpretation as discussed in the previous sections and summarized in Table 5.1.

### **5.5 Approximation**

This element of the proposed approach is crucial for the other elements to work. The need for approximate models stems from loosening the coupling on the semantic and pragmatic level. As discussed in Section 2.10, coupling is necessary to cross semantic and pragmatic boundaries. Nonetheless, it limits scalability. Loosening coupling at these levels is a compromise to tackle the trade-off between decoupling for scalability

TABLE 5.1: Semantic Models and Requirements

	Symbolic	Conceptual			Connectionist
		Conceptual Spaces [139]	LSA [199]	ESA [42]	
<i>R1.</i> Loose semantic coupling	-	+	++	++	+
<i>R2.</i> Loose pragmatic coupling	NA	NA	NA	NA	NA
<i>R3.</i> Efficiency	+++	++	++	++	+
<i>R4.</i> Effectiveness	+	++	++	+++	++
Support for similarity	-	+++	+++	+++	++
Easy to build	- - -	+	+++	+++	+
Easy to interpret	+++	++	-	+	- -

+++	the model excellently addresses the requirement
++	the model moderately addresses the requirement
+	the model slightly addresses the requirement
Legend -	the model mildly affects the requirement in a negative way
- -	the model moderately affects the requirement in a negative way
- - -	the model extremely affects the requirement in a negative way
NA	the requirement is not in question

and crossing boundaries. The cost of this compromise is a loss in effectiveness while crossing the boundaries, i.e. loss of some precision and context when processing the events.

### 5.5.1 Approximate Computing Versus Time

Approximate computing has been getting acceptance with the computing community. Nair declares in the Communications of the ACM that “Big Data Needs Approximate Computing” [204]. He states that:

“What would systems look like if we had to deal with only this new body of nontransactional data? Typical applications that mine and process this data can often tolerate lower precision, imprecise ordering, and even some unreliability in the operation of the system. Thus, an occasional stale or approximately correct piece of information delivered promptly is often more useful than up-to-date and precise information delivered later or at greater cost.” [9]

Thus, approximation from Nair's perspective is a compromise for time efficiency. In fact, this view has been dominant in computing literature, and specifically in operational research. For instance, approximation algorithms have been the focus of research where finding an optimal solution can have a combinatorial time [205]. In such a case the value of the solution can be measured as a ratio to the optimal value, e.g. the least possible weight and most valuable items in a Knapsack problem. The algorithm is then concerned with guaranteeing a bound to this ratio, given a time limit.

In databases, approximate query processing has been used to deal with large data volume and tackle the response time requirements. For instance, calculating the exact average price over a large number of products can take a few minutes and lead to a value of \$59.1415. If one takes a sample of the products and calculate the average price of the sample, it can take a few seconds and lead to a value of \$59.2. Such trade-offs could be acceptable in many settings. Techniques in databases include: sampling-based techniques [206], histogram-based techniques [207], and wavelets-based techniques [208].

In stream processing systems, similar techniques have been applied to compensate for an event input rate that exceeds an engine's capacity. Cugola and Margara [8] account for this in their Information Flow Processing (IFP) model under the name of *load shedding*, a task attributed to the receiver component (refer to Section 2.5.2). For instance, Tabul et al. [209] implement load shedding through a dynamic insert and removal of a *drop* operator within a query plan according to the input rate.

Thus, approximate computing can be seen as a viable and acceptable approach to tackle many problems when requirements such as response time become pressing. All the above approaches provide answers with a precision of less than 100%. Nonetheless they are inevitable when other parameters are factored into the equation, mostly the response time as is the dominant case in the literature.

### 5.5.2 Approximate Computing Versus Full Integration

Approximation can also be found in the literature in a different context from time. For instance, Gravano et al. [210] propose an approximate join approach in database systems over *strings*. The rationale for their work comes from the need to join tables on a string that represents a name, for example 'John Smith'. This name can exist in the

first table as ‘John A. Smith’ and in another table as ‘Smith, John’. The need to join tables on strings comes from data quality issues such as typographical errors or even more generally in data integration contexts. The result of such joins are uncertain, and can be cut using thresholds.

In a more generic context, integration processes acknowledge the fact that heterogeneous schema and data models could be used to describe different databases. Schema matching, for instance, addresses the matching and mapping of various schema attributes. The target of schema matching for integration is to map a set of source schemas into a global schema. Mapping provides a means of translation from queries against the global schema into the source schemas.

During the schema matching process, some mappings are created between items, which are not necessarily identical or originally mean the same thing. That implies a form of approximation at the schema level that accounts for the lack of a complete unified view of all schema. Such an approximation is apparent in evaluations of schema matching approaches as discussed by Chaudhri et al. [211]. Metrics that reflect approximation such as precision and recall are prime quality measures for matching effectiveness.

Although approximation is conducted during schema matching, thresholds are used, and the top-1 mapping is usually picked. Thus, the approximation becomes hidden in the process. In recent years, uncertain schema matching research has gained more attention with the realization that matchers are inherently uncertain [173]. Statistically monotonic matchers may assign a slightly lower similarity than it should to mappings of a specific precision, thus matching with top- $k$  mappings results becomes a potential solution [172]. Gal proposes a model for uncertain schema matching with top- $k$  and investigates various algorithms within several mapping constraints [172]. Uncertainty scores that reflect approximation in schema matching can be preserved and propagated to query processing, which becomes uncertain in nature.

Thus, I conclude that approximation has been in fact acknowledged regarding issues related to semantics. In this work, approximation can be used when loosening the semantic coupling leads to a lack of complete control over the event semantics. It forms a suitable model to address the need for a matching model that works within such assumptions.

### 5.5.3 Limitations of Approximation

Current event processing systems, as with most database systems, follow an exact model of matching (see Table 3.3). This trend stems from the need for precision in computing systems. In fact, the main limitation of an approximate model is that it achieves a less-than-optimal value of the solution. This value can take different forms according to the domain and application, but could be measured in the case of event matching in terms of true positives and negatives achieved by the matcher, and the degree of completeness of events. An approximate event model would typically miss some relevant events, and pass some irrelevant ones.

However, I argue that this limitation be, in fact, the result of giving up some control over the system, as full control is infeasible at large scales. This control takes the form of semantic and pragmatic agreements in event systems. Thus, loose coupling at these dimensions implies the loss of some precision in order to scale. Accepting this approach is fundamental to scale event processing systems in open, distributed, and heterogeneous environments. In fact, this has been the case in Information Retrieval (IR) [197, p. 151–175] where precision, recall, and derived measures are standard to evaluate IR systems.

From this perspective, approximation appears as a proper fit for event systems to meet the requirements *R1* and *R2* of loose semantic and pragmatic coupling. The use of approximate event models could lead to the use of techniques such as semantic relatedness instead of exact string comparison that is costly from a time performance point of view. Thus, approximation is not the best solution from the perspective of requirements *R3* and *R4* of effectiveness and efficiency. However, it should target high values of these measures within an approximate paradigm. Applications with hard real-time deadlines or an exact precision requirement such as security systems or critical infrastructures may not be the ideal applications. It can be better to afford the cost of establishing semantic and pragmatic agreements and use an exact event processing system rather than leaving approximation to the matcher.

### 5.5.4 How Approximation Meets the Requirements

As discussed throughout Section 5.5 approximation is required in event processing systems that are distributed and decoupled by nature. Approximate and exact models are

TABLE 5.2: Approximation and Requirements

	<b>Exact Model</b>	<b>Approximate Model</b>
<i>R1.</i> Loose semantic coupling	+	+++
<i>R2.</i> Loose pragmatic coupling	+	+++
<i>R3.</i> Efficiency	+++	++
<i>R4.</i> Effectiveness	+++	++

Legend    +++    the model excellently addresses the requirement  
           ++     the model moderately addresses the requirement  
           +     the model slightly addresses the requirement

the main models for event systems, but the approximate approach seems the one to meet the main requirements as summarized in Table 5.2.

## 5.6 Event Flow Model

The event processing engine plays the key role to filter and make sense out of the event cloud. Cugola and Margara present an abstract functional model for information flow processing systems in [8]. The core components of an event engine in their model are the event *Receiver*, the *Decider*, the *Producer* and the event *Forwarder*. Event *Sources*, *Consumers* and *Users* interact with the engine through protocols and condition/action *Rules*. Figure 5.2 presents an elaboration of Cugola and Margara’s model with components as described in Section 2.5.2.

The proposed model extends the event processing engine as follows:

- *Rules* are equipped with the *tilde*  $\sim$  semantic approximation operator. *Rules*, which consist only of a detection part for single events, are called *Subscriptions* throughout the rest of this work.
- The *Single Event Matcher* is equipped with matching and mapping algorithms to detect events semantically *relevant* to approximate subscriptions. The *Single Event Matcher* works in two modes
  - Top-1 which forwards the best mapping between an event and a subscription to the *Consumers*.

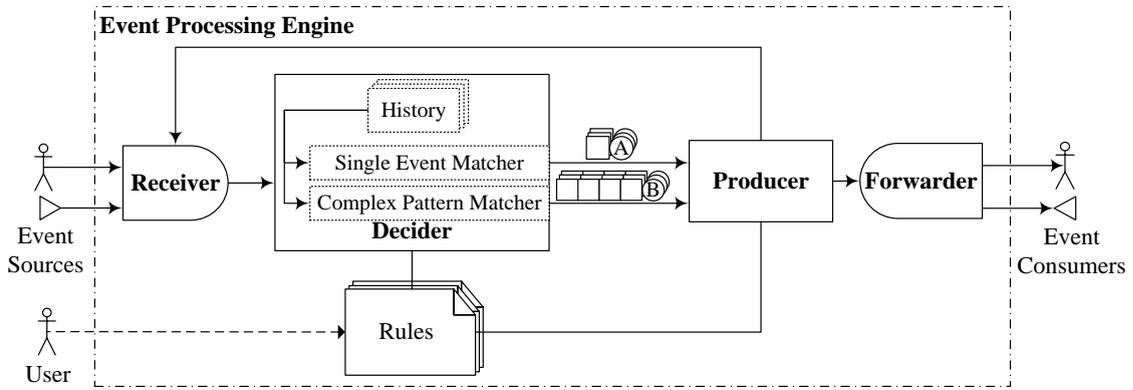


FIGURE 5.2: Event flow model

- Top- $k$  that results in a list of top- $k$  possible mappings between an event and a subscription. The top- $k$  mappings of various events to subscriptions go to the *Complex Pattern Matcher*.
- The *Complex Pattern Matcher* performs a probabilistic reasoning to deduce the probabilities of occurrences of the derived events in the action parts of the complex rules.

The focus of this thesis is on the *Single Event Matcher* sub-component of the *Decider*.

## 5.7 Event Model

The event model used in this chapter is an attribute-value model, but the discussion is as relevant to other models such as hierarchical or graph-based event models. Each event is a set of tuples. Each tuple consists of an attribute-value pair. Example 5.1 represents an event complying to the model.

**Example 5.1** (Increased Energy Consumption Event).

*{type: increased energy consumption event,*  
*measurement unit: kilowatt-hour, device: computer,*  
*desk: desk 112c, office: room 112, floor: ground floor,*  
*zone: building, city: Galway, country: Ireland, continent: Europe}*

The formal definition of the event model is as follows: Let  $E$  be the set of all events and let  $A$  and  $V$  be the sets of possible attributes and values respectively. Let  $T$  be

the set of possible attribute-value pairs, i.e. tuples, such that  $T = \{(a, v) : a \in A \wedge v \in V\}$ . Let's define two functions  $Attribute : T \rightarrow A$  and  $Value : T \rightarrow V$  that give the attribute and value respectively when applied to a tuple such that  $Attribute(a, v) = a$  and  $Value(a, v) = v$ . An event  $e \in E$  is a set of tuples where no two distinct tuples can have the same attribute as in Equation 5.1.

$$e = \{t : t \in T \wedge \forall t_1, t_2 \in e, t_1 \neq t_2 \Rightarrow Attribute(t_1) \neq Attribute(t_2)\} \quad (5.1)$$

## 5.8 Language Model

Subscriptions follow a conjunctive query form of attribute-value predicates. Each predicate uses the equality operator to signify exact equality or approximate equality when indicated. Other Boolean and numeric operators such as  $! =$ ,  $>$ , and  $<$  are kept out of the language for the sake of simplicity and to focus the discourse on semantic matching. Nonetheless, the model can be extended to encompass such operators as discussed in Section 5.9.5.

Each predicate consists of an attribute, a value, and specifications of the semantic approximation for the attribute and the value. The most notable feature of the language is the *tilde*  $\sim$  operator that helps specify the approximation for an attribute/value when it follows it. The *tilde*  $\sim$  operator also helps specify optionally the semantic measure to be used for the approximation as shown in Example 5.2.

**Example 5.2** (Approximate Subscription).

*{type =increased energy consumption event,*  
*device = laptop~,*  
*room~esa = room 112}*

The author of the subscription in Example 5.2 is interested in an event of exactly the type '*increased energy consumption event*'. The subscription specifies that the device can be '*laptop*' or something related semantically to '*laptop*' with no specification of what semantic measure to use, meaning that the default should be used. The subscription also states that the event's '*room*' must be '*room 112*'. However, it states that the attribute '*room*' itself can be semantically relaxed using the *esa* semantic measure.

The formal definition of the language model is as follows: Let  $S$  be the set of subscriptions and let  $A$  and  $V$  be the sets of possible attributes and values respectively which can be used in a subscription. Typically there are no restrictions on  $A$  or  $V$  and the user is free to use any term or combination of terms. Let  $SM$  be the set of all possible semantic relatedness measures available for approximate subscriptions. Each predicate is a sextuple that consists of the attribute, the value, whether or not the attribute/value are approximate, and the semantic measure to relax the attribute/value if applicable. Let  $P$  be the set of possible predicates. Thus,  $P$  is a subset of a Cartesian product as shown in Equation 5.2.

$$P = \{p : p = (a, v, app_a, app_v, sem_a, sem_v) \in A \times V \times \{0, 1\}^2 \times SM^2\} \quad (5.2)$$

A subscription  $s \in S$  is a set of predicates such that  $s = \{p : p \in P\}$ . Let  $Attribute : P \rightarrow A$  and  $Value : P \rightarrow V$  be two functions that give the attribute and value respectively when applied to a predicate. Let  $App_A^\sim : P \rightarrow \{0, 1\}$  be a Boolean function that specifies if the attribute of a predicate  $p \in P$  must be approximated if  $App_A^\sim(p) = app_a = 1$ . Let  $Sem_A^\sim : P \rightarrow SM$  be a function that specifies for a predicate  $p \in P$  the semantic measure  $Sem_A^\sim(p) = sem_a$  to be used to approximate its attribute if the predicate is approximated, i.e. if  $App_A^\sim(p) = 1$ . Let  $App_V^\sim$  and  $Sem_V^\sim$  be two functions for values approximation in a similar way. An exact subscription is a special case of approximate subscriptions where all attributes and values are not approximated.

## 5.9 Matching

Given an approximate subscription  $s \in S$  and an event  $e \in E$ , an approximate semantic single event matcher  $\mathcal{M}$  decides on the semantic relevance between  $s$  and  $e$ . The relevance results from a semantic mapping between attribute-value predicates of  $s$  and attribute-value tuples of  $e$ . Example 5.3 shows a possible mapping between the event in Example 5.1 and the approximate subscription in Example 5.2.

**Example 5.3** (Mapping between the Subscription and the Event).

$$\begin{aligned} \sigma = & \{(type=increased \textit{ energy consumption event} \\ & \leftrightarrow type:increased \textit{ energy consumption event}), \\ & (device = laptop \sim esa \leftrightarrow device:computer), \end{aligned}$$

$(\mathbf{room} \sim \mathbf{esa} = \mathbf{room} \ \mathbf{112} \leftrightarrow \mathbf{office:} \ \mathbf{room} \ \mathbf{112})\}$

$\mathcal{M}$  works in two modes: the top-1 mode which decides on the most probable mapping between  $s$  and  $e$ , and the top- $k$  mode which decides on the top- $k$  probable mappings to be used later for complex event processing. It has been shown in [172] that producing the top- $k$  mappings increases the chance of hitting the correct mapping. That is due to the statistical monotonicity principle which roughly states that mappings with higher similarities tend to have higher precisions but with a statistical distribution such that a mapping with a slightly smaller similarity can have a better precision than that of higher similarity [172]. Uncertain mapping between predicates and tuples is inherent in both matching modes with probabilities being the final outcome.

The formal definition of the matching model is as follows: Let  $C = s \times e$  be the set of all possible correspondences between the predicates of  $s$  and the tuples of  $e$ .  $\forall c = (p, t) \in C \Rightarrow p \in s \wedge t \in e$ .  $\Sigma = 2^C$  is the power set of  $C$  and represents all the possible mappings between  $s$  and  $e$ . Let  $\Gamma : \Sigma \rightarrow \{0, 1\}$  be a Boolean constraint function which defines the validity of a mapping  $\sigma \in \Sigma$ . I adopt in this work an  $n : 1$  cardinality constraint function which allows every predicate to be mapped to one and only one event tuple. The set of all *valid* mappings according to  $\Gamma$  is denoted as  $\Sigma_\Gamma$ . There are exactly  $n$  correspondences in any valid mapping  $\sigma$  where  $n$  is the number of predicates in the subscription  $s$ .

For any valid mapping  $\sigma \in \Sigma_\Gamma$  there exists a probability function that quantifies the probability of every predicate-tuple correspondence  $(p, t) \in \sigma$  such as (**device** = **laptop**  $\sim$  **esa**  $\leftrightarrow$  **device:computer**). The probability of  $(p, t)$  is denoted as  $p_{(p,t)}$  where  $p_{(p,t)} \in [0, 1]$ . The probabilities  $p_{(p,t)}$  form a probability space  $\mathcal{P}_\sigma$  over all  $(p, t) \in \sigma$  as shown in Equation 5.3.

$$\sum_{(p,t) \in \sigma} p_{(p,t)} = 1 \quad (5.3)$$

For any valid mapping  $\sigma \in \Sigma_\Gamma$  there exists a probability function which quantifies the probability of the overall mapping  $\sigma$  among other possible mappings. The probability of  $\sigma$  is denoted as  $p_\sigma$  where  $p_\sigma \in [0, 1]$ . To realize a matcher such as the matcher  $\mathcal{M}$  described above, I propose an ensemble of matchers as illustrated in Figure 5.3.

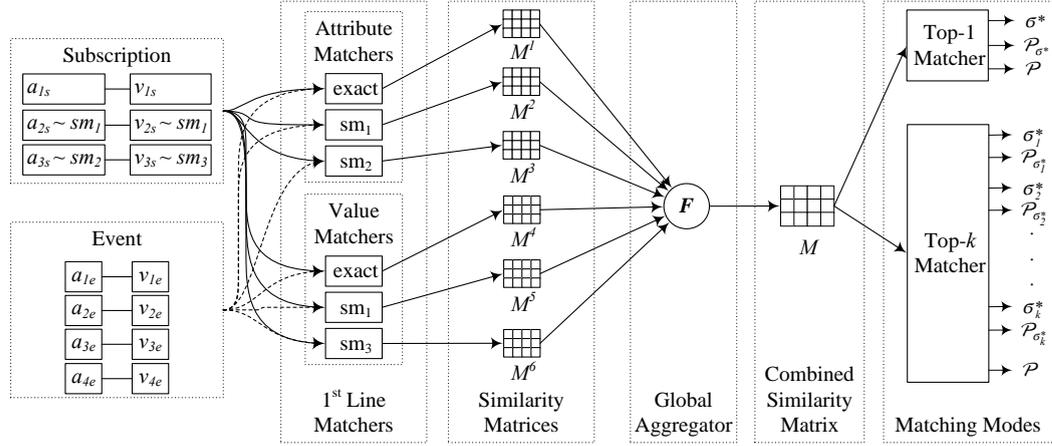


FIGURE 5.3: The approximate semantic event matcher model

### 5.9.1 First-Line Matchers and Similarity Matrices

First-line matchers operate on actual attributes and values of  $s$  and  $e$  and output similarity matrices according to the semantic measures  $sm \in SM$  in  $s$ . A similarity matrix  $M^l$  is an  $n \times m$  matrix where  $n$  is the number of predicates in  $s$  and  $m$  is the number of tuples in  $e$ . Each element  $M_{i,j}^l$  of  $M^l$  represents the degree of similarity between predicate  $p_i \in s$  and tuple  $t_j \in e$  according to the matcher  $l$ . Typically  $M_{i,j}^l \in \mathbb{R}$ . For instance, the cell  $M_{i,j}^{l_1}$  of the correspondence (**device = laptop**  $\sim$   $esa \leftrightarrow$  device:computer) would be assigned the value 1 by the matcher  $l_1$  responsible for attribute exact matching. Another cell  $M_{i,j}^{l_2}$  in another matrix  $M^{l_2}$  would be assigned a value  $< 1$  by the matcher  $l_2$  responsible for value approximate matching.

There are two sets of first-line matchers: matchers which operate on the attributes of predicates/tuples and those which operate on values. There is an exact matcher for attributes and an exact matcher for values. These exact matchers handle the predicates' attributes/values which do not have any approximation specification and ignore the rest. An exact matcher operates on attributes or values and produces a Boolean similarity matrix, i.e.  $M_{i,j}^{exact} \in \{0, 1\}$ . Let the matchers labelled  $exa$  and  $exv$  be the attributes and values exact matchers respectively. Let  $p_i \in s, t_j \in e$ , Equation 5.4 shows how the attributes exact matcher assigns similarities. The same applies to the values exact matcher.

$$M_{i,j}^{exa} = \begin{cases} 0 & \text{if } App_A^{\sim}(p_i) = 0 \wedge Attribute(p_i) \neq Attribute(t_j) \\ 1 & \text{otherwise} \end{cases} \quad (5.4)$$

The remaining first-line matchers are approximate matchers, each of which uses one of the semantic measures used in the subscription. An approximate first-line matcher handles the predicates' attributes/values which are relaxed by its corresponding semantic measure and ignores the rest. It operates on attributes or values and produces a similarity matrix as shown in Equation 5.5 which explains the behaviour of an approximate attribute matcher  $l$  which corresponds to a semantic measure  $sm$ . The same applies to values approximate matchers. Let  $p_i \in s, \forall t_j \in e$ :

$$M_{i,j}^l = \begin{cases} sm(Attribute(p_i), Attribute(t_j)) & \text{if } App_A^{\sim}(p_i) = 1 \wedge Sem_A^{\sim}(p_i) = sm \\ 1 & \text{otherwise} \end{cases} \quad (5.5)$$

The inner working and order of first-line matchers can be changed according to optimization strategies as discussed in Section 5.9.5.2.

### 5.9.2 Global Aggregator and the Combined Similarity Matrix

The global aggregator  $F$  operates on the resulting similarity matrices from first-line matchers and produces a single combined similarity matrix  $M$  as shown in Figure 5.3. For example, the correspondence (**device = laptop**  $\sim$  *esa*  $\leftrightarrow$  *tool:computer*) would be assigned a similarity value of 0 by the attribute exact first-line matcher because '*device*'  $\neq$  '*tool*' and they are not approximated. It would be assigned a similarity value  $x > 0$  by the value approximate first-line matcher of *esa* as '*laptop*' is related to '*computer*' and they are approximated. The global aggregator shall combine the 0 similarity from the first matrix with the similarity  $x$  from the other matrix and conclude a judgement of 0 as an overall similarity according to matching semantics as the correspondence violates it for attributes.

$M$  represents an overall judgement on the similarity between the subscription's predicates and the event's tuples. The global aggregator chosen for the model is the element-wise matrix multiplication operator  $\circ$ , also called the Hadamard product as defined in Equation 5.6.

$$M_{i,j} = (M^1 \circ M^2 \circ \dots \circ M^L)_{i,j} = \prod_{l=1}^{l=L} M_{i,j}^l \quad (5.6)$$

The Hadamard product is commutative and associative. It is also efficient to be implemented as it can be computed in  $O(n \times m \times L)$  time. The zero and identity elements

of the Hadamard product easily extends from the familiar zero and identity elements of the multiplication operator  $\times$ , i.e. 0 and 1. This makes it easy to pass information from the first-line matchers to the aggregator. i.e. to neglect or skip a correspondence  $(p_i, t_j)$  by assigning 0 or 1 as its similarity.

### 5.9.3 Top-1 Matcher

In the top-1 matching mode, the top-1 matcher operates over the combined similarity matrix  $M$ . It produces the best mapping  $\sigma^*$  and the space  $\mathcal{P}_{\sigma^*}$  which defines the probabilities of correspondences  $c_i \in \sigma^*$ . It also produces the space  $\mathcal{P}$  which defines the probability that  $\sigma^*$  is the correct mapping between the subscription  $s$  and the event  $e$  as illustrated in Figure 5.3. Given the combined similarity matrix  $M$ , the best mapping  $\sigma^*$  can be computed by choosing the tuple  $t_{j_i}$  which has the maximal similarity for every predicate  $p_i$  as shown in Equation 5.7.

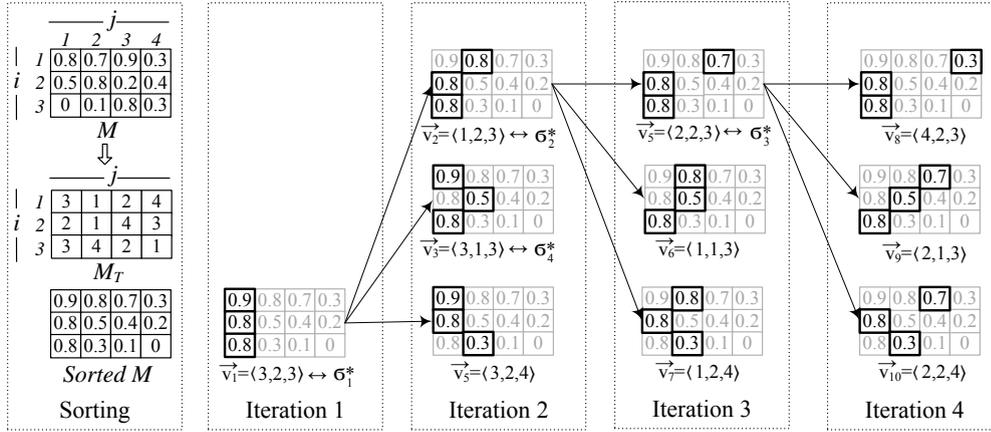
$$\sigma^* = \{(p_i, p_{j_i}) : 1 \leq i \leq n \wedge j_i = \arg \max_j (M_{i,j})\} \quad (5.7)$$

According to Equation 5.7,  $\sigma^*$  can be found in  $O(n \times m)$  operations.  $\sigma^*$  contains exactly  $n$  predicate-tuple correspondences under the  $n : 1$  matching semantics.

To create the probability space  $\mathcal{P}_{\sigma^*}$  Equation 5.8 defines the  $n$  probabilities of the correspondences (subscription predicate  $\leftrightarrow$  event tuple) of  $\sigma^*$ . That is done by dividing each (predicate  $\leftrightarrow$  tuple) similarity in the mapping by the sum of all similarities so the sum of probabilities becomes 1. These probabilities can be computed in  $O(n)$  time.

$$p_{i,j_i} = \frac{M_{i,j_i}}{\sum_{i=1}^n M_{i,j_i}}, 1 \leq i \leq n \quad (5.8)$$

To create the probability space  $\mathcal{P}$  which defines the probability that  $\sigma^*$  is the correct mapping between  $s$  and  $e$  among other possible mappings, there is a need to normalize the similarity matrix  $M$  among other possible matrices. The maximal possible similarity value  $max_{sm}$  of each measure  $sm \in SM$  is used as they are universal among all mappings so the probability that  $\sigma^*$  is correct  $\leq 1$ . The maximum value of any element in  $M$  is  $max_{SM} = \prod_{sm \in SM} max_{sm}$ . Thus, the probability that  $\sigma^*$  is correct is defined in Equation

FIGURE 5.4: Top- $k$  by an evolving frontier algorithm

5.9 and can be computed in  $O(n)$  time.

$$p_{\sigma^*} = \sum_{(p_i, t_{j_i}) \in \sigma^*} \frac{M_{i, j_i}}{n * \max_{SM}} \quad (5.9)$$

#### 5.9.4 Top- $k$ Matcher

In the top- $k$  matching mode, the matcher  $\mathcal{M}$  produces a ranked list of the best  $k$  mappings  $\sigma_1^*, \sigma_2^*, \dots, \sigma_k^* \in \Sigma_\Gamma$  along with the probability spaces of correspondences  $\mathcal{P}_{\sigma_r^*}$  and the probability space of mappings  $\mathcal{P}$  as illustrated in Figure 5.3. Given the combined  $n \times m$  similarity matrix  $M$  between a subscription  $s$  and an event  $e$ , I propose an efficient algorithm to find the top- $k$  mappings  $\sigma_r^*$  based on an evolving Pareto frontier in a vector space as shown in Figure 5.4.

Consider the set  $V$  of all  $n$ -dimensional vectors where the components of each vector are tuples of  $e$ , i.e.  $V = \{\vec{v} : \vec{v} \in \{1, 2, \dots, m\}^n\}$ . Each vector  $\vec{v} \in V$  encodes a valid mapping  $\sigma \in \Sigma_\Gamma$  as shown in Equation 5.10. This is denoted as  $\vec{v} \leftrightarrow \sigma$ .

$$\vec{v}^i = j \Leftrightarrow (p_i, t_{j_i}) \in \sigma : \sigma \in \Sigma_\Gamma, p_i \in s, t_{j_i} \in e \quad (5.10)$$

For example, let an approximate subscription be

{**type** =increased energy consumption event,  
**device** = laptop~,  
**room**~esa = room 112}

Let an event be

{**type**: increased energy consumption event,  
**office**: room 112,  
**device**: computer}

Let a mapping

$\sigma = \{(\mathbf{type} = \mathbf{increased\ energy\ consumption\ event}$   
 $\leftrightarrow \text{type:increased energy consumption event}),$   
 $(\mathbf{device} = \mathbf{laptop} \sim \mathbf{esa} \leftrightarrow \text{device:computer}),$   
 $(\mathbf{room} \sim \mathbf{esa} = \mathbf{room\ 112} \leftrightarrow \text{office: room 112})\}$

A vector  $\vec{v} = \langle 1, 3, 2 \rangle$  corresponds to this mapping between the subscription's predicates 1, 2, and 3 to the event's tuples 1, 3, and 2 respectively.

To quantify a vector, and hence its corresponding mapping, I define an operator  $\|\dots\|_M : V \rightarrow \mathbb{R}$  given the similarity matrix  $M$  as in Equation 5.11.

$$\|\vec{v}\|_M = \sum_{i=1}^{i=n} M_{i,j} : j = \vec{v}^i \quad (5.11)$$

The more similarity a vector  $\vec{v}$  encodes according to  $M$ , the more becomes  $\|\vec{v}\|_M$ .

A vector  $\vec{v} \in V$  is dominated by a vector  $\vec{u} \in V$  if and only if the similarity encoded by all components of  $\vec{v}$  is greater than or equal to the similarity encoded by the corresponding components of  $\vec{u}$  with at least one similarity to be greater than and not equal. That means that the mapping  $\sigma_{\vec{v}} \leftrightarrow \vec{v}$  is better than the mapping  $\sigma_{\vec{u}} \leftrightarrow \vec{u}$  and that  $\|\vec{v}\|_M > \|\vec{u}\|_M$ . This is denoted as  $\vec{v} \prec \vec{u}$ .

A vector  $\vec{u}$  directly dominates a vector  $\vec{v}$  if there exists no vector  $\vec{w}$  different from  $\vec{u}$  and  $\vec{v}$  where  $\vec{v} \prec \vec{w} \prec \vec{u}$ . This is denoted as  $\vec{v} \prec\prec \vec{u}$ . For instance,  $\vec{v}_2 \prec\prec \vec{u}_7$  in Figure 5.4. In terms of similarity, this means that  $\vec{v}_2$  encodes a mapping that has more similarity between predicates and tuples than the mapping encoded by  $\vec{v}_7$ .

The proposed algorithm which is called *Top-k by an Evolving Frontier* is depicted in Figure 5.4. It starts by sorting the rows in the similarity matrix  $M$  in descending order and keeping track of the new locations of tuples in a matrix called  $M_T$ . The best

mapping  $\sigma_1^*$  is represented by the elements of the sorted matrix which have  $j = 1$ . This is equivalent to a vector  $\vec{v}_1$  which is dominated by all other vectors of  $V$ . Vector  $\vec{v}_1$  is a Pareto frontier. A Pareto frontier is a set of vectors which are dominated by all other non-searched vectors not in the frontier, and which do not dominate each other.

---

**ALGORITHM 1:** Top- $k$  by an Evolving Frontier
 

---

**Input:** the similarity matrix  $M$ , the required number of mappings  $k$ 
**Result:** top- $k$  mappings  $\Sigma_k$ 

```

1 begin
2    $\Sigma_k \leftarrow \Phi$ ;
3    $SortedM \leftarrow SortRows(M)$ ;
4    $M_T \leftarrow TupleIndicesOf(SortedM)$ ;
5    $Frontier \leftarrow \Phi$ ;          /* a priority queue of vectors whose key is  $\|\vec{v}\|_M$  */
6    $\vec{v}_1 \leftarrow \langle 1, 1, \dots, 1 \rangle$ ;
7   Enqueue( $\vec{v}_1$ ,  $Frontier$ );
8   for  $1 \leq r \leq k$  do
9      $\vec{v}_r \leftarrow Head(Frontier)$ ; /* get the best vector  $\vec{v}_r$  from head of the queue */
10     $\Sigma_k \leftarrow \Sigma_k \cup \sigma_r^* : \vec{v}_r \leftrightarrow \sigma_r$ ;
11     $Frontier \leftarrow Frontier \setminus \vec{v}_r$ ;          /* remove  $\vec{v}_r$  from head of the queue */
12     $D \leftarrow \{\vec{d} : \vec{v}_r \prec \prec \vec{d}\}$ ;          /* set of  $n$  vectors directly dominating  $\vec{v}_r$  */
13    Enqueue( $D$ ,  $Frontier$ );
14  end
15  return  $\Sigma_k$ ;
16 end

```

---

Because the dominance as defined is equivalent to the quality of mapping, then the best mapping of non-searched mappings must lie on the frontier. The algorithm works in iterations, and the frontier keeps evolving. When a vector is found to correspond to the best mapping, it is removed from the frontier. All vectors that directly dominate the removed vector are candidates for search, and thus, they are added to the frontier. Vectors, which directly dominate a vector, can be found by changing one of its  $n$  components at a time by moving one step rightwards in the rows of the sorted similarity matrix. As a result, the algorithm can find the top- $k$  best mappings within  $k$  iterations and the search space is kept to a minimum and updated with  $n$  vectors at each iteration. The correctness of the algorithm follows from the previous discussion. Algorithm 1 shows its main steps.

The frontier is presented as a priority queue of vectors  $\vec{v} \in V$  on the key  $\|\vec{v}\|_M$ . So, searching the frontier is efficient as the best vector sits at the head of the queue. Sorting  $M$  in Step 3 has a time complexity of  $O(n.m.\log(m))$ . Taking the head of the queue in Step 9 for  $k$  times has an overall complexity of  $O(k)$ . Generating the set  $D$  in Step 12 for  $k$  times has an overall complexity of  $O(k.n^2)$ . Enqueuing the  $n$  vectors of  $D$  in

Step 13 for  $k$  times has an overall complexity of  $O(n.\log(n) + k.\log(k) - k)$ . That makes the overall time complexity of the proposed algorithm proportional to  $O(n.m.\log(m) + n.\log(n) + k.n^2 + k.\log(k))$ . Creating the probability spaces  $\mathcal{P}_{\sigma_r^*}$  of correspondences and the probability space  $\mathcal{P}$  of mappings  $\sigma_r^*$  is achievable in the same way as in the top-1 mode by normalizing  $M$  as shown in Equations 5.8 and 5.9, with the difference that in the top- $k$  mode there are  $k$  probabilities  $p_{\sigma_r^*}$  in  $\mathcal{P}$  to be calculated.

### 5.9.5 Matcher Extensibility

This section tackles the extensibility of the matcher to include Boolean and numeric operators and to leverage common optimization strategies in event processing.

#### 5.9.5.1 Boolean and Numeric Operators

The current language as described in Section 5.8 is confined to the equality operator. However Boolean and numeric operators such as  $! =$ ,  $<$ ,  $\leq$ ,  $>$ , and  $\geq$  can be added as exact first-line value matchers in Figure 5.3. Let  $(\mathbf{temperature} > 25)$  be a predicate and let  $\{\mathbf{location}:\text{first floor}, \mathbf{temperature}:26\}$  be an event of two tuples. Then an exact matcher for the  $>$  operator will produce a Boolean matrix which contains 0 for the cell which corresponds to the predicate and first tuple, while it contains 1 for the predicate and the second tuple.

If the predicate contains an approximate attribute, i.e.  $(\mathbf{temperature} \sim \text{esa} > 25)$  then the approximate first-line matcher of attributes produces the similarities for the mappings ('temperature'  $\leftrightarrow$  'location') and ('temperature'  $\leftrightarrow$  'temperature'). This result will need to be combined then with the matrix produced by the  $>$  operator first-line matcher.

#### 5.9.5.2 Optimization

A distinguishing aspect of matching in event processing systems is that there is typically a large number of subscriptions  $S$  to be matched against every event  $e$ . There are two main types of optimization strategies which can be recognized in the literature:

leveraging commonalities between subscriptions and changing the evaluation order of predicates [212, 213].

Leveraging commonalities is based on the observation that two subscriptions  $x, y \in S$  may share one or more predicates. Thus, it is more efficient to evaluate unique atomic predicates first and then propagate the results to subscriptions. In the model in Figure 5.3, this can be achieved by decomposing registered subscriptions into their predicates before entering the matcher. The set of predicates then forms the entries of the similarity matrices. The top-1 and top- $k$  matchers then aggregate the matching results according to each subscription. This affects the creation of probability spaces which shall consider only those predicates which are a part of the subscription in question. A matcher equipped with this strategy is called a *commonalities-based* matcher.

The idea of ordering the evaluation of predicates stems from inter-dependencies between predicates. In the proposed model, there are two distinct types of predicates: exact and approximate. If an exact predicate of a subscription evaluates to *False* then there is no need to evaluate the rest of the subscription's predicates if they do not belong to other subscriptions. Thus, the execution of first-line matchers is ordered by starting with the exact matchers first. Another observation is that when an approximate attribute/value of a predicate evaluates to 0 then the whole predicate evaluates to 0. A matcher equipped with this strategy is called an *order-based* matcher.

## 5.10 Evaluation

The model introduced in this chapter loosens the semantic coupling dimension of event processing. Thus, to test the hypotheses  $H1$  and  $H4$  I evaluate to what extent this model is effective and efficient, and to what extent it loosens the semantic coupling. This section describes the evaluation methodologies and the experiments' results.

### 5.10.1 Evaluation Metrics

Evaluation metrics can be classified into two categories: effectiveness and efficiency metrics [214]. Effectiveness metrics measure the quality of event matching. A fundamental

TABLE 5.3: Base Concepts for Effectiveness Evaluation

	<b>Ground Truth Relevant Events</b>	<b>Ground Truth Irrelevant Events</b>
<b>Matcher Relevant Events</b>	TP (True Positive)	FP (False Positive)
<b>Matcher Irrelevant Events</b>	FN (False Negative)	TN (True Negative)

requirement is the existence of a ground truth which divides events into relevant and irrelevant with respect to each approximate subscription.

Table 5.3 shows the base concepts needed for evaluating effectiveness. For all these concepts to exist, the resulting events from the matcher must be divisible into two distinct sets of matcher relevant and irrelevant events. In the case of the approximate matcher which assigns probabilities to events with respect to a subscription, the two sets can be achieved by ranking and cutting off using recall levels. *Precision*, *Recall*, and the combined *F<sub>1</sub>Score* have been used for effectiveness evaluation.

*Precision* measures the proportion of relevant events discovered by the matcher with respect to all the discovered events such that  $Precision = TP / (TP + FP)$ . *Recall* measures the proportion of relevant events discovered by the matcher with respect to all the known relevant events from the ground truth such that  $Recall = TP / (TP + FN)$ . Precision and recall are calculated for the whole set of subscriptions  $S$  by averaging the precision and recall achieved for all individual subscriptions respectively.

*F<sub>1</sub>Score* is computed at 11 recall points,  $\{0, 0.1, 0.2, \dots, 1.0\}$ , to cover all the precision-recall curve without using thresholds and the maximal *F<sub>1</sub>Score* is then used. The *F<sub>1</sub>Score* equally combines *Precision* and *Recall* such that  $F_1Score = (2 \times Precision \times Recall) / (Precision + Recall)$ . The metric used for evaluating time efficiency is the matcher *Throughput* defined as  $Throughput = (Number\ of\ processed\ events) / (Time\ unit)$ .

Additionally, to measure the loosening in semantic coupling, I use two measures: alternative *number of exact subscription rules* that would be needed in a coupled model, and the *degree of approximation* used in the approximate subscriptions. These two measures reflect to a large extent the loosening in coupling. These measures are compared to the exact matching model's numbers which would typically have a large number of exact subscription rules that have zero degree of approximation as a result of coupling.

### 5.10.2 Methodology for Effectiveness Evaluation

The evaluation methodology for effectiveness is based on the methodologies of the schema matching/mapping community [211]. The task of schema matching/mapping is to find the best mapping between a source schema  $S$  and a target schema  $T$ . The common evaluation methodology is based on a real world workload of a relatively small number of schemas, and manually decided ground truth mappings for the baseline [211]. However, due to the large-scale nature of the Internet of Things it is preferable to evaluate with large sets of events and subscriptions. Thus, specifying the ground truth mappings becomes a challenge.

In recent years there has been a trend towards synthetic evaluation [214]. Two approaches can be recognized: a top-down approach and a bottom-up approach. In the top-down approach a source schema  $S$  is used. Then, by systematically removing and transforming parts of  $S$ , it is possible to generate various target schemas and their corresponding ground truth mappings to  $S$  as in eTuner [215]. In the bottom-up approach pairs of relative small source and target schemas with known ground truth mappings are used. Systematic transformations are then applied to the schemas and the mappings to generate other pairs with corresponding ground truth mappings as in STBenchmark [216].

Synthetic evaluation for various purposes has been widely used in event processing as discussed in Chapter 3, see [26, 29–31, 33, 34, 37, 39, 158–160, 165]. Within the context of event matching there are approximate subscriptions and events instead of source and target schemas. Similarly to the idea in STBenchmark [216], I start with pairs of exact subscriptions  $X$  and events  $E$  with known ground truth which are simply the result of exact matching of events to subscriptions. A semantic expansion transformation is then applied to the events and subscriptions based on thesaurus similarly to the synonyms transformation based on the Merriam-Webster thesaurus [217] in eTuner [215]. Along with semantic expansion, the ground truth is updated accordingly. The methodology is outlined in Figure 5.5 and detailed in the following sections.

TABLE 5.4: Sensor Capabilities

Sensor Capabilities
solar radiation, particles, speed, wind direction, wind speed, temperature, water flow, atmospheric pressure, noise, ozone, rainfall, parking, radiation par, CO, ground temperature, light, NO <sub>2</sub> , soil moisture tension, relative humidity, energy consumption, CPU usage, memory usage

### 5.10.2.1 Generation of a Seed Event Set

The goal of the event set is to simulate the case of a large set of events that would exist in a large-scale heterogeneous environment of event producers and consumers. The seed event set,  $SE$  in Figure 5.5, has been synthesized based on a set of IoT sensors identical to the ones deployed in the SmartSantander smart city project [50] and the Linked Energy Intelligence (LEI) dataspace [218]. The SmartSantander project proposes a city-scale experimental research testbed for IoT applications and services based on sensors deployed in a set of European cities. The LEI project targets sensing buildings for energy saving and management purposes. The used sensor capabilities are shown in Table 5.4. Some capabilities have been the subject of study by Derguech et al. in [219, 220].

A set of car brands from the Yahoo! directory [221] has been used to generate vehicle platforms for mobile sensors. A set of home based appliances from the BLUED KDD dataset has been used as indoor platforms [222]. For indoor locations, rooms from the LEI DERI Building has been used [223]. For geographical locations, the SmartSantander project locations, as well as the LEI location of Galway city, have been used.

Seed events are generated by randomly combining various attributes and values from the datasets. A set of 165 seed events has been used to generate events for the experiments. Example 5.4 represents a resulting seed event generated at this stage.

**Example 5.4** (Seed Event).

*{type: increased energy consumption event,*  
*measurement unit: kilowatt-hour, device: laptop,*  
*desk: desk 112c, room: room 112, floor: ground floor,*  
*zone: building, city: Galway, country: Ireland, continent: Europe}*

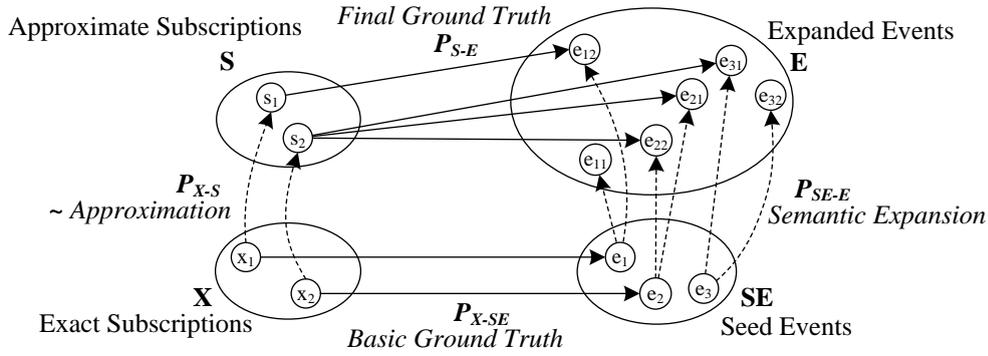


FIGURE 5.5: Methodology for effectiveness evaluation

### 5.10.2.2 Generation of an Exact Subscription Set

Exact subscriptions are generated by randomly picking a number of tuples from the seed events and turning them into exact subscriptions, the set  $X$  in Figure 5.5, Example 5.5 represents an exact subscription of length 3 generated from the seed event in Example 5.4.

**Example 5.5** (Exact Subscription).

*{type = increased energy consumption event,*  
*device = laptop,*  
*floor = ground floor}*

### 5.10.2.3 Generation of Ground Truth for Exact Subscriptions and Seed Events

An exact matcher has been used to find the relevant and irrelevant seed events to exact subscriptions, function  $P_{X-SE}$  in Figure 5.5. An event is relevant to an exact subscription if all the predicates in the subscription are exactly matched by at least one tuple from the event.

### 5.10.2.4 Semantic Expansion of Seed Events

The purpose of semantic expansion of seed events, transformation  $P_{SE-E}$  in Figure 5.5, is to generate a relatively large amount of events for evaluation where the semantic heterogeneity property holds. Thus, the Merriam-Webster online thesaurus has been used [217] as in eTuner [215].

A set of 50,000 expanded events of a length up to 10 tuples has been generated starting from seed events by replacing one or more terms in an event's tuples by synonyms or related terms from the thesaurus. Example 5.6 represents an event resulting from semantically expanding the seed event in Example 5.4. The latter has different terms used for attributes, e.g. such as 'place' instead of 'room', and different terms used for values, e.g. 'computer' instead of 'laptop'.

**Example 5.6** (Event Resulting from Expansion).

{*type*: power consumption rise event,  
*magnitude unit*: kilowatt per hour, *apparatus*: computer,  
*bureau*: bureau 112c, *place*: room 112, *level*: ground level,  
*area*: building, *metropolitan*: Galway, *homeland*: Ireland, *landmass*: Europe}

#### 5.10.2.5 Generation of an Approximate Subscription Set

An approximate subscription set,  $S$  in Figure 5.5, can be generated from an exact subscription set by introducing the *tilde*  $\sim$  operator into one or more predicates in the exact subscription, the transformation  $P_{X-S}$  in Figure 5.5. This generation can also be guided by: the percentage of predicates parts to be relaxed by the *tilde*  $\sim$  operator that is called the degree of approximation and the semantic measure to be used at the attribute/value part of predicates tuples. Example 5.7 represents an approximate subscription resulting from relaxing 50% of the exact subscription in 5.5 using the *esa* semantic measure.

**Example 5.7** (Approximate Subscription).

{*type* =increased energy consumption event $\sim$  esa,  
*device* $\sim$  esa = laptop $\sim$  esa,  
*floor* = ground floor}

#### 5.10.2.6 Generation of Ground Truth for Approximate Subscriptions and Expanded Events

The goal of this stage is to find the resulting relevance function between approximate subscriptions and expanded events, function  $P_{S-E}$  in Figure 5.5.  $P_{S-E}$  is isomorphic to the basic exact relevance function  $P_{X-SE}$  thus it is an exact relevance function. As

a result, an expanded event is relevant to an approximate subscription if it exactly matches the subscription or a version of it which results from it by replacing the *tilde*  $\sim$  approximated parts with related terms from the thesaurus.

### 5.10.3 Methodology for Efficiency Evaluation

Efficiency evaluation aims to position the proposed approach on the throughput scale with respect to an approach based on an exact matcher, and namely rewriting of rules based on WordNet [224] as a knowledge representation followed by an exact matching. Given a set of approximate subscriptions, each approximate subscription can be rewritten as a set of conjunctive statements, each of which is a set of attribute-value pairs resulting by replacing the approximate parts of a subscription with related terms from the WordNet [224] dictionary. Example 5.8 shows a rewritten statement arising from the approximate subscription in Example 5.7.

**Example 5.8** (Exact Rewritten Statement).

*{type = increased energy use event,*  
*appliance = portable computer,*  
*floor = ground floor}*

### 5.10.4 Results

The following sections explain the experiments that study the top- $k$  algorithm performance, the effects of optimization, approximation degree, and the comparison with the exact model. All experiments have been conducted on a Windows 7 machine, with an Intel Core i7-3520 2.90 GHz CPU and 8GB of RAM running JVM 1.7.

#### 5.10.4.1 Top-k by an Evolving Frontier Algorithm Performance

Figures 5.6, 5.8 and 5.7 show that the algorithm time performance is polynomial and approximately linear with  $k$  and the number of event's tuples  $m$  while it is polynomial and approximately quadratic with the number of subscription's predicates  $n$ . These findings confirm the anticipated contribution of  $n$ ,  $m$ , and  $k$  to the algorithm complexity analysed in Section 5.9.4. They also show that the proposed algorithm is efficient in

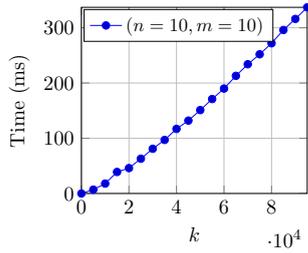
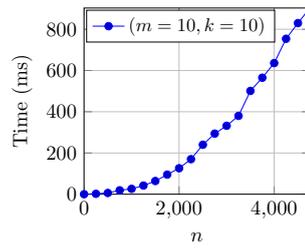
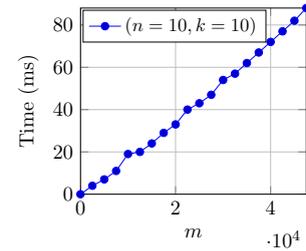
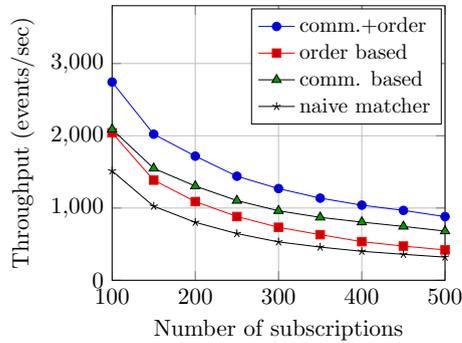
FIGURE 5.6: Top- $k$  time vs.  $k$ FIGURE 5.7: Top- $k$  time vs.  $n$ .FIGURE 5.8: Top- $k$  time vs.  $m$ .

FIGURE 5.9: Optimized matcher

finding the top- $k$  mappings between a subscription and an event. Thus, the efficiency part of hypotheses  $H1$  and  $H4$  is validated for event processing when further processing of single events is needed, e.g. in the case of complex event processing.

#### 5.10.4.2 Optimization Strategies

This experiment has been conducted with 9 sets of 100 – 500 approximate subscriptions of 50% degree of approximation with *esa*. 43% of the predicates on average are unique in the subscriptions. Figure 5.9 shows that a matcher equipped with the commonalities and order optimization strategies outperforms a *naive* matcher for any number of subscriptions with an average optimization of 134%. The commonalities-based matcher and the order-based matcher both outperform the *naive* matcher.

In the selected sample the commonalities-based optimization outperforms the order-based one. That is caused by the relatively high number of shared predicates (about one shared predicate per each two subscriptions). Besides, 50% degree of approximation seems to leave a little to do to the exact matchers for early elimination of subscriptions. The higher proportion of shared-predicates and the lower degree of approximation,

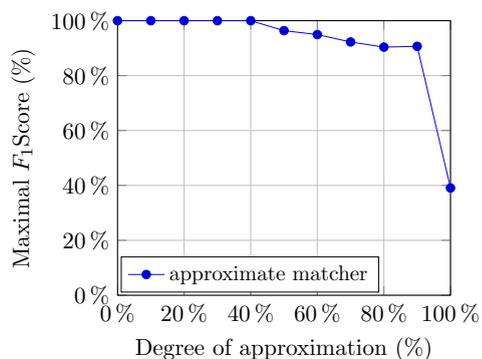


FIGURE 5.10: Effectiveness vs. % of  $\sim$

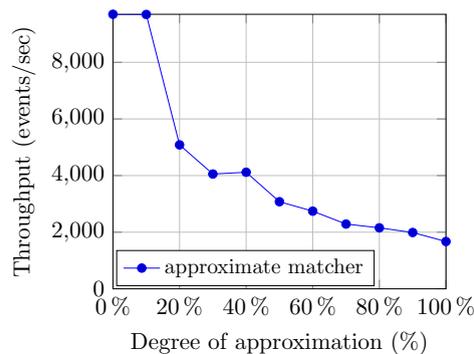


FIGURE 5.11: Efficiency vs. % of  $\sim$ .

the better optimization that shall be achieved by commonalities-based and order-based strategies respectively.

These findings show that the proposed approximate matching model is naturally and effectively extendible by optimization strategies commonly used in event processing. Thus, the efficiency part of hypotheses  $H1$  and  $H4$  is validated for event processing with the single event matching functionality.

#### 5.10.4.3 The Effect of the Degree of Approximation

The experiment has been conducted with 11 sets of increasing degrees of approximation of 100 approximate subscriptions with *esa*. Figure 5.10 shows that the matcher performs well with low degrees of approximation. Effectiveness slightly drops with medium degrees, 90% – 100% F<sub>1</sub>Score with degrees up to 90%. It then sharply drops to 40% when the subscriptions become mostly or fully approximated, i.e. > 90%. That is because exact predicates can better discriminate relevant events and as they disappear in higher degrees of approximation it becomes difficult for the matcher to decide on relevance and F<sub>1</sub>Score drops consequently. Thus, the effectiveness part of hypotheses  $H1$  and  $H4$  is validated for event processing with single event matching functionality, the scope of this work, for a reasonable amount of approximation.

Figure 5.11 shows that throughput decreases sharply from 9,700 events/sec to 5,100 events/sec when approximation starts to appear in subscriptions at around 20% degree of approximation. It then decreases almost linearly from 5,100 events/sec to 1,700 events/sec when the degree increases from 20% to 100%. That is because approximate

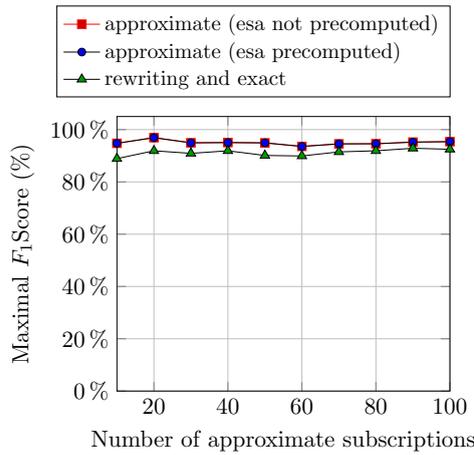


FIGURE 5.12: Effectiveness vs. the number of approximate subscriptions

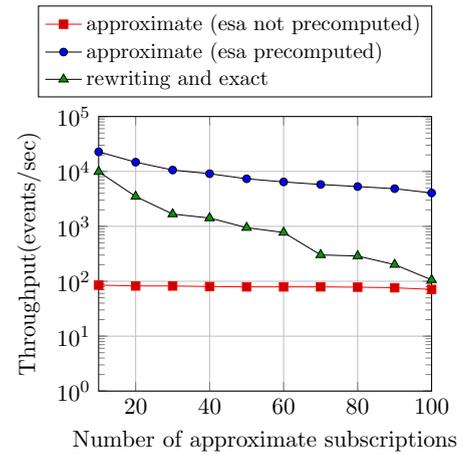


FIGURE 5.13: Efficiency vs. the number of approximate subscriptions

predicates, which increasingly appear in higher degrees, are more time consuming to test than exact comparisons and throughput decreases as a result.

These results suggest that the best use cases for the proposed model are where small-to-medium degrees of approximation are expected with the user having, at least, a partial knowledge of the event semantics. This would be the case for many IoT applications.

The efficiency part of hypotheses  $H1$  and  $H4$  is validated for event processing with single event matching functionality, the scope of this work, with scalability with a reasonable amount of approximation.

#### 5.10.4.4 Comparison to the Exact Model

This experiment has been conducted with 10 sets of 10–100 approximate subscriptions of 50% degree of approximation with *esa*. Figure 5.12 shows that the approximate matcher delivers 94% – 97% matching quality, which is higher than the 89% – 92% delivered by the WordNet-based rewriting approach equipped with exact matching. The rewriting approach outperforms the approximate model in throughput when the pairwise semantic relatedness scores are calculated at run-time. However, the approximate matcher based on precomputed *esa* scores outperforms in throughput with around 91,000 events/sec compared to around 19,100 events/sec on average as shown in Figure 5.13.

In this experiment, around 16 million pairwise comparisons are needed, less than 100,000 of them, i.e. less than 1% of them, need to be calculated just once. That is a valid

TABLE 5.5: Approximate versus Exact Model

	<b>Maximal <math>F_1</math>Score</b>	<b>Number of Subscription Rules</b>	<b>Degree of Approximation</b>	<b>Coupling</b>
<b>Exact Model</b>	100%	74,000	0%	high
<b>Approximate Model</b>	95%	100	50%	low

assumption as pre-computation can happen at the semantic measure side beforehand or when the system caches newly calculated scores, so no re-computation is required. These results show the validity of an approximate model enhanced with a loosely coupled semantic model such as the distributional semantic model to achieve good effectiveness and efficiency as opposed to other approaches based on semantically coupled knowledge representations.

Finally, to achieve the 100% of  $F_1$ Score and a throughput of an exact matcher there is a need to write manually all the possible rules that are equivalent to the approximate rules as shown in Table 5.5. To quantify this situation, I measure how many exact rules are required to compensate for approximate rules given that the rewriting is done with the ground truth thesaurus which is Merriam-Webster. This showed that about 74,000 exact rules are needed to cover all events compared to a maximum of only 100 rules for the approximate matcher. This is a non-feasible situation in semantically heterogeneous environments.

These figures show a trade-off between effectiveness and efficiency on the first hand versus loose semantic coupling and ease of use on the other hand. These results suggest that the proposed approximate event processing model is suitable for scenarios such as the IoT with a high level of semantic heterogeneity and where having complete prior semantic knowledge of events is unfeasible. Thus, the effectiveness and efficiency parts of hypotheses  $H1$  and  $H4$  are validated with clear identifiable loosening in the semantic coupling dimension.

## 5.11 Chapter Summary

This chapter constructed a model that realizes the elements of subsymbolic distributional event semantics and approximation. The rationale for subsymbolic distributional event semantics as a bottom-up, coarse-grained model for semantics has been discussed with

respect to other semantic models such as symbolic and non-symbolic semantic models. The rationale for the approximation element as a model for tackling uncertainty that results from the lack of full semantic agreements has also been discussed.

The approximate semantic event matching model is designed to extend the current event processing paradigm in that:

- Rules are equipped with the *tilde*  $\sim$  semantic approximation operator.
- The single event matcher is equipped with matching and mapping algorithms to detect events semantically relevant to approximate subscriptions. The single event matcher works in two modes: top-1 that forwards the best mapping between an event and a subscription to the consumers; and top- $k$  which results in a list of top- $k$  possible mappings between an event and a subscription. The top- $k$  mappings of various events to various subscriptions go to the complex pattern matcher.
- The complex pattern matcher can then perform a probabilistic reasoning to deduce the probabilities of occurrences of the derived events in the action parts of the complex rules.

A synthetic evaluation framework has been used to evaluate the model as opposed to an exact model, which uses re-writing of rules based on the WordNet thesaurus. The evaluation event set of 50,000 events has been semantically expanded out of seed event sets from actual deployments of IoT, energy management, building, and relevant datasets. Approximate subscriptions are synthesized with the ground truth being updated. Evaluation showed that the approach outperformed the baseline with a throughput of 1,000 events/sec, and over than 95% F<sub>1</sub> Measure of matching quality. Experiments also showed that 100 approximate subscriptions could compensate for 74,000 exact subscriptions otherwise needed, representing a low semantic coupling. Hypotheses *H1* and *H4* with respect to the use of the elements of subsymbolic distributional event semantics and approximation have been validated. The results suggested that the best use cases for the proposed model are where small-to-medium degrees of approximation are expected with the user having, at least, a partial knowledge of the event semantics, which would be the case for many IoT applications.



## Chapter 6

# The Thematic Event Matching Model

“Necessity is the theme and inventress of nature.”

— Leonardo da Vinci

### 6.1 Introduction

This chapter tackles mainly research question *Q1* that states the following:

*Q1.* The first research question is concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rules and their meanings but rather a form of statistical loose agreements on the meanings. The question is how to achieve timely event matching with high true positives and negatives in such a loosely semantically coupled environment?

This chapter tests the hypothesis *H2* that is formulated as the following:

*H2.* Free tagging of events and subscriptions does not add to the cost of defining and maintaining rules with respect to the use of terms, and the cost of building and agreeing on an event semantic model required by subsymbolic

event semantics; and at the same time it can achieve timely event matching with high true positives and negatives more than event processing based on non-tagged subsymbolic event semantics.

To test the hypothesis, this chapter constructs a model that realizes the element of free tagging. A generic model called thematic event matching is proposed and discussed in Section 6.2. This model has been mainly presented in the IEEE Internet Computing (2015) [151], and the International ACM/IFIP/USENIX Middleware Conference (Middleware 2014) [152].

The background on free tagging and the rationale for the proposed model and the hypothesis are detailed in Section 6.3. An instantiation of the model, along with concrete event, language, and matching models are defined in Section 6.4. The new concept of parametric vector spaces along with thematic projection and semantic measures are discussed in Section 6.5.

The constructed model of thematic event matching is then empirically validated in Section 6.6. This chapter shows that the proposed hypothesis  $H2$  is valid. Thus, the element of free tagging can answer the research question and consequently can address the requirements of effective and efficient event processing that is loosely coupled in semantics. The chapter is summarized in Section 6.7.

## 6.2 The Thematic Model

The thematic event matching approach suggests associating representative terms with events and subscriptions to describe the themes of types, attributes, and values and clarify their meanings as shown in Figure 6.1. A theme is a lightweight method to convey semantics when combined with a semantic model such as distributional semantics. At the same time, themes are meant to be used in situations where little or no agreements can be achieved on a fixed taxonomy.

Event publishers associate their events with a number of terms that describe their payload. Subscribers also associate their subscriptions with a number of terms that clarify their interests. If agreements on themes can be achieved, then a theme is decided for each event type. If agreements cannot be assumed, then event publishers and subscribers

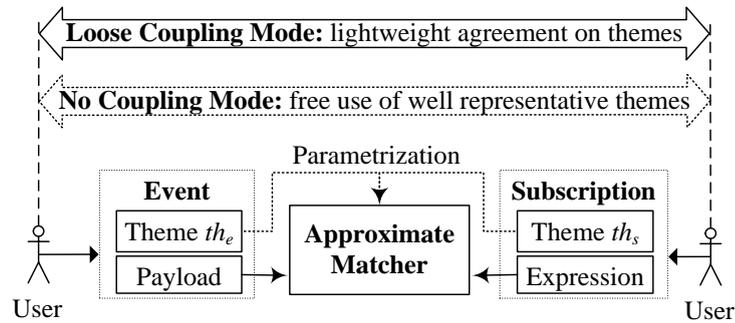


FIGURE 6.1: Thematic event processing

freely add themes that better represent their artefacts. Thematic events can more easily cross semantic boundaries as: (1) they free users from needing a prior semantic top-down agreements and thus enable event exchange across such boundaries, and (2) they carry approximations of events meanings composed of payloads and theme tags which when combined carry less semantic ambiguities. An approximate matcher exploits the associated theme tags to improve the quality of its uncertain matching of events and subscriptions.

This generic architecture applies to various types of event payloads. For example, an event payload can be an image and its theme is a set of tags describing its content like  $\{ 'girl', 'football', 'outdoor' \}$ . A subscription can be an image too associated with a set of tags such as  $\{ 'female', 'soccer', 'play', 'nature' \}$ . The approximate matcher performs an uncertain matching on images based on their pixels and other intrinsic image features. It also exploits the tags associated with the event and the subscription to parametrize its matching algorithm and improve its matching quality. For instance, it weighs up some object recognition candidates more like *'girl'* versus *'boy'* in the event image. Event sources and consumers can either (1) agree to use representative terms when an agreement is possible and thus having a lightweight *loose coupling*, or (2) freely use representative terms in open environments when an agreement is not feasible, thus having *no coupling*.

The generic thematic event processing model is instantiated for structured attribute-value events and subscriptions. The attribute-value model is simple, widely used, and may be used to convey other models. Theme tags are exchanged with the events and utilized by the matcher to more accurately filter a distributional representation of terms in a vector space as discussed in Sections 6.4 and 6.5.

In the following, Section 6.3 gives a detailed account of the concept of free tagging and its role in managing information resources. The section also discusses the rationale for hypothesising that the free event tagging approach can answer the research question *Q1* on the loose semantic coupling. Various aspects of the thematic event matching model instantiation are discussed in the afterwards from Section 6.4 to Section 6.6.

## 6.3 Free Event Tagging

Subsymbolic or non-symbolic communication can be a sound solution to semantic coupling within event processing systems. Nonetheless, humans are still symbolic in nature. This second element of the proposed approach deals with this fact. Tagging is a mechanism by which the humans behind event producers and consumers can add information to events and subscriptions. The proposed approach uses added tags to enhance the meaning exchanged with events via a better symbolic approximation of the non-symbol meaning space.

### 6.3.1 The Web and Social Tagging

Web 2.0 forms a platform over the Web where users are no longer *readers* of HTML rendered pages. Users can contribute with content to Web objects such as images, tweets, and blog posts. One type of contribution is the tagging of such Web objects. Gupta et al. recognize this as a significant trend in online social communities:

“Social tagging on online portals has become a trend now. It has emerged as one of the best ways of associating metadata with web objects. With the increase in the kinds of web objects becoming available, collaborative tagging of such objects is also developing along new dimensions.” [43]

Websites supporting social tagging emerged and became popular as investigated by Breslin et al. [225, 226]. Table 6.1 shows some example social websites with a categorization by type and web resources subject to tagging. The basic concepts within social tagging are three: the users, the resources, and the tags. Out of these, matrices can be built where each cell show what tags  $t$  a user  $u$  uses to tag a resource  $r$ .

TABLE 6.1: Example Social Tagging Websites

Website	Type	Web Resource
Delicious <sup>1</sup>	link sharing	bookmarked URL
Flicker <sup>2</sup>	photo sharing	photo
Blogger <sup>3</sup>	blogging	post
Twitter <sup>4</sup>	micro-blogging	tweet
LibraryThing <sup>5</sup>	cataloguing	book
Digg <sup>6</sup>	social news	news story
YouTube <sup>7</sup>	video sharing	video

Gupta et al. summarize the main incentive behind social tagging:

“Different web portals focus on sharing of different types of objects like images, news articles, bookmarks, etc. Often to enrich the context related to these objects and thereby support more applications like search, metadata needs to be associated with these objects.” [43]

Several motivations for tagging exist such as: future retrieval, contribution and sharing, attracting attention, play and competition, self-presentation, opinion expression, task organization, social signalling, money, and technological ease. Tags can also be of various kinds: content-based tags, context-based tags, attribute tags, ownership tags, subjective tags, organizational tags, purpose tags, factual tags, personal tags, self-referential tags, and tag bundles [43].

### 6.3.2 Metadata Generation and Fixed Taxonomies

While social tagging websites provide users with the tools to tag resources, the question of how taxonomies of tags are created arises. It has been found within the social media research that using fixed static taxonomies is not a suitable approach within the social tagging context as put by Gupta et al. [43] for the following reasons:

1. Centralized fixed taxonomies are rigid.

---

<sup>1</sup><http://www.delicious.com/>

<sup>2</sup><http://www.flickr.com/>

<sup>3</sup><http://www.blogger.com>

<sup>4</sup><http://www.twitter.com/>

<sup>5</sup><http://www.librarything.com/>

<sup>6</sup><http://www.digg.com/>

<sup>7</sup><http://www.youtube.com/>

2. Items do not fit necessarily in just a single category.
3. Hierarchical classifications represent the one view of the world which is the cataloguer's and thus they are subject to bias.
4. Fixed taxonomies do not account for the case when the corpus evolves.
5. Fixed taxonomies for social tagging need cataloguers who think exactly the same way as users.
6. A controlled vocabulary is expensive to build and maintain in terms of development time.
7. Enforcing a controlled vocabulary presents a steep learning curve to users.

Gupta et al. state a summary to these problems:

“This implies a loss of precision, erases difference of expression, and does not take into account the variety of user needs and views.” [43]

These problems add to the cognition-based discussion of semantics in Section 5.4 an important *social dimension*. In a more generalized way, using fixed, centralized, and top-down authoritative semantic models is not scalable within large-scale event processing systems. I argue that such top-down organization of semantic models increase the problem of semantic coupling, which already exists due to the granularity of these models, which are symbolic in nature. A more flexible approach to how semantic models are managed is required to address this aspect of the semantic coupling problem. This element discussed in Section 6.3 builds upon the analogy with the free event tagging approach, called folksonomies, and adopted within social tagging systems.

### 6.3.3 Folksonomies

Folksonomies, (folk (people) + taxis (classification) + nomos (management)), are terms, freely generated by users, and freely used by users to tag resources. This is the type of taxonomy adopted in most social tagging platforms such as those in Table 6.1. Several advantages of folksonomies could be recognized [43]:

1. Folksonomies support entry and cooperation with no barriers.
2. Folksonomies require a very low cognitive cost.
3. Folksonomies are inclusive in terms of having terms related to popular and rare topics.
4. Folksonomies are capable of matching real users' needs and languages.

To this end, I argue that bottom-up free tagging of events and subscriptions be a good way to manage their semantics in a loosely coupled manner. This meets the Requirement *R1* of loose semantic coupling. Beside their scalable nature, folksonomies have proved to be useful to enhance the results of various computing systems. For example Xu et al. [227] showed that using folksonomies for information retrieval significantly improves search quality. This meets the Requirement *R3* of effectiveness.

My interpretation of the effectiveness aspect of folksonomies lies in the interplay between the two spaces of symbols and meanings, made clear in semiotics [143, p. 18–21]. For example, let us take the case of polysemy which means that one word, i.e. symbol, can have multiple meanings. For instance, the word '*energy*' can refer to the meaning of *acting or being active*, or to the *usable power such as electricity*, along with other meanings. If the word *energy* has been associated by the word '*bulb*', then the second meaning is probably meant. This kind of association could be formulated as a type of tagging. The phenomenon of polysemy is the subject of research in Word Sense Disambiguation (WSD) [228].

Thus, tags can be used to provide a better approximation of the meaning space, based solely on the symbol space. I discussed in Section 5.4 how statistical semantics can approach the meaning space using vectors of co-occurring words. I believe that an enhanced approach can be supported using free tags, combined with statistical distributional semantics. Additionally, tags associated with statistical models can filter many unrelated meanings of the words used in events and subscriptions. This meets the Requirement *R4* of efficiency.

#### 6.3.4 Limitations of Free Event Tagging

Folksonomies suffer from three main limitations as discussed by Pan et al. [229]:

TABLE 6.2: Free Event Tagging and Requirements

	<b>Fixed Taxonomy</b>	<b>Free Event Tagging</b>
<i>R1.</i> Loose semantic coupling	-	+++
<i>R2.</i> Loose pragmatic coupling	NA	NA
<i>R3.</i> Efficiency	++	+
<i>R4.</i> Effectiveness	+	++

	+++	the model excellently addresses the requirement
	++	the model moderately addresses the requirement
Legend	+	the model slightly addresses the requirement
	-	the model mildly affects the requirement in a negative way
	NA	the requirement is not in question

1. *Tag variation (synonymy)* which states that two synonyms tagging a web resource could be handled differently by the social tagging system.
2. *Tag ambiguity (polysemy)* which states that the same tag can be used to mean different meanings by two different users.
3. *Flat organization of tags* which states that tags do not have an explicit hyponymy relationship.

I argue that these limitations can be largely reduced in the context of the proposed approach. Those limitations arise mainly due to the lack of a back-end semantic model in classical social tagging systems. Objects in such systems are passive being, for instance, a photo, a URL, etc. I propose the use of tags: (1) to tag other symbols in events and subscriptions, and (2) in association with a statistical semantic model. Thus, merging those together can move the system from the symbolic space to a meaning space, and thus lower the limitations proposed above which emerge mainly due to sticking to a symbolic level only.

### 6.3.5 How Free Event Tagging Meets the Requirements

As discussed throughout Section 6.3, free event tagging addresses the social management aspect of computing systems, inherent to event processing systems that are distributed and decoupled by nature. Fixed and free taxonomies are the main models for enforcing semantic models, but the free event tagging approach appears to be the ideal choice for satisfying the main requirements as summarized in Table 6.2.

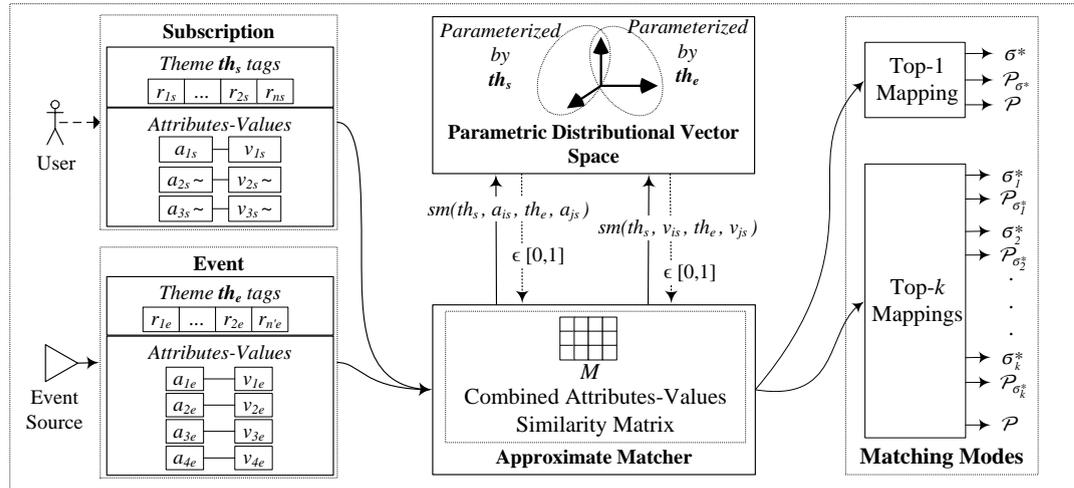


FIGURE 6.2: Thematic event matching model

## 6.4 Model Instantiation

The main elements of the model instantiation are illustrated in Figure 6.2. Let an event of *increased energy consumption* be represented as follows:

{**type**: increased energy consumption event,  
**measurement unit**: kilowatt-hour,  
**device**: computer, **office**: room 112}

In the thematic model, this event is accompanied by a set of key terms that represent approximately the domain and meaning of the event attributes and values. These terms are called the *event theme tags*. An example of terms for the above event are:

{energy, appliances, building}

Similarly, subscriptions are associated with *subscription theme tags*. The proposed model language introduces the *tilde*  $\sim$  operator that signifies that the user wants the matcher to match the term used or any term semantically similar to it. A subscription for *increased energy consumption* can be represented as follows:

{**type**= increased energy usage event $\sim$ ,  
**device** $\sim$ = laptop $\sim$ , **office**= room 112}

Example theme tags for this subscription are:

{power, computers}

The example event and subscription do not use exactly the same terms to describe the type or the device, hence ‘*energy consumption*’ vs. ‘*energy usage*’, and ‘*computer*’ vs. ‘*laptop*’. Nevertheless, the event should not be considered as a negative match to the subscription. For this reason, the model employs an approximate probabilistic semantic matcher as discussed in Chapter 5. It uses a measure to estimate semantic similarity and relatedness between various terms. Functionally, it tries to establish the top-1 or top- $k$  possible mappings between subscription predicates and event tuples along with probability spaces of each predicate-to-tuple and the overall mapping. For example, the most probable mapping of the previous examples, or top-1 mapping, is described as follows:

$$\sigma^* = \{(\mathbf{type=increased\ energy\ consumption\ event} \\ \leftrightarrow \text{type:increased energy usage event}), \\ (\mathbf{device\sim = laptop\sim} \leftrightarrow \text{device:computer}), \\ (\mathbf{office = room\ 112} \leftrightarrow \text{office: room 112})\}$$

The approximate matcher uses a semantic measure to estimate semantic similarity and relatedness between each pair of attributes or values from the subscription and the event. The matcher then combines that in a similarity matrix that encodes similarity between all possible pairs of subscription predicates and event tuples. The model proposes the use of a semantic measure based on distributional semantics as described in Section 6.4.1. While typical semantic measures take as input two terms and return a value in  $[0, 1]$ , the thematic matcher passes the subscription and event themes as additional parameters along with the terms. The themes are used to adapt the terms meaning vector space before the actual semantic distance is measured as described in Section 6.5.

### 6.4.1 Distributional Semantics

Distributional models are useful for the task of assessing semantic similarity and relatedness between terms. A Wikipedia-based Explicit Semantic Analysis (ESA) builds an index of words based on the Wikipedia articles they appear in as shown in Figure 6.3. A word becomes a vector of articles and the more articles that are common between

two words, the more related the words are. For example,  $esa('parking', 'garage') > esa('parking', 'energy')$  as the first pair frequently appear in common articles. Typically semantic relatedness between a pair of terms is measured using cosine or Euclidean distance between the two vectors representing the two terms. In the proposed thematic parametric vector space model, the *esa* measure is parametrized with the theme tags. They are used to project the term vectors to get more domain-specific meaning vectors and then are passed to the distance function as illustrated in Figure 6.3 and detailed in Section 6.5.

### 6.4.2 Themes

A theme is defined as a set of terms that describe the content of an event or a subscription. For instance, the set  $\{'energy', 'appliances', 'building'\}$  refers to an event that conveys energy consumption of appliances in a building. A theme combines with the actual content to form an approximation of the meaning of concepts. It is meant to be exchanged in addition to the actual symbols, i.e. words, used to represent attributes and values. I build in this chapter the thematic vector space model on top of the ESA vector space developed by Freitas et al. [174, 175, 200].

### 6.4.3 Thematic Event Model

The event model used in the thematic event model is an attribute-value model, but the discussion is as relevant to other models such as hierarchical or graph-based event models. Each event is a pair of sets: a set of theme tags and a set of tuples. Each theme tag is a single-word or a multi-word term. Each tuple consists of an attribute-value pair. No two distinct tuples can have the same attribute. An example energy consumption event is represented as follows:

$(\{energy, appliances, building\},$   
**{type:** increased energy consumption event,  
**measurement unit:** kilowatt-hour,  
**device:** computer, **office:** room 112})

The formal definition of the event model is as follows: let  $E$  be the set of all events, let  $TH$  be the set of all possible theme tags, and let  $A$  and  $V$  be the sets of possible attributes and values respectively. Let  $AV$  be the set of possible attribute-value pairs, i.e. tuples, such that  $AV = \{(a, v) : a \in A \wedge v \in V\}$ . An event  $e \in E$  is a pair  $(th, av)$  such that  $th \subseteq TH$  and  $av \subseteq AV$  are the set of theme tags and the set of tuples respectively.

#### 6.4.4 Thematic Language Model

Each subscription is a pair of two sets: a set of theme tags and a set of conjunctive attribute-value predicates. Each theme tag is a single-word or a multi-word term. Each predicate uses the equality operator to signify exact equality or approximate equality when indicated. Other Boolean and numeric operators such as  $! =$ ,  $>$ , and  $<$  are kept out of the language for the sake of discourse simplicity. Each predicate consists of an attribute, a value, and specifications of the semantic approximation for the attribute and the value. The most notable feature of the language is the *tilde*  $\sim$  operator that helps specify the approximation for an attribute/value when it follows it. An example subscription to energy usage events is as follows:

( $\{power, computers\}$ ,  
 {**type**= increased energy usage event $\sim$ ,  
**device** $\sim$ = laptop $\sim$ , **office**= room 112})

The author of the subscription specifies that the device can be a *'laptop'* or something related semantically to *'laptop'*. The subscription also states that the attribute *'device'* itself can be semantically relaxed. However, it states that the event's *'office'* must be exactly *'room 112'*, etc.

The formal definition of the language model is as follows: let  $S$  be the set of subscriptions, let  $TH$  be the set of all possible theme tags, and let  $A$  and  $V$  be the sets of possible attributes and values respectively which can be used in a subscription. Typically there are no restrictions on  $A$  or  $V$  and the user is free to use any term or combination of terms. Each predicate is a quadruple which consists of the attribute, the value, and whether or not the attribute/value is approximated. Let  $P$  be the set of possible predicates, thus  $P = \{p : p = (a, v, app_a, app_v) \in A \times V \times \{0, 1\}^2\}$ . A subscription  $s \in S$  is a pair

$(th, pr)$  where  $th \subseteq TH$  and  $pr \subseteq P$  are the set of theme tags and the set of predicates respectively. The *degree of approximation* is the proportion of relaxed attributes and values. An exact subscription has 0% degree of approximation.

### 6.4.5 Thematic Matching Model

The thematic matching model builds upon the approximate probabilistic model detailed in Chapter 5. An approximate semantic single event matcher  $\mathcal{M}$  decides on the semantic relevance between a subscription  $s$  and an event  $e$  based on the semantic mapping between attribute-value predicates of  $s$  and attribute-value tuples of  $e$ . An example mapping between the event in Section 6.4.3 and the approximate subscription in Section 6.4.4 is as follows:

$$\begin{aligned} \sigma = & \{(\mathbf{type=increased\ energy\ consumption\ event} \\ & \leftrightarrow \text{type:increased\ energy\ usage\ event}), \\ & (\mathbf{device\sim =\ laptop\sim} \leftrightarrow \text{device:computer}), \\ & (\mathbf{office =\ room\ 112} \leftrightarrow \text{office: room\ 112})\} \end{aligned}$$

$\mathcal{M}$  works in two modes: the top-1 mode that decides on the most probable mapping between  $s$  and  $e$ , and the top- $k$  mode which decides on the top- $k$  probable mappings to be used later for complex event processing. It has been shown in [172] that producing the top- $k$  mappings increases the chance of hitting the correct mapping.

The formal definition of matching is as follows: let  $C = s \times e$  be the set of all possible correspondences between predicates of  $s$  and tuples of  $e$ .  $\forall c = (p, t) \in C \Rightarrow p \in s \wedge t \in e$ .  $\Sigma = 2^C$  is the power set of  $C$  and represents all the possible mappings between  $s$  and  $e$ . There are exactly  $n$  correspondences in any valid mapping  $\sigma$  where  $n$  is the number of predicates in the subscription  $s$ .

For any valid mapping  $\sigma$  a probability function quantifies the probability of every predicate-tuple correspondence  $(p, t) \in \sigma$  such as (**device = laptop $\sim$**   $\leftrightarrow$  device: computer). There also exists a probability function that quantifies the probability of the overall mapping  $\sigma$  among other possible mappings. Both functions form probability spaces  $\mathcal{P}_\sigma$  and  $\mathcal{P}$ . In this work, all probabilities are calculated based on the combined similarity matrix that is based on the thematic pairwise attributes or values semantic

relatedness scores. Thematic semantic relatedness measure is discussed in Section 6.5. For more details on the generic matcher model and detailed evaluation of top-1 and top- $k$  modes, please refer to Chapter 5.

## 6.5 Parametric Vector Space Model

I introduce the concept of a Parametric Vector Space Model (PVSM). Vector space models are widely used in information retrieval and known to be computationally efficient. Thus, I propose an extension suitable for event processing where time efficiency is a requirement. Figure 6.3 shows the main elements of the parametric space. Building the PVSM is identical to building the non-thematic distributional space model based on indexing the corpus. Nonetheless, vectors in PVSM are projected into thematic dimensions, which are passed as parameters before the vectors are used, as discussed in the following sections.

### 6.5.1 Distributional Vector Space Model

Given a set of documents  $D$ , each document is tokenized into terms, stop words are removed, and an inverted index is built to have an entry for each term [175], Step 1 in Figure 6.3. The inverted index encodes a vector space model whose basis is the set of unit vectors that represent the documents, i.e.  $\{\vec{d}_i : d_i \in D\}$ . Each term  $t$  is then represented as a weighted vector  $\vec{v}_t$  in the vector space as shown in Equation 6.1.

$$\vec{v}_t = \sum_{i=1}^{i=|D|} w_{ti} \vec{d}_i \quad (6.1)$$

The Term Frequency Inverse Document Frequency (TF/IDF) weighting scheme is used. It gives more weight to a term if it appears more often in a document and less often in other documents than another term. It is important to keep the raw  $tf$  and  $idf$  values for each pair (term, document) in the inverted index so they can be used later for thematic projection. The TF/IDF scheme is shown in Equations 6.2, 6.3, and 6.4.

$$tf(t, d) = 0.5 + \frac{0.5 \times freq(t, d)}{\max\{freq(t', d) : t' \in d\}} \quad (6.2)$$

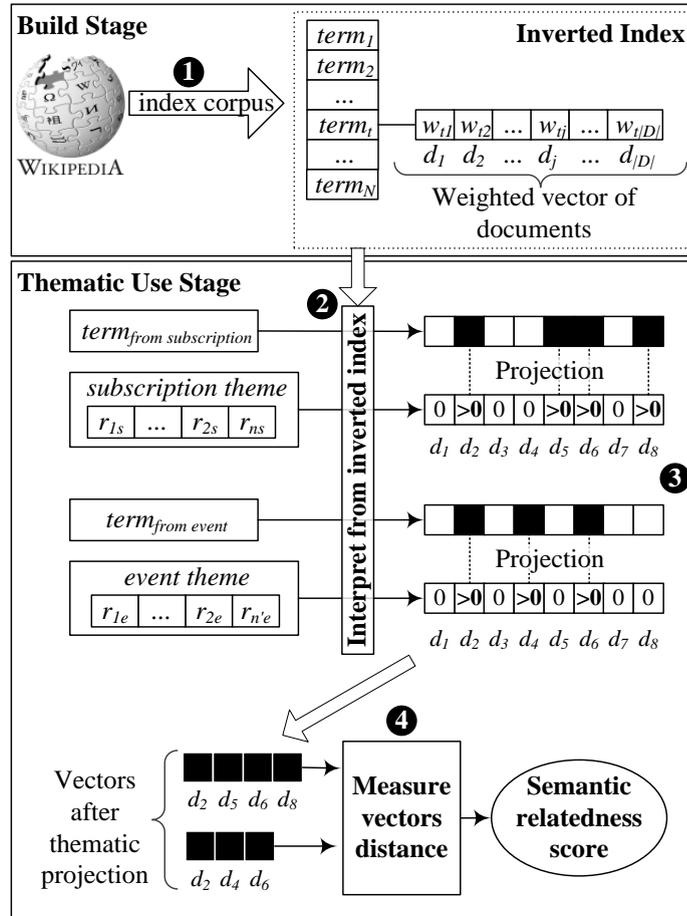


FIGURE 6.3: Parametric distributional vector space for thematic event processing

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \quad (6.3)$$

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) \quad (6.4)$$

### 6.5.2 Thematic Projection

At the usage stage, the ultimate goal is to measure the semantic relatedness between two terms  $term_s$  and  $term_e$  given the subscription and the event themes  $th_s$  and  $th_e$  respectively. Given a term and a theme, the key operation is to use the theme to filter the space into a thematic subspace. The basis of the thematic space is the set of documents that define the themes representative tags.

The thematic basis can be derived by getting the vector representation of the theme, Step 2 in Figure 6.3, and then the documents where its weights are greater than 0, Step 3 in Figure 6.3. Given the new basis, the term vector is transformed to have 0

components for documents not in the thematic basis and to have new  $tf/idf$  weights for the basis documents as the overall number of documents is now different from  $|D|$ . These steps are shown in Algorithm 2. Projection can be computed in  $O(|D|)$  time if all vectors components are stored and  $O(|V|)$  where  $V$  is the non-zero components if only those are stored in the index.

---

**ALGORITHM 2:** Thematic projection
 

---

**Input:** a term  $t$ , a set of theme tags  $th$ , parametric distributional vector space  $PVSM$ 
**Result:** thematic projection vector  $t_{th}$  of  $t$  given  $th$ 

```

1 begin
2    $\vec{t} \leftarrow$  distributional vector of  $t$  from  $PVSM$ ;
3    $\vec{th} \leftarrow$  distributional vector of  $th$  from  $PVSM$ ;
4   for  $d \in D$  s.t.  $\vec{th}_d = 0$  do
5      $t_{th_d} \leftarrow 0$ ;
6   end
7   for  $d \in D$  s.t.  $\vec{th}_d > 0$  do
8      $tf \leftarrow$  original  $tf(t, d)$  from  $PVSM$ ;
9      $idf \leftarrow \log \frac{|\{d \in D: \vec{th}_d > 0\}|}{|\{d \in D: \vec{th}_d > 0 \wedge t_d > 0\}|}$ ;           /* recalculate idf */
10     $t_{th_d} \leftarrow tf \times idf$ ;                                       /* update weight */
11  end
12  return  $\vec{t}_{th}$ ;
13 end

```

---

### 6.5.3 Distance and Semantic Relatedness

Let  $T$  be the set of terms, and  $TH$  the set of all possible thematic tags. The semantic measure  $sm$  is defined as a function that operates on a pair of terms associated with their themes such that  $sm : T \times 2^{TH} \times T \times 2^{TH} \rightarrow [0, 1]$ . Given two terms  $t_s$  and  $t_e$  from a subscription and an event respectively and their associated themes  $th_s \in 2^{TH}$  and  $th_e \in 2^{TH}$  respectively,  $sm$  works by finding the thematic projections  $t_{sth_s}^{\vec{}}$  and  $t_{eth_e}^{\vec{}}$  and then calculating the vector distance between the resulting projected vectors, Step 4 in Figure 6.3.

The *Euclidean* distance to measure projected vectors distance is used as defined in Equation 6.5.

$$dis(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{i=|D|} (\vec{a}_i - \vec{b}_i)^2} \quad (6.5)$$

Semantic relatedness is estimated to be the opposite of the distance and can be calculated as defined in Equation 6.6.

$$\text{relatedness}(\vec{a}, \vec{b}) = \frac{1}{\text{dis}(\vec{a}, \vec{b}) + 1} \quad (6.6)$$

The more filtering that occurs during thematic projection due to smaller themes, the less time is required for computing relatedness.

## 6.6 Evaluation

To evaluate the thematic approach, I compare it with non-thematic approximate semantic event processing, specifically the approach from Chapter 5. As the thematic model already uses the elements of subsymbolic distributional semantics and approximation, and as Chapter 5 validated hypotheses *H1* and *H4*, this evaluation herein adds to the power of hypothesis testing done in the previous chapter. Besides, to test the hypothesis *H3*, I evaluate to what extent this model is effective and efficient, and to what extent it loosens the semantic coupling.

In Chapter 5 the non-thematic approximate approach was compared with a concept-based approach that uses query rewriting using WordNet [224]. Experiments were conducted with 10 sets of 10–100 approximate subscriptions of 50% degree of approximation with *esa*. Results show that the approximate matching model delivers 94%–97% matching quality, higher than the 89%–92% delivered by the WordNet rewriting approach.

The rewriting approach outperforms the approximate approach in throughput when the pairwise semantic relatedness scores are calculated at run-time. However, the approximate model based on precomputed *esa* scores outperforms in throughput with around 91,000 events/sec compared to around 19,100 events/sec on average. Distributional semantics-based approximation is based on a very loose model of semantic coupling which scales to heterogeneous environments. That is different from the case of rewriting that is based knowledge bases. Building knowledge bases is time-consuming and establishing agreements is granular and difficult to achieve.

In this chapter, a large event set is generated with a particular theme as well as a set of subscriptions which assume no semantic agreements and 100% degree of approximation.

The thematic matcher is compared with the non-thematic matcher when different theme tags are used. Evaluation metrics and a detailed methodology are described in the following sections.

### 6.6.1 Evaluation Metrics

Evaluation metrics are similar to the ones used in Chapter 5. They can be classified into two categories: effectiveness and efficiency metrics [214]. Effectiveness metrics measure the quality of event matching. Table 5.3 shows the base concepts needed for evaluating effectiveness. *Precision*, *Recall*, and the combined *F<sub>1</sub>Score* have been used for effectiveness evaluation. *Precision* measures the proportion of relevant events discovered by the matcher with respect to all the discovered events such that  $Precision = TP / (TP + FP)$ . *Recall* measures the proportion of relevant events discovered by the matcher with respect to all the known relevant events from the ground truth such that  $Recall = TP / (TP + FN)$ . Precision and recall are calculated for the whole set of subscriptions by averaging the precision and recall achieved for all individual subscriptions respectively.

The *F<sub>1</sub>Score* equally combines *Precision* and *Recall* such that  $F_1Score = (2 \times Precision \times Recall) / (Precision + Recall)$ . *F<sub>1</sub>Score* is computed at 11 recall points,  $\{0, 0.1, 0.2, \dots, 1.0\}$ , to cover all the precision-recall curve without using thresholds and the maximal *F<sub>1</sub>Score* is then used. The metric used for evaluating time efficiency is *Throughput* defined as  $Throughput = (Number\ of\ processed\ events) / (Time\ unit)$ .

Additionally, to measure the loosening in semantic coupling, I use the measures of alternative *number of exact subscription rules* that would be needed in a coupled model, the *degree of approximation* used in the approximate subscriptions, and the *amount of tagging* needed. These three measures reflect to a large extent the loosening in coupling as they represent the effort and agreements assumed by users. These measures are compared to an exact model's numbers that would typically have a large number of exact subscriptions, which have zero degree of approximation as a result of coupling and need no tagging.

TABLE 6.3: Sensor Capabilities

Sensor Capabilities
solar radiation, particles, speed, wind direction, wind speed, temperature, water flow, atmospheric pressure, noise, ozone, rainfall, parking, radiation par, CO, ground temperature, light, NO <sub>2</sub> , soil moisture tension, relative humidity, energy consumption, CPU usage, memory usage

## 6.6.2 Methodology

The evaluation methodology for effectiveness is similar to the methodology used in Chapter 5. It is outlined in Figure 6.4 and is based on schema matching methodologies [214] concerned with finding the best mapping between a source schema and a target schema. For event matching, approximate subscriptions and events are used. Specifying the ground truth mappings is a challenge for large sets of events and subscriptions.

In recent years, there has been a trend towards synthetic evaluation [214]. Similarly to the idea in STBenchmark [216], I start with pairs of exact subscriptions and events with a known ground truth which is simply the result of exact matching. A semantic expansion transformation is then applied to the events and the subscriptions based on a thesaurus, similarly to the synonyms transformation in eTuner [215]. The ground truth is updated accordingly. The *EuroVoc*<sup>8</sup> thesaurus is used for themes and ground truth generation as it has many domains and can be used for semantic expansion according to specific themes. EuroVoc is a multilingual and multidisciplinary thesaurus that provides common lexis to cover the activities of the European Union.

### 6.6.2.1 Generation of the Seed Event Set

To create a heterogeneous IoT environment, a dataset of events using a set of real-world datasets has been established. Seed events have been synthesized from a set of IoT sensors identical to those deployed in the SmartSantander smart city project [50] and the Linked Energy Intelligence (LEI) dataspace [218]. SmartSantander proposes a city-scale experimental research testbed for IoT applications and services based on sensors deployed in a set of European cities. The LEI project targets smart buildings for energy saving purposes. The used sensor capabilities are shown in Table 6.3.

<sup>8</sup>©European Union, 2014, <http://eurovoc.europa.eu/>

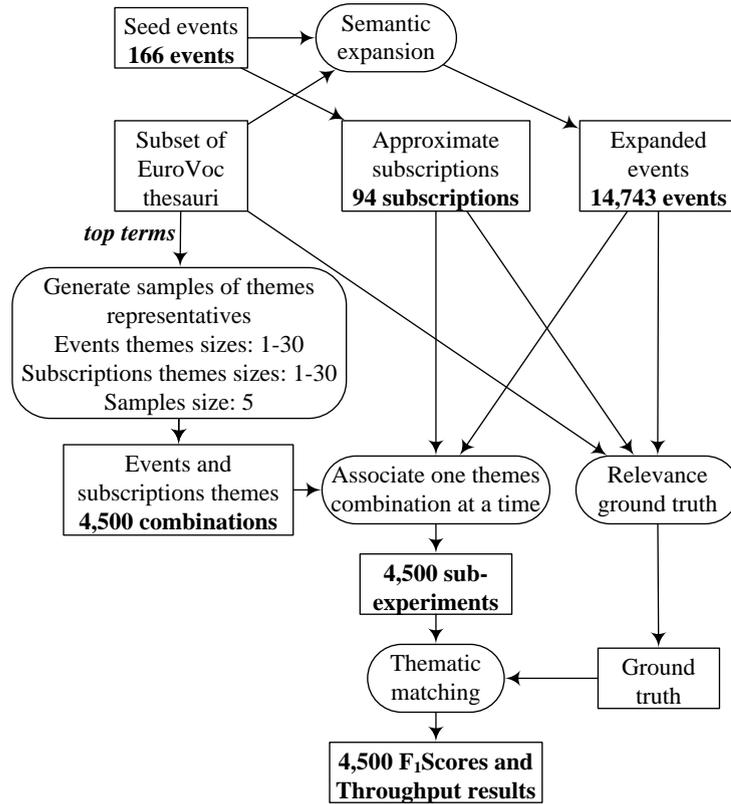


FIGURE 6.4: Evaluation methodology for thematic event processing

A set of car brands from the Yahoo! directory [221] is used to generate vehicle mobile sensors platforms. A set of appliances from the BLUED KDD dataset is used as indoor platforms [222]. For indoor locations, rooms from the DERI Building [223] have been used. For geographical locations, the SmartSantander project locations, as well as Galway City, have been used. Seed events are generated by randomly combining various attributes and values from the aforementioned datasets. A set of 166 seed events has been used. An example seed event generated is as follows:

```
{type: increased energy consumption event, measurement unit: kilowatt-hour,
device: laptop, desk: desk 112c, room: room 112, zone: building,
city: Galway, country: Ireland, continent: Europe}
```

### 6.6.2.2 Semantic Expansion of Seed Events

The purpose of semantic expansion of seed events is to generate a large number of events for evaluation where semantic heterogeneity holds. The EuroVoc thesaurus has been used and specifically its micro-thesauri belonging to domains ‘*transport*’, ‘*environment*’,

TABLE 6.4: Thematic Model versus Exact Model

	Maximal $F_1$ Score	Number of Subscriptions	Degree of Approximation	Number of Tags	Coupling
<b>Exact Model</b>	100%	48,000	0%	0	high
<b>Thematic Model</b>	62% – 85% (effective region)	94	100% (worst case scenario)	2 – 15 (effective region)	low

‘energy’, ‘geography’, ‘education and communications’, and ‘social questions’. This is because those micro-thesauri conform to the theme of the events used in the experiments. A set of 14,743 expanded events of a length up to 10 tuples has been generated starting from seed events by replacing one or more terms in an event’s tuples by synonyms or related terms from the thesaurus. An example event resulting from semantically expanding the seed event in Section 6.6.2.1 is as follows:

{**type**: increased energy consumption event,  
**measurement unit**: kilowatt-hour, **device**: laptop,  
**desk**: desk 112c, **room**: room 112, **zone**: building,  
**urban area**: Galway, **country**: Eire, **continent**: European countries}

### 6.6.2.3 Generation of Approximate Subscription Set and Ground Truth

A set of 94 exact subscriptions is generated by randomly picking a number of tuples from the seed events. A set of 94 approximate subscriptions is then generated by introducing the *tilde*  $\sim$  operator into all the predicates in the exact subscriptions to exclude the non-approximation effect. These are equivalent to about 48,000 subscriptions that would be needed by a non-approximate approach to cover events heterogeneity as shown in Table 6.4, reflecting a loose semantic coupling by the thematic matcher. Effectiveness and efficiency parts of  $H2$ , which correspond to the remainder of Table 6.4, are discussed later in Section 6.6.3 on results. An example approximate subscription resulting from relaxing all predicates of the exact subscription is as follows:

{**type** $\sim$ : increased energy consumption event $\sim$ ,  
**device** $\sim$ : laptop $\sim$ , **floor** $\sim$ : ground floor $\sim$ }

The resulting relevance function between approximate subscriptions and expanded events is isomorphic to a basic exact ground truth function between exact subscriptions and seed events. Thus, it is an exact relevance function. As a result, an expanded event is relevant to an approximate subscription if it exactly matches the subscription or a version of it which results from it by replacing the approximated parts with related terms from the thesaurus used for semantic expansion.

#### 6.6.2.4 Generation of Theme Tags

The target of this step is to associate events and subscriptions with tags. EuroVoc has *top terms* for each of its micro-thesauri. The top terms associated with the domains ‘*transport*’, ‘*environment*’, ‘*energy*’, ‘*geography*’, ‘*education and communications*’, and ‘*social questions*’ that are originally used to expand the event set, are randomly picked. For each sub-experiment, two sets of representative tags are chosen to represent the subscriptions theme and the events theme. The purpose is to study the behaviour of the thematic matcher with different combinations of themes tags. An example subscription theme tags from EuroVoc of size 2 is  $\{\textit{land transport, protection of nature}\}$ .

Given the events and subscriptions sets, various combinations of theme tags have been associated to them. For each combination, there is a sub-experiment that gives an F<sub>1</sub>Score and a throughput result. In every combination, the event theme tags set contains the subscription theme tags set or vice versa. Each combination is defined by the size of the event and the subscription themes. For example, a 3 – 2 combination means that the event theme contains 3 terms while the subscription theme contains 2 terms and the former contains the latter.

For each combination of sizes, a random sample of 5 different pairs of theme tags sets is used. The experiment has been conducted with different sizes of 1 to 30 tags for subscriptions and 1 to 30 tags for events. This gives  $30 \times 30 \times 5 = 4,500$  sub-experiments. The thematic matcher was executed in each sub-experiment to give F<sub>1</sub>Score and throughput results. The choice of the sample size is due to the high dimensionality of the experiments, which poses practical constraints. Future work shall use more resources to allow experimentation with larger samples.

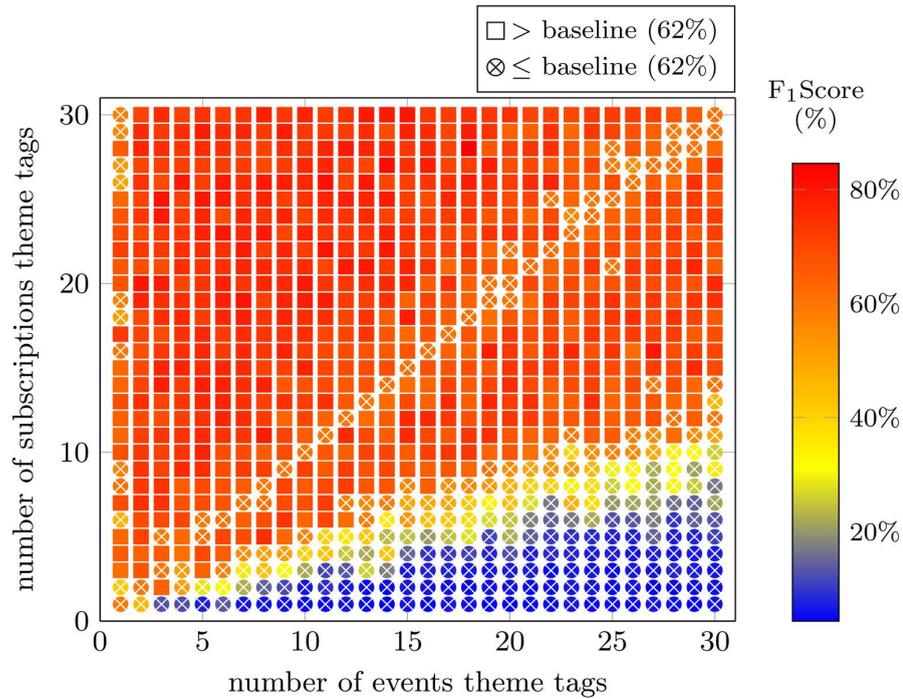


FIGURE 6.5: Effectiveness of thematic matcher

### 6.6.2.5 Baseline

Given the generated events and subscriptions sets, a non-thematic approximate matcher with domain-independent *esa* from Chapter 5 has been used. The matcher gives 62% of F<sub>1</sub>Score and a throughput of 202 events/sec averaged over 5 runs which represents its worst case due to full approximation by using  $\sim$  on all subscriptions predicates.

## 6.6.3 Results

The following sections discuss the effectiveness and efficiency results. All experiments have been conducted on a Windows 7 machine, with an Intel Core i7-3520 2.90 GHz CPU and 8GB of RAM running JVM 1.7.

### 6.6.3.1 Effectiveness

Each cell in Figure 6.5 represents the average F<sub>1</sub>Score of the sample of 5 sub-experiments, each of which uses a different combination of events and subscriptions themes tags. For instance, the sub-experiments of the cell in the 2<sup>nd</sup> column and 10<sup>th</sup> row from the bottom left, all use 2 terms to describe events theme, and 10 terms to describe subscriptions

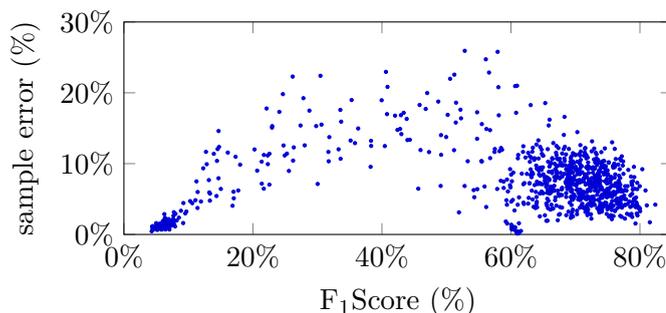


FIGURE 6.6: Effectiveness sample error

theme and the event theme terms set is a subset of the subscription theme terms set. The sub-experiments of the cell in the 10<sup>th</sup> column and 10<sup>th</sup> row from the bottom left, all use 10 terms to describe events theme and 10 terms to describe subscriptions theme and the event theme terms are the same as the subscription theme terms. Square cells are sub-experiments that exceed the baseline while circular ones score below the baseline. Cell colour reflects the average F<sub>1</sub>Score for the sample of combinations for that cell. Colours range from blue (low F<sub>1</sub>Score) to red (high F<sub>1</sub>Score).

Figure 6.5 shows that thematic matching outperforms non-thematic matching in F<sub>1</sub>Score for more than 70% of combinations with scores 62% – 85% and an average of 71% versus 62% for the baseline. Those are more concentrated in the upper left two-thirds of Figure 6.5. F<sub>1</sub>Score on the diagonal line is also slightly lower for the thematic matcher, 59% – 62% versus 62%, suggesting that the projection stage of the vector space by the same tags seems to be less discriminative as opposed to using different tags which could disambiguate attributes/values better.

Thematic matching performs worse when the number of thematic tags is very small, e.g. using just one term as a theme tag. Also, in the bottom triangular half of the figure with F<sub>1</sub>Score widely ranging from 4% to 62%. Larger themes for subscriptions quickly improve effectiveness as opposed to an opposite effect by event themes. That reflects the asymmetric relationship between the many heterogeneous events versus fewer subscriptions. Thus, more terms are needed in subscription themes to discriminate relevant events.

Figure 6.6 illustrates the standard deviation (standard error) of the samples conforming to each set of 5 combinations. The average standard error is 7% of F<sub>1</sub>Score in effectiveness. Most of these errors are around sub-experiments of medium F<sub>1</sub>Scores where it

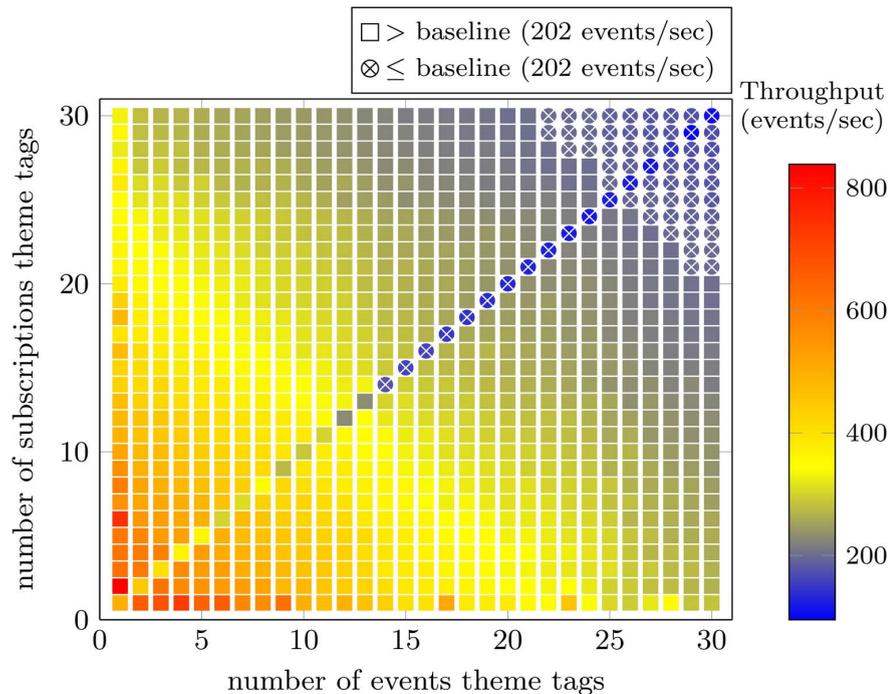


FIGURE 6.7: Throughput of thematic matcher

reaches values around 10% – 25%. Very small errors are concentrated around the sub-experiments of very low  $F_1$  Scores but those are not of concern as theme combinations conforming to such areas of Figure 6.5 should be avoided. More importantly, error converges to smaller values around 7% for sub-experiments of high  $F_1$  Scores which mainly exceed the baseline. This suggests that the experiments are more predictive for higher  $F_1$  Scores and the areas of Figure 6.5 which outperforms the non-thematic approach are more probable to outperform it in other samples.

### 6.6.3.2 Time Efficiency

Figure 6.7 shows the average throughput for each combination of events and subscriptions theme tags. It suggests that the thematic approach outperforms the non-thematic matcher for more than 92% of the sub-experiments, with a throughput of 202 – 838 and an average of 320 versus 202 events/sec. Better throughput is due to the thematic filtering of the space during the thematic projection phase, which saves time during semantic relatedness calculation. This has less effect given more tags towards the top right corner with throughput as low as 95 events/sec.

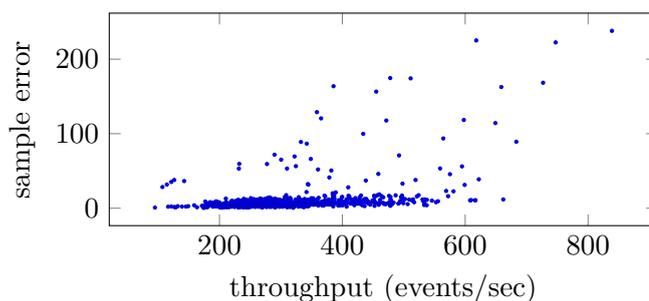


FIGURE 6.8: Throughput sample error

Figure 6.7 shows that throughput decreases gradually when larger sets of theme tags are used to describe events and subscriptions due to less thematic filtering. The last half of the diagonal line shows a drop in throughput, 95 – 177 versus 202 events/sec, as two equal sets of thematic tags for events and subscriptions causes more common dimensions for the semantic measure to be calculated and thus more time is needed for the calculation.

Figure 6.8 shows that few sub-experiments outliers (around 5%) have high standard deviation ranging from 20 to 240 events/sec. The outliers can be explained by rare thematic tags that do not exist in the original indexed corpus. That causes the space to be filtered completely to zero dimensionality and results in a very different time consumption behaviour from other combinations in the same sample. This causes higher errors and less predictability. However, most other sub-experiments have a standard error around the average of 10 events/sec, which is small compared to the overall throughput. Most of the small errors are identified around sub-experiments with throughput from 200 – 600 events/sec, which is mainly above the non-thematic baseline. This small error shows that throughput results are predictive and should be expected in other samples of subscriptions and events theme combinations.

In Chapter 5 I discussed lower degrees of approximation when some agreements can be assumed, and throughput of a magnitude of thousands of events/sec was achieved. Experiments here represent a worst case scenario with 100% degree of approximation and were conducted on a single laptop. There can be further opportunities to optimize the matcher with commonalities, evaluation order, caching, and indexing techniques to improve efficiency.

### 6.6.3.3 Discussion

Results show that the thematic approach is limited when users can provide only a small number of tags for subscriptions, and when hard real-time deadlines are required. Otherwise, results suggest that the use of fewer terms to describe events, around 2 – 7, and more to describe subscriptions, around 2 – 15, can achieve a good matching quality and throughput together with less error rates. That is concentrated in the middle to the upper left side of Figures 6.5 and 6.7. This result shows that events and subscriptions can be associated with only a lightweight number of thematic tags.

For containment between subscriptions themes and events themes to hold, it can be handled in two ways:

- Event sources and consumers loosely agree on terms to use which guarantee containment but causes some semantic coupling.
- Event sources and consumers use more theme tags when no agreement can be achieved in vastly open and decoupled scenarios. Containment and overlap can be assumed to hold due to the distribution of term usage by humans where some terms are more probable to be used by both parties.

Thus, the effectiveness and efficiency parts of hypothesis *H2* are validated with identifiable loosening in the semantic coupling dimension represented by a low amount of required free tagging.

## 6.7 Chapter Summary

This chapter constructs a model that realizes the element of free event tagging. The rationale for using free tagging as loosely coupled mode of improving information content has been discussed. A generic model called thematic event processing was proposed with an instantiation based on structured attribute-value events and subscriptions. The proposed approach suggests associating representative terms, called themes or thingonomies, which describe the themes of types, attributes and values and clarify their meanings. Thematic events can more easily cross semantic boundaries as: (1) they free

users from needing a prior semantic top-down agreements, and (2) they carry approximations of events meanings composed of payloads and theme tags which when combined carry less semantic ambiguities. An approximate matcher exploits the associated thematic tags to improve the quality of its uncertain matching of events and subscriptions.

The concept of vector space semantic models was extended with the idea of parametric vector spaces. A vector that represents a term is adapted by the vector, which represents its thematic tags, in a process called thematic projection. The resulting new vector represents the modified meaning of the original term. Those new thematic vectors are used for actual approximate matching.

The model has been evaluated with a synthetic evaluation framework and compared to a non-thematic matcher. For the evaluation dataset, 14,743 events and 94 approximate subscriptions were used from IoT and energy management domains. Tags are associated with events and subscriptions based on the EuroVoc thesaurus. Experiments showed that a lightweight amount of tags to describe events, around 2 – 7, and subscriptions, around 2 – 15 is needed. That reflects a loose semantic coupling model. The evaluation also showed that the thematic matcher outperformed the baseline with a throughput magnitude of 800 events/sec and 85%  $F_1$ Measure in the worst case of full approximation. As a result, the hypothesis *H2* on the use of free event tagging for loose semantic coupling in event systems has been validated.

## Chapter 7

# The Dynamic Native Event Enrichment Model

“No knowledge is completed except by knowledge of  
its accidents and accompanying essentials.”

— Avicenna

### 7.1 Introduction

This chapter mainly tackles the second research question *Q2* that states the following:

*Q2.* The second research question is concerned with the case when event producers and consumers do not have equal assumptions on the amount of contextual information included in events and how much they are complete with respect to evaluating some consumers’ rules. The question is how to complement events with context at high precision and completeness needed to meet consumers expectations in such a loosely contextually coupled environment?

This chapter tests the hypothesis *H3* and the pragmatic part of hypothesis *H4* that are formulated as follows:

- *H3.* Dynamic native event enrichment decreases the cost needed to define and maintain the context parts of rules, and to agree on contextual data that is needed

in events more than dedicated enrichers; and at the same time, it can achieve high precision integration of event context with a high completeness of events comparable to that of event processing based on dedicated enrichers.

- *H4*. Approximate event processing can operate in event environments with low-cost agreements on event semantics and pragmatics more than exact event processing; and at the same time achieve timely event matching with high true positives and negatives, and high precision integration of event context with high completeness of events, comparable to that of event processing based on exact models.

To test the hypotheses, this chapter constructs a model which realizes the element of dynamic native event enrichment, along with the element of approximation as outlined in Section 7.2. This model has been mainly presented in the ACM International Conference on Distributed Event-Based Systems (DEBS 2013) [154], and the International Workshop on Semantic Sensor Networks (SSN 2011) at the International Semantic Web Conference (ISWC 2011)[156].

The rationale for using these elements and formulating the hypotheses is discussed in Section 7.3 and Section 7.4. The main elements of the enrichment model are discussed in Section 7.5. The model and its formalism are discussed in Section 7.6 and Section 7.7. Section 7.8 discusses a Linked Data instantiation of the model based on semantic relatedness and spreading activation.

The constructed model is then empirically validated as detailed in Section 7.9. This chapter shows that the proposed hypotheses *H3* and *H4* are valid. Thus, the elements of dynamic native enrichment and approximation can answer the research question and consequently can address the requirements of effective and efficient event processing that is loosely coupled in pragmatics. The chapter is summarized in Section 7.10.

## 7.2 Overview of the Dynamic Native Event Enrichment Model

This model tackles the requirements of efficient and effective loose pragmatic coupling. In this model, events are assumed incomplete under an open world assumption. Enrichment is the process of complementing events from background knowledge. The model uses

four aspects for event enrichment: determination of the enrichment source, retrieval of information items from the enrichment source, finding complementary information for an event in the enrichment source, and fusion of complementary information with the event.

The model proposes that the enrichment logic is described using a set of declarative language constructs similar to the ones used currently for matching purposes. Four language clauses that are mapped to the four enrichment aspects are proposed: ENRICH FROM, RETRIEVE BY, FIND BY, and FUSE BY. All the enrichment clauses are described by the event consumer. The resulting subscription, which contains enrichment and matching elements, is called a unified subscription. For instance, the following unified subscription tells the engine to explore a Linked Data graph by a method called Spreading Activation to enrich an RDF event with triples that can be missing such as the *floor* in the building where it was generated.

```
ENRICH FROM <www.myenterprise.org>
RETRIEVE BY 'DEREF'
FIND BY 'Spreading Activation'
FUSE BY 'UNION'
{?event rdf:type ont:EnergyConsumption.
 ?event (?p){3} building:SecondFloor.}
```

In the following two sections, Section 7.3 and Section 7.4, the rationale for using dynamic native enrichment and approximation, along with the hypotheses are detailed. The concrete model is then discussed afterwards from Section 7.5 to Section 7.9.

### 7.3 Dynamic Native Event Enrichment

The elements of subsymbolic distributional event semantics and free event tagging target the proper semantic interpretation of events within a loosely coupled paradigm. The element of dynamic native event enrichment of the proposed model is concerned mainly with building upon the semantic interpretation through pragmatic loose coupling in event processing systems for improved contextual interpretation.

Pragmatics, as discussed in Chapter 2 and Chapter 3, aligns with the meaning of pragmatics used in semiotics: “interpretation of the sign in terms of relevance, agreement, etc.” [143, p. 196]. In computing systems, this meaning covers the use of the context of data upon processing. This notion has been recognized within event processing systems as put by Antollini et al.:

“These events encapsulate data which can only be properly interpreted when sufficient context information about its intended meaning is known. In general, this information is left implicit and as a consequence it is lost when data/events are exchanged across system or institutional boundaries.” [230]

The dynamic native event enrichment element aims at providing the context of events in a loosely coupled manner.

### 7.3.1 Information Incompleteness

While the basic information item in an event-based system is an event, it is not uncommon that users require the system to handle contextual information that is not encoded in the event. Such information typically comes from legacy databases or web data sources. This causes an information incompleteness problem for events to be sufficient for tasks such as subscription matching.

One current solution to the information incompleteness issue is to develop external, static and dedicated event processing agents that retrieve information from legacy data sources and enrich the event before it is propagated for further processing. For example, an *energy consumption event* is generated by a smart electric heater containing the heater’s serial number. An enricher retrieves information about the room and floor of the heater from a building management system database and adds it to the event which can then be considered when matching the users’ interests in high energy consumption events from that specific room or floor.

Large-scale applications of event-based systems such as the Internet of Things would have an increasing number of tasks that require information not included in events. In these environments, the enrichment agents can quickly become difficult to develop and maintain. I argue that the problem lie in the approach taken in current event-based

middleware where an event is assumed as a *closed world*. For example, if a subscription tests a specific property that is not included in the event, then that is considered a negative match by default. No attempt is made to complement information in the event before judging of positive or negative matching.

The need to complement incomplete events has been recognized by the event processing community. Hinze et al. states that:

“event enrichment calls for an understanding not only of the events but also for the external sources of information.” [9]

Hohpe and Woolf [18] dedicate a set of patterns such as message translator, content enricher, and aggregator to address several problems that can be classified under event incompleteness. Teymourian et al. [36] investigate the improvement of expressiveness and flexibility of complex event processing systems via the usage of background knowledge about events and their relations to other concepts in the application domain.

Related work from the database community identifies the problem of incomplete databases and incomplete queries. While the proposed approaches are more attached to databases in general and the relational model, in particular, they give good insight into the problem. Some work focuses on missing tuples and missing values such as [231]. Some are more aligned with the query answering perspective such as [232] and [233]. While other works focus on improving the quality of incomplete databases [234].

### 7.3.2 Dimensions of Incompleteness

Event incompleteness is a relative concept; it does not only depend on the event but also on the event consumption logic that is implemented by an event consumer. Event consumers may vary from simple User Interface (UI) agents to complex event processing engines. To simplify the discussion on event consumers, I limit the discussion to content-based matchers of single events using a subscription language to match events. These are common in the publish/subscribe paradigm and are usually implemented using a message-oriented middleware [235]. However, generalization to other types of event consumers is possible in light of the formalism presented in Section 7.7.

Let us consider the following example subscription:

**Example 7.1** (Subscription for High Energy Consumption in the Second Floor).

$$\{ \mathbf{type} = \text{energy consumption},$$

$$\mathbf{floor} = \text{second floor},$$

$$\mathbf{consumption} = \text{high} \}$$

Given a particular event consumption logic, event incompleteness has a broad set of orthogonal dimensions. I define the dimensions based on an analysis of the patterns of Hohpe and Woolf [18]. This analysis produces general dimensions of incompleteness as follows:

1. *Event Format* where the event lacks the syntactical structure that can be processed by an event consumer. For example, let an event be as follows:

$$\{ \text{energy consumption of the heater in the second floor is high} \}$$

This event is in a plain text language syntax and thus cannot be processed by an event consumer which uses the subscription from Example 7.1. This is because the subscription expects attribute-value syntax not available in the event.

2. *Event Semantics* where the event lacks references to an interpretation scheme that can be used by an event consumer to understand what the event payload means. For example, let an event be as follows:

$$\{ \text{energy consumption, second floor, high} \}$$

This event is in tuple structure. It lacks the reference scheme according to which an event consumer which uses the subscription from Example 7.1 can interpret the actual indication of the term ‘*high*’.

3. *Complementary Background Knowledge* where the event lacks the amount of information required by an event consumer and the complementary information resides in an enrichment source. For example, let an event be as follows:

$$\{ \mathbf{type}: \text{energy consumption},$$

$$\mathbf{device}: \text{heater1},$$

$$\mathbf{consumption}: \text{high} \}$$

This event cannot be processed by an event consumer that uses the subscription from Example 7.1 because the event lacks any information about the ‘*floor*’ in which the event occurred. This complementary information is likely to exist in a building management system database, which has a fact such as (*heater1, exists in, second floor*).

4. *Complementary Transformation* where the event lacks the amount of information required by the event consumer and the complementary information can be obtained via a reasoning process over the event. For example, let an event be as follows:

```
{type: energy consumption,
  device: heater1,
  watt-hour: 1500}
```

Let the event consumer use the following subscription:

```
{type= energy consumption,
  device= heater1,
  kilowatt-hour= 1.5}
```

The event lacks the property ‘*kilowatt-hour*’ and thus is incomplete for the consumer. However, this information can be obtained by calculation on the actual event itself using a transformation rule such as: *kilowatt-hour*= *watt-hour*/1000.

5. *Temporal Segmentation* where a single event does not have the amount of information required by an event consumer and the complementary information resides in other events that occurred previously or are going to occur in the future. For example, it is common to have three-phase electricity power feeds to buildings. Clamp-on power monitoring sensors are usually installed on every 1-phase cable entering the building. This results in three events arriving at a specified rate one after the other:

```
{type:power consumption, consumer:building, watt phase 1, 3000}
```

```
{type:power consumption, consumer:building, watt phase 2, 2800}
```

```
{type:power consumption, consumer:building, watt phase 3, 3200}
```

Let an event consumer use a subscription such as the following:

```
{type= power consumption,  
consumer= building,  
watt all phases: 9000}
```

The consumer finds that all the events lack the knowledge about the three-phases power consumption. However, such information can be obtained by temporally aggregating three events from all the phases in order to get the overall power consumption that can be processed by the consumer.

The event format is a part of the event syntactic level transformation, which is out of the scope of this work. Event semantics are handled within the context of the previous two elements in Sections 5.4 and 6.3. As this work is scoped on single event processing, I leave the temporal segmentation dimension to future work. The reasoning for complementary transformation over events is assumed to be done beforehand with the result stored in a knowledge base. That turns the complementary transformation dimension into the complementary background knowledge dimension, which is the definition of contextual enrichment as used in this work.

### 7.3.3 Unified and Native Event Enrichment

I define event enrichment as the “process to complement events.” Patterns by Hohpe and Woolf [18] reflect the current state-of-the-art and practice in the design of event processing networks where dedicated agents are assigned with well-defined tasks to overcome some incompleteness issues. For example, they propose the use of dedicated event enrichment agents to access a database and retrieve necessary information that is added to events before they propagate to consumers. However, such agents are ad-hoc and tailored to the particular situations they are designed for. That challenges the event processing vision detailed by Etzion and Niblett [10] which calls for a *unified and declarative* way to process events.

Enrichment agents are non-native to the paradigm, and as event processing systems scale out to open, distributed, and heterogeneous environments, the maintenance of such enrichment agents becomes difficult. Other related work, as discussed in Chapter 3,

focuses on the fusion of background knowledge with events using a query answering paradigm that spans events and background knowledge. However, such approaches make some assumptions that may not hold in many situations. For example, the work of Teymourian et al. [36] assumes that the background knowledge and events have the same data format and semantics and that the knowledge base is accessible via a query service making the federation of the query feasible.

To make advancement on the event incompleteness problem, it is crucial to deal with the abstract characteristics of the problem and to integrate it into the event processing paradigm, so it becomes a native component of event processing engines. By *unified* I mean a model of event enrichment that takes place in coordination with event matching. The reason is that event matching and enrichment can be seen as both important tasks to provide a better interpretation of events at the time of processing, at two levels: semantics and pragmatics. By *native* I mean that event enrichment takes place as a component within event processing engines, rather than having dedicated engines or external agents only for enrichment.

The inherent feature of decoupling has its own virtues, but it introduces other challenges in the event-based paradigm. An important one is the fact that event producers should have minimal assumptions on the information needs of event consumers. As a result, the content of an event payload becomes independent of the consumers' needs. This independence can lead to information incompleteness on the consumers' side. If an event consumer ignores the concerns of information incompleteness and attempts to conduct matching between its subscription and events, this may result in a high rate of false positives or false negatives due to the lack of relevant information in the events needed for the correct matching result.

Unified and native enrichment can operate within a loosely coupled paradigm and thus addresses the Requirement *R2* of loose pragmatic coupling. Besides, aligning enrichment with matching can lead to an efficient and effective interpretation of events, meeting requirements *R3* and *R4* of efficiency and effectiveness.

### 7.3.4 Late Dynamic Event Enrichment

Event enrichment can be done closer to the producer's side or closer to the consumer's side. Dedicated enrichers cannot be easily classified as early or late, as that goes back to the architecture, which specifies if they function closer to the event producer or the event consumer. Nonetheless, I argue that consumers can better judge the content completeness of events concerning their information needs. Thus, I explore in this work enrichment that is unified with consumption logic, i.e. matching.

While a unified native enricher can address information incompleteness and adds contextual information, it partially addresses the loose pragmatic coupling requirement. Thus, enrichment should be dynamic. By *dynamic* I mean that the consumer should roughly define the basic elements needed for enrichment to happen, but most of the process is done by the engine at the time of matching to decide how to complement the event and with which data. Such an approach frees the users from having to agree on contextual information of the events and shifts most of that burden to the event engine itself, meeting the Requirement *R2* of loose pragmatic coupling.

### 7.3.5 Limitations of Dynamic Native Event Enrichment

The two main limitations of a dynamic native event enrichment approach are:

1. *Low effectiveness in controlled environments* which means that such an approach is not superior to dedicated ad-hoc enrichers in controlled environments from a precision point of view. When event producers and consumers can pay the cost of agreements on contextual information to include in events, it can still be better than delegating this task to the dynamic enricher, which might miss some context during enrichment. This can be the case in critical applications such as security systems.
2. *Low efficiency* as such an approach requires the enricher to look for complementary information, retrieve it, and fuse it within events. Dedicated ad-hoc enrichers follow a *join* operator paradigm typically in a query language which is more efficient than figuring out complementary information at a later stage.

TABLE 7.1: Dynamic Native Event Enrichment and Requirements

	<b>Dedicated Ad-hoc Enrichers</b>	<b>Dynamic Native Event Enrichment</b>
<i>R1.</i> Loose semantic coupling	NA	NA
<i>R2.</i> Loose pragmatic coupling	-	+++
<i>R3.</i> Efficiency	+++	+
<i>R4.</i> Effectiveness	+++	++

	+++	the model excellently addresses the requirement
	++	the model moderately addresses the requirement
Legend	+	the model slightly addresses the requirement
	-	the model mildly affects the requirement in a negative way
	--	the model moderately affects the requirement in a negative way
	NA	the requirement is not in question

I argue though that when the loose pragmatic coupling is a requirement, which is the case in open distributed systems such as the Internet of Things, these limitations are reduced as the building of dedicated ad-hoc enrichers can be costly or infeasible.

### 7.3.6 How Dynamic Native Event Enrichment Meets The Requirements

As discussed throughout Section 7.3 dynamic native event enrichment loosens pragmatic coupling, which is required in distributed and decoupled event processing systems. Dynamic native enrichers and dedicated ad-hoc enrichers are the main models for crossing pragmatic boundaries, but the dynamic native event enrichment approach can meet the main requirements as summarized in Table 7.1.

## 7.4 Approximate Computing Versus Incomplete Information

Section 5.5 showed the role the approximation element within computing systems. It mainly tackled the use of approximation for time optimization and integration. Approximation has also been used, but much less, in areas related to information quality in databases. For instance, Parsian et al. [234] provides a model to assess information completeness within the relational data model. Based on the Closed World Assumption (CWA) and the Open World Assumption (OWA), and on the existence of *NULL* values,

one can assess with a value how complete the database is in relation to a query [236]. Queries can still return tuples that are assumed to meet or not to meet the query. CWA and OWA assumptions and the use of *NULL* values in the database include approximation of what the query result should in fact be.

Extensions of the relational model to make such process explicit have been done, early by Codd [237]. He extended the two-valued logic into a three-valued logic, to include the *NULL* value into account along with *true* and *false*. For example, the open world assumption that the *NULL* value of an *EMPLOYEE*'s *ADDRESS* represents some incomplete information that exists somewhere out of the database. Under such an assumption, the selection of employees who live in *California* leads to a set of employees who are known to the database to live in *California*, and a set of employees who *MAYBE* a part of the answer but the incomplete information restricts the certainty on the later set.

Thus, having incomplete information has been acknowledged in the literature. Assumptions, interpretations of *NULL* values, and extensions of Boolean logic have been used to cope with this problem. Approximation is at the heart of this discussion either in assessing how much a database is complete, how much it is complete with respect to a query, or which tuples should go into a query's answer to approximate reality. Based on this work, I argue that although quantification of approximation in query processing over incomplete data has not been dominant within databases, approximation is a logical model to adopt when contextual agreements cannot be guaranteed. In loosely pragmatically coupled environments, approximation can be the right model to complement events during enrichment and to match the resulting approximately completed events.

## 7.5 Elements of Enrichment

Given the final set of incompleteness dimensions, four fundamental challenges are recognized.

### 7.5.1 Determination of the Enrichment Source

The first challenge to face event enrichment is the decision on which enrichment source(s) to use. The challenge comes from the fact that event producers and consumers are

decoupled and potentially have various perspectives of where complementary information for an event may exist. Determining the enrichment source may be statically stated by the event producer or consumer making this challenge easy to overcome. However, if sources are not known beforehand, then a source discovery process is needed. Some possible enrichment sources include:

- *Wikis*: The Wikipedia online corpus ‘<http://en.wikipedia.org/wiki/>’ can be considered as a textual domain-agnostic enrichment source.
- *Relational Databases*: An example is a Building Management System (BMS) database described by a connection string ‘`Server=www.example.com rdbms;Database=BMS-DB;`’.
- *Linked Data* [238]: The DBpedia corpus, for example, can be addressed by its domain ‘<http://dbpedia.org/resource/>’.

### 7.5.2 Retrieval of Information Items from the Enrichment Source

The access and retrieval mechanism poses a challenge to the enrichment process as it affects its ability to retrieve atomic information items from the enrichment source. Retrieval of information items can be challenging if network transfer has reliability issues or if the retrieval speed forms a bottleneck in the system. The exact retrieval mechanism will depend on the selected enrichment source. Some example retrieval mechanisms include:

- *Wikis*: A retrieval mechanism for Wikipedia is a search operation against its search API followed by an HTTP GET request to access a Wikipedia article as the information item.
- *Relational Databases*: A retrieval mechanism for a relational database is an SQL query against a query interface, with the retrieved rows as the information items.
- *Linked Data*: A retrieval mechanism for the DBpedia corpus, for instance, is looking up, i.e. *dereferencing*, URIs [239] of the resources, with the RDF [240] graphs of these URIs being the information items retrieved.

### 7.5.3 Finding Complementary Information in the Enrichment Source

The ability of the enrichment process to retrieve atomic information items from an enrichment source is faced with the challenge to determine which of the information items can complement an event and should be retrieved. Several ways to find complementary information are:

- *Wikis*: To find complementary information in the Wikipedia corpus, articles related to a term in the event can be searched and then links from these articles are followed one step deep and ultimately all the resulting articles are retrieved.
- *Relational Databases*: To find complementary information in a relational database, one can formulate a SQL query with some specific primary keys coming from the event.
- *Linked Data*: To find complementary information in the DBpedia corpus, a spreading activation [241] of URIs can be conducted starting from seed URIs and following the links in the data cloud with some termination conditions.

### 7.5.4 Fusion of Complementary Information with the Event

The final challenge is fusing the complementary information items with the event. Multiple instances of fusion are presented in the following example:

*Example for Fusion Methods*: Let an event be the attribute-value map:

```
{(type= energy consumption),
(device= heater1),
(consumption= high)}
```

Let the enrichment source be a relational database with two relations as follows:

heater	room
heater1	room123

room	floor
room123	second floor

One possible fusion method is to add two attribute-value pairs to the event so it becomes:

```
{(type= energy consumption),
```

```
(device= heater1),  
(consumption= high),  
(room= room123),  
(floor= second floor)}
```

Another fusion method is to add one attribute value pair that contains the location to the event so it becomes:

```
{(type= energy consumption),  
(device= heater1),  
(consumption= high),  
(room= room123),  
(location= room123, second floor)}
```

## 7.6 Event and Enrichment Flow Model

The key pillar of the proposed model is the recognition of enrichment as a core task of event processing engines. Also, the enrichment behaviour of an event processing engine can be dictated to the engine using a uniform and declarative mechanism. The cornerstone of the model is the concept of an enrichment element: a declarative specification for the engine to enrich events with complementary information items. The model proposes that the enrichment element is described using a set of declarative language constructs similar to the ones used currently for matching purposes. To systematically characterize the language constructs needed for the enrichment element, I propose four language clauses that are mapped to the four enrichment challenges as follows:

- *ENRICH FROM* clause allows the engine to determine the enrichment source(s) explicitly.
- *RETRIEVE BY* clause allows the engine to determine the retrieval mechanism for atomic information items.
- *FIND BY* clause specifies the approach which would dictate the retrieval of a subset of information items from the enrichment source(s) that can complement the event.

- *FUSE BY* clause defines the fusion approach to integrating retrieved complementary information with the incomplete event.

The next issue is to determine who is responsible for defining the enrichment elements. Reviewing the clauses of an enrichment element shows that some of these can be specified by the event producer and/or the event consumer. Specifically, the enrichment source and retrieval mechanism can be defined by the producer who may know them at the time of producing the event.

The model proposes that all the enrichment clauses are described by the event consumer. That is because the consumer has a better understanding of the information need on the consumption side. This is also aligned with scenarios where the event producer has little assumptions on information needs of the consumers and where decoupling is the norm. This adds to the aspect of loose semantic coupling and approximate matching in event processing systems. Consequently, the model suggests that the enrichment element co-exists with the matching element, which forms subscriptions in current practices. The resulting subscription, which contains enrichment and matching elements, is called a *unified subscription*.

By having unified subscriptions, enrichment can be brought to the core of the engine. It operates based on the enrichment element and uses the matching element to conduct an enrichment process over the incoming incomplete events and enrichment source(s) to produce enriched events that can then be matched against the matching element. It is called a *native enricher* in this model. While implementation details of the enricher are left to the particular instantiations, the proposed model suggests that the enricher not only uses the enrichment clauses to operate but also the matching element to guide the enrichment process. Figure 7.1 depicts the proposed enrichment model.

The following example presents a simple instantiation of the enrichment clauses and the native enricher.

*Example of Instantiation for Plain Text Events:* Events are represented as bags of words:

Let an event be as follows:  $\{energy, consumption, heater1, high\}$ . Let the matching element of subscriptions be a bag of words as follows:  $\{energy, second, floor, high\}$ . The semantics of event matching is that all words in the matching element need to be found in the event. Otherwise, it is a negative match.

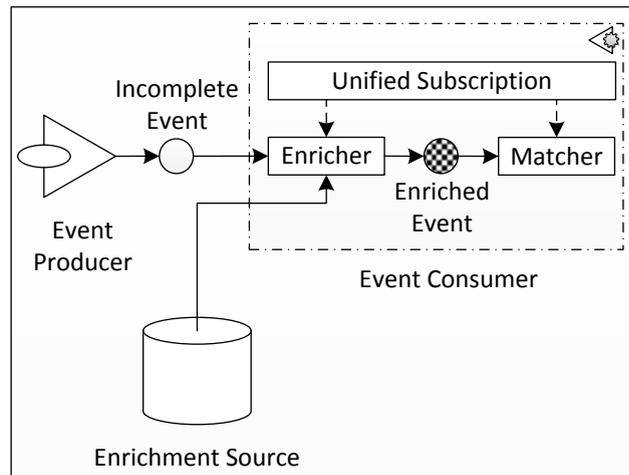


FIGURE 7.1: Unified and native enrichment model

The enrichment source for the system is assumed to be an enterprise wiki of text articles called enterprise-wiki. The wiki contains an article titled *second floor* that contains the term *heater1*. The wiki can be searched via a term search API which returns a list of articles containing the term. The API is accessible via a RESTful web service. When the API is searched with the term *heater1*, the article titled *second floor* is returned.

The enrichment clauses are defined as the following:

- *ENRICH FROM* specifies the name of the wiki.
- *RETRIEVE BY* specifies the access protocol.
- *FIND BY* specifies the search mechanism.
- *FUSE BY* defines whether to extract words from the retrieved article's title or content, and if to add/replace the event words to/with the new found words.

A full example unified subscription becomes:

```

ENRICH FROM 'enterprise-wiki'.
RETRIEVE BY 'HTTP GET'.
FIND BY 'term search'.
FUSE BY 'title terms' 'add'.
{'energy', 'second', 'floor', 'high'}.
  
```

When the event  $\{energy, consumption, heater1, high\}$  arrives at the system, the native enricher uses the specified values in the event to search the enterprise-wiki using each

word at a time. Assuming that the enricher firstly retrieves the article titled *second floor*, it extracts the single words from the article's title and adds them to the event. The enriched event becomes as follows:  $\{energy, consumption, heater1, high, second, floor\}$ . Other articles are retrieved and fused in a similar manner. The matching element is then evaluated against the enriched event. As a result, the matcher finds a positive match.

## 7.7 Formal Model

The model is represented using the quadruple  $(\mathcal{L}, E, ES, U)$ , where:

- $\mathcal{L}$  is the unified subscription language.
- $E$  is the set of events.
- $ES$  is a set of information items that form the source of enrichment.
- $U$  is the universe that contains all the possible information items.

The model has two underlying assumptions concerning *valid information items* and *common information items*. Valid information items are those which are considered to be true facts. Given an event  $e \in E$ , let's assume that the only valid information items are those which exist in the event  $e$  or the enrichment source  $ES$ . In other words, this assumption is equivalent to a Closed World Assumption (CWA) where the world  $W = e \cup ES$ . In fact, it is worth mentioning that traditional event processing systems usually make a closed world assumption at the matching stage, where the world  $W = e$ . The principal assumption that the world is limited to the event causes the incorrect decisions of the matcher in judging many positive and negative matches.

The other assumption concerns common information items between events and the enrichment source. Let's assume that there is no intersection between the content of  $e$  and  $ES$ , i.e.  $e \cap ES = \emptyset$ . The purpose of this assumption is to simplify the description of the model. However, in reality, the event may have been published with some information items that also exist in the enrichment source. Nevertheless, the model is easily extended to the case where  $e \cap ES \neq \emptyset$ . When conducting enrichment in practice, the information items in  $ES$ , which are already in  $e$ , can simply be discarded to turn the

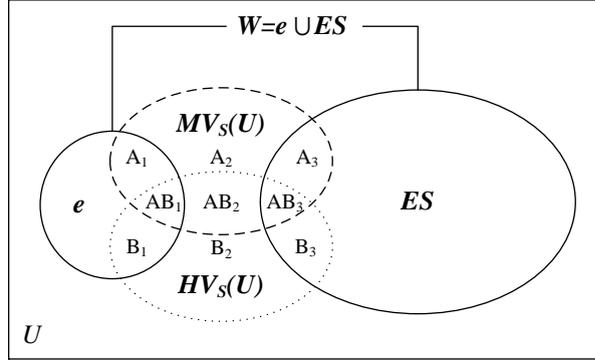


FIGURE 7.2: The universe  $U$ , the event  $e$ , the enrichment source  $ES$ , the world  $W$ , the enrichment view  $HV_S$ , and a matching view  $MV_S$ .

assumption into a valid assumption. Figure 7.2 illustrates the various concepts of the model.

Let  $S$  be a subscription in  $\mathcal{L}$ ,  $S$  is a pair  $(HS, MS)$ , where:

- $HS$  is the enrichment clauses element of  $S$ .
- $MS$  is the matching predicates element of  $S$ .

The model is described through the following definitions.

**Definition 7.1** (Boolean Matching Element). Let  $S$  be a unified subscription and  $I$  a set of information items,  $M_S$  is a Boolean matching element if

$$M_S(I) \in \{True, False\} \quad (7.1)$$

**Definition 7.2** (Approximate Matching Element). Let  $S$  be a unified subscription and  $I$  a set of information items,  $M_S$  is an approximate matching element if

$$M_S(I) \in \mathbb{R} \quad (7.2)$$

**Definition 7.3** (Unknown Matching Result). Let  $S$  be a unified subscription and  $I$  a set of information items,  $M_S = Unknown$  if

$$\begin{aligned} & (M_S \text{ is a Boolean matching element} \wedge M_S(I) \notin \{True, False\}) \\ & \vee (M_S \text{ is an approximate matching element} \wedge M_S(I) \notin \mathbb{R}) \end{aligned} \quad (7.3)$$

**Definition 7.4** (Matching View). Let  $S$  be a unified subscription and  $I$  a set of information items,  $MV_S$  is a matching view of  $S$  on  $I$  if

$$M_S(MV_S(I)) \neq \text{Unknown} \quad (7.4)$$

**Definition 7.5** (Enrichment View). Let  $S$  be a unified subscription and  $I$  a set of information items,  $HV_S$  is an enrichment view of  $S$  on  $I$  if

$$HV_S(I) = \{ii : ii \in I \wedge ii \text{ is retrieved during enrichment}\} \quad (7.5)$$

**Definition 7.6** (Complete Event). Let  $S$  be a unified subscription and  $e$  an event from  $E$ ,  $MV_S$  is complete with respect to  $M_S$  if

$$\exists MV_S \text{ such that } M_S(MV_S(e)) = M_S(MV_S(W)) \quad (7.6)$$

**Definition 7.7** (Enriched Event). Let  $S$  be a unified subscription, and  $e$  an event from  $E$ . Let  $ES$  be the enrichment source,  $HV_S$  the enrichment view of the  $H_S$  element of  $S$ ,  $\oplus$  the FUSE BY operator of  $H_S$ ,  $ee$  is the enriched event of event  $e$  according to  $H_S$  if

$$ee = e \oplus HV_S(U) \quad (7.7)$$

**Definition 7.8** (Valid Enrichment). Let  $S$  be a unified subscription, and  $e$  an event from  $E$ ,  $ES$  the enrichment source,  $HV_S$  the enrichment view of the  $H_S$  element of the unified subscription  $S$ ,  $HV_S(U)$  is valid if

$$HV_S(U) \setminus HV_S(W) = \emptyset \quad (7.8)$$

**Definition 7.9** (Successful Enrichment). Let  $S$  be a unified subscription, and  $e$  an event from  $E$ ,  $ES$  the enrichment source,  $HV_S$  the enrichment view of the  $H_S$  element of the unified subscription  $S$ ,  $\oplus$  the FUSE BY operator of  $H_S$ ,  $HV_S(U)$  is successful if

$$HV_S(U) \text{ is valid } \wedge e \oplus HV_S(U) \text{ is complete with respect to } M_S \quad (7.9)$$

**Definition 7.10** (Minimal Successfully Enriched Event). Let  $e$  be an event from  $E$  and  $ES$  be the enrichment source. Let  $S_1, S_2, \dots, S_n$  be a set of unified subscriptions in

$\mathcal{L}$  where the matching element of all of them is the same  $M_S$ , while they vary in the enrichment elements being  $H_{S_1}, H_{S_2}, \dots, H_{S_n}$  respectively. Let  $HV_{S_1}, HV_{S_2}, \dots, HV_{S_n}$  be the set of enrichment views corresponding to the subscriptions. Let  $ee_1, ee_2, \dots, ee_n$  be the enriched events of  $e$  according to the enrichment views respectively,  $ee_k$  is a minimal successfully enriched event if

$$\forall ii \in ee_k \Rightarrow ee_k \setminus \{ii\} \text{ is not complete with respect to } M_S \quad (7.10)$$

An ideal event enrichment process would always turn events into minimal successfully enriched events. Ideally, the areas in Figure 7.2 of  $MV_S(W)$  and  $HV_S(W)$  would be identical for at least one  $MV_S$ . Besides, the enrichment view would be valid, i.e.  $HV_S(W) = HV_S(U)$ . Thus, the areas  $A_1, B_1, A_2, B_2, AB_2, A_3,$  and  $B_3$  become all empty.

The definition above can be interpreted as a hard constraint, meaning that an enrichment process is considered successful for an event only if it produces a minimal successfully enriched event. This interpretation is suitable in many cases such as when the matching element  $M_S$  is a Boolean matching element. However, there are cases where the event processing system may accept approximation. One example is when the matching element  $M_S$  is an approximate matching element. In such cases, it is suitable to adapt Definition 7.10 to a softer interpretation, leading to Definitions 7.11 and 7.12.

**Definition 7.11** (Cost of Transformation into a Minimal Successfully Enriched Event).

Let  $e$  be an event from  $E$  and  $ES$  the enrichment source, let  $S_1, S_2, \dots, S_n$  be a set of all possible subscriptions in  $\mathcal{L}$  where the matching element of all of them is the same  $M_S$ , while they vary in the enrichment elements being  $H_{S_1}, H_{S_2}, \dots, H_{S_n}$  respectively. Let  $ee_{m1}, ee_{m2}, \dots, ee_{mk}$  be the set of minimal successfully enriched events of  $e$  according to the various enrichment clauses elements  $H_{S_1}, H_{S_2}, \dots, H_{S_n}$ . Let  $S$  be a subscription with the enrichment element  $H_S$ . Let  $ee$  be the enriched event of  $e$  according to  $H_S$ . The cost function  $MSECost$  is defined as follows:

$$MSECost : W \times W \rightarrow \mathbb{R}^+ \cup \{0\} \quad (7.11)$$

$$MSECost(ee, ee_{mi}) \text{ is the minimum cost to turn } ee \text{ into } ee_{mi} \quad (7.12)$$

$$MSECost(ee_{mi}, ee_{mi}) = 0 \quad (7.13)$$

**Definition 7.12** (Approximately Minimal Successfully Enriched Event). Let  $ee$  be a successfully enriched event and  $ee_{mi}$  any minimal successfully enriched event,  $ee$  is an approximately minimal successfully enriched event if

$$\text{Min}_{ee_{mi}}(\text{MSECost}(ee, ee_{mi})) > 0 \quad (7.14)$$

## 7.8 A Linked Data Instantiation

This section details the implementation of the proposed model illustrated in Figure 7.1 and Figure 7.2 through the instantiation of the following elements: the event model, the enrichment source model, the matching element of subscriptions, and the enrichment element along with a native enricher. The instantiation is designed for Linked Data events. Linked Data [239] along with its core RDF graph model can be seen as a generic model for events, making the concepts applied in this instantiation also applicable in other implementations. A large amount of openly accessible Linked Data has been published on the Web in the recent years making it easier to experiment with Linked Data events to study the enrichment model. Linked Data has also been used as a mechanism to link contextual data within different domains including finance, life sciences, public sector and energy [242].

### 7.8.1 Event Model

Events are instantiated as Linked Data events. Thus, an overview of Linked Data is given before proceeding.

#### 7.8.1.1 Linked Data

Emerging from research into the Semantic Web [59, 243, 244], Linked Data proposes an approach for information interoperability based on the creation of a global information space. Linked Data leverages the existing open protocols and standards for the World Wide Web (WWW) architecture for sharing structured data on the web. The overall objective of Linked Data is to provide flexible data publishing and consumption.

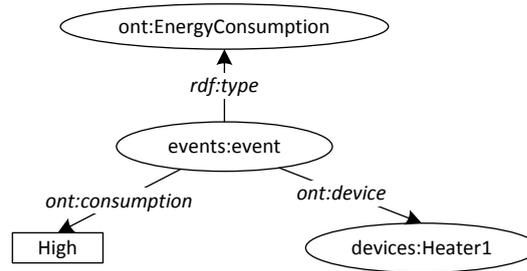


FIGURE 7.3: An example Linked Data event

Berners-Lee [239] summarizes Linked Data in four principles:

1. Using URIs as names for things.
2. Using HTTP URIs so that people can look up those names.
3. When someone looks up a URI, providing useful information using standards such as RDF [240].
4. Including links to other URIs so that people can discover more things.

#### 7.8.1.2 Event Model

An event is instantiated as a labelled directed graph. The Resource Description Framework (RDF) is used to represent event information using statements or triples. A statement consist of a (subject, property, object) triple.

Subjects are references to information resources and are represented as URIs. Objects may be URIs or literal values. Properties come from various vocabularies, the Linked Data name of ontologies, and are represented as URIs of terms in these vocabularies. One subject may have multiple statements with the same property and different objects.

The resulting event can be represented as follows: Let  $E$  be the set of events conforming to the event model,  $P$  the set of properties. Let  $URIs$  be the set of all URIs, and  $Lit$  the set of all Literals such as strings and numbers, then an event can be seen as a finite set of triples as follows:

$$e \in E \Leftrightarrow e = \{(s, p, v) : (s, p, v) \in URIs \times P \times (URIs \cup Lit)\} \quad (7.15)$$

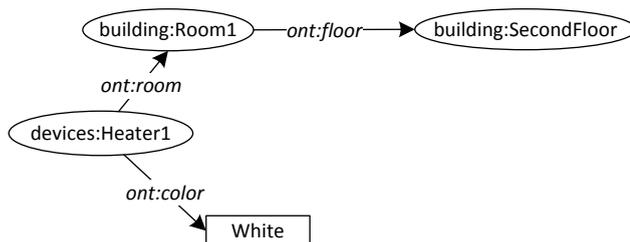


FIGURE 7.4: An example Linked Data enrichment source

A URI can be written using prefixes for clarity. `http://www.example.com#event` can be written as `example:event` with the prefix representing `http://www.example.com`. Figure 7.3 illustrates an example event where `ont` represents a prefix for the vocabulary of terms in the energy domain, `devices` a prefix for instances of devices in the environment, and `events` a prefix for all event instances.

## 7.8.2 Enrichment Source Model

The enrichment source is instantiated as a labelled directed graph. RDF is used to represent enrichment information. The enrichment source is a set of triples (subject, property, object) following the Linked Data principles. Let  $ES$  be the enrichment source,  $P$  the set of properties,  $URIs$  the set of all URIs and  $Lit$  the set of all Literals such as strings and numbers then:

$$ES = \{(s, p, v) : (s, p, v) \in URIs \times P \times (URIs \cup Lit)\} \quad (7.16)$$

Figure 7.4 illustrates an example enrichment source where `building` is a prefix for instances such as `rooms` and `floors`. The enrichment source is assumed to be accessible by dereferencing URIs associated with it. Dereferencing a URI means sending an HTTP request to its host, specifying the content type to be returned such as RDF, and finally receiving the HTTP response. The validity of a triple as required by Definition 7.8 is judged by its existence in the event or the enrichment source.

## 7.8.3 Matching Element Model

The instantiation of the matching element of a subscription is a simplified version of the SPARQL Protocol And RDF Query Language (SPARQL) patterns [245] which can

contain basic graph patterns with variables. The matching element uses property paths in the place of properties to describe a regular expression of properties, or a path. The matching element is Boolean as defined in Definition 7.1. A matching view as defined in Definition 7.8 is the set of all triples that forms a solution to the graph pattern. Example 7.8.1 presents an example matching element.

**Example 7.8.1** (A Matching Element). The following matching element matches any event of type energy consumption whose URI has a path to the second floor URI within three nodes:

```
?event rdf:type ont:EnergyConsumption.
?event (?p){3} building:SecondFloor.
```

#### 7.8.4 Enrichment Element Model

The instantiation of the enrichment element of a subscription is as follows:

- ENRICH FROM specifies the domain URI of the enrichment source.
- RETRIEVE BY specifies dereferencibility as the method for retrieval, notated as DEREf.
- FIND BY specifies how to explore the enrichment source to find complementary information. I propose a spreading activation strategy to be used by the enricher as explained in Section 7.8.5. The enrichment view defined in Definition 7.5 is the set of all triples whose subjects are activated during the spreading activation.
- FUSE BY realizes the  $\oplus$  operator of the model presented in Definition 7.7. The RDF UNION is a suitable instantiation.

Example 7.8.2 presents a unified subscription that enriches from an Enterprise Linked Data cloud, retrieves by dereferencibility, finds via a spreading activation strategy called *UniformWeightsAllAdjacent* and fuses via union. It aims at matching any event of type energy consumption whose URI has a three-links path to the *second floor*.

**Example 7.8.2** (A Unified Subscription).

```
ENRICH FROM <www.myenterprise.org>
RETRIEVE BY 'DEREF'
FIND BY 'Spreading Activation' 'UniformWeightsAllAdjacent'
FUUSE BY 'UNION'
{?event rdf:type ont:EnergyConsumption.
?event (?p){3} building:SecondFloor.}
```

The minimality of enriched events as defined in Definition 7.8 is realized by removal of triples from an enriched event. Finally, the approximation between an enriched event and a minimal successfully enriched event defined by the function  $MSECost$  in Relations 7.11, 7.12, and 7.13 is realized by the cardinality of the relative complement operation ' $\setminus$ ' on sets of triples. Thus, the cost to turn an enriched event  $ee$  into a minimal successfully enriched event  $ee_m$  is composed of two costs:

- The cost to include all the successful enrichment triples in  $ee_m$  into  $ee$ . That is equivalent to  $|ee_m \setminus ee|$ .
- The cost to remove all unnecessary enrichment triples from  $ee$ . That is equivalent to  $|ee \setminus ee_m|$ .

The first point measures the *completeness* while the second measures the *precision*. These two measures and their combination form the basis for evaluation as shown in Section 7.9.

### 7.8.5 Native Enricher

The enrichment model is realized through a spreading activation algorithm [241]. Spreading activation (SA) originated in cognitive psychology as a network processing model for a supposed model of human memory. Applications of SA can be found in Artificial Intelligence, Cognitive Science, Databases, and Information Retrieval. The pure spreading activation model incorporates a processing technique for a generic graph data structure such as the RDF graphs. Spreading activation has been employed within semantic web and Linked Data processing as shown, for example, by Rocha et al. [246], Jiang and Tan [247], and Freitas et al. [174, 200].

Spreading activation is based on the idea of marking some nodes as active and then spreading the activation into other nodes iteratively. The way in which spreading takes place, and the semantics of the active nodes depend on the application. The processing is defined by a sequence of iterations that continue until a termination condition is activated. Each iteration consists of one or more pulses and a termination check [248].

Each pulse of the spreading activation consists of three stages: pre-adjustment, spreading and post-adjustment [248]. The spreading phase consists of a number of activation waves where each node calculates activation inputs transferred to it from its neighbours, which can be done using the formula:

$$I_j = \sum_i O_i w_{ij} \quad (7.17)$$

Where  $I_j$  is the total input to node  $j$ ,  $O_i$  is the output of neighbour  $i$  and  $w_{ij}$  is a weight associated with the edge from node  $i$  to node  $j$ . When a node computes its total input  $I_j$  it calculates its output  $O_j$  as a function of  $I_j$ :

$$O_j = f(I_j) \quad (7.18)$$

The function can be simply a threshold function which decides if the node  $j$  is activated or not. The output of the node is in turn sent to neighbouring nodes in the next pulse and so on. Activation spreads from the initially activated nodes to further nodes in the network. Pure SA may fall in a deadlock and run forever unless controlled. Constraints can be enforced in the pre-adjustment stage. Four sorts of constraints can be recognized [248]:

- *Distance Constraint*: The SA should decay as it reaches nodes far from the initially activated nodes.
- *Fan-out Constraint*: The SA should cease at nodes with very high connectivity.
- *Path Constraint*: The SA should be selective in the path it spreads in making use for example of the semantics of labels on the edges.
- *Activation Constraint*: Using various thresholds can affect the behaviour of the SA.

Spreading activation within the enricher along with the Linked Data instantiation of the event, and the enrichment source models, can realize the enrichment model. Spreading Activation can be used to explore the enrichment source and retrieve a set of triples to be fused in the event. To guide the SA in the enrichment source, I propose a path constraint to favour some links over others. The path constraint is based on ranking the links connected to a spread node based on their semantic relatedness with terms in the matching element and then just follows the top two or three links. The semantic relatedness used in the experiment is a WordNet-based measure called the Path measure. Further discussion on WordNet and semantic measures can be found in [249].

## 7.9 Evaluation

To demonstrate how to evaluate a particular instantiation of the proposed enrichment model, an experiment has been conducted in association with the Linked Data instantiation of the enrichment model described in Section 7.8. The experiment has been performed using real-world data, namely, events extracted from Wikipedia, and uses the DBpedia dataset as an enrichment source. A set of event subscriptions is generated where each subscription conforms to the unified language instantiation in Section 7.8. Matching elements use the property path variables to express a path of predicates between an event and a value. The minimal successfully enriched events for each subscription are calculated to form a baseline to measure the effectiveness of enrichment.

The purpose of the experiment is to compare three strategies of event enrichment, which vary the mechanism used by the enricher to find complementary information items in the enrichment source. The variation is expressed by using different parameters in the spreading activation algorithm using the FIND BY clause of the subscription enrichment element. The three strategies are:

- *UniformWeightsAllAdjacent*: A spreading activation strategy where activation from one node spreads equally to all adjacent nodes.
- *UniformWeightsRandomAdjacent*: A spreading activation strategy where activation from one node spreads equally to a random set of adjacent nodes.

- *DifferentWeightsSemRel*: A spreading activation strategy where activation of a node spreads unequally to a set of adjacent nodes based on the semantic relatedness of the adjacency edges with the terms in the subscription matching element.

The key difference between the evaluated strategies is that the former two guide enrichment independently from the matching element of the subscription while that last one benefits from the fact that enrichment and matching logic exist together in the unified subscription. The last strategy guides the enrichment algorithm according to the semantic relatedness between the terms in the matching element and terms on the links in the enrichment source. Thus, if it performs better than the former two, then this supports that unified subscriptions native to the event processing engine is suitable for event enrichment.

It is worth mentioning that the objective is not to investigate the best approach for enrichment in this particular Linked Data instantiation but rather to validate the hypothesis that the element of dynamic native enrichment can address the requirement of loose pragmatic coupling. Investigating the best performing enrichment strategies for Linked Data events is indeed an important future direction.

### 7.9.1 Event Set and Enrichment Source

The event set used is a structured representation of events in the English Wikipedia <sup>1</sup>. DBpedia <sup>2</sup> is a community project to extract structured information from Wikipedia [238], and is one of the efforts under the Linked Open Data initiative that targets the publication of structured data on the web according to the Linked Data principles [239]. The data model used to represent DBpedia data is RDF.

The event set, of 24,000 events, contains all resources of type `dbpedia-owl:Event`. Each event is a triple in the form of `<eventURI, rdf:type, dbpedia-owl:Event>`. Examples event types in the event set are: Football Match, Race, Music Festival, Space Mission, 10th-century BC Conflicts, Academic Conferences, etc.

The enrichment source is the set of all triples that are stored in the online DBpedia and can be retrieved by looking up DBpedia resource URIs. Events are played sequentially

---

<sup>1</sup><http://www.wikipedia.org/>

<sup>2</sup><http://downloads.dbpedia.org/3.8/en/>. Last modified on the 1st of August 2012. Accessed on 25th of February 2013.

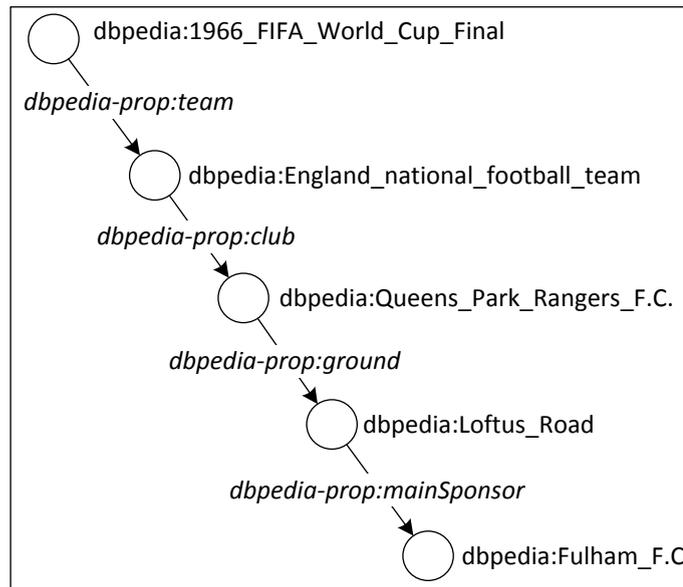


FIGURE 7.5: The base path-shaped graph used to generate the matching elements of the subscriptions

and pushed to the native enricher which searches the enrichment source for complementary information, fuses it with the events and forwards them to the event matcher.

### 7.9.2 Unified Subscriptions Set

The subscription set consists of four subscriptions. The matching element of subscriptions was automatically generated using the following method:

1. The seed Uniform Resource Identifier (URI) of the 1966 FIFA World Cup Final [http://dbpedia.org/resource/1966\\_FIFA\\_World\\_Cup\\_Final](http://dbpedia.org/resource/1966_FIFA_World_Cup_Final) is used first and resources linked to it are retrieved to build a path-shaped graph of 4-triples long. Figure 7.5 shows the resulting full path-shaped graph.
2. For the first subscription, the first triple is considered as the matching element.
3. For the second subscription, the first two triples are picked, and a matching element is constructed as defined in Section 7.8 using the two terminal URIs of the two-triple long path as subject and object and a property path variable in between.
4. The last step is repeated for subscriptions 3 and 4.

The resulting matching elements are shown in Table 7.2. Subscriptions range in complexity concerning the length of the property path in their matching elements with the

TABLE 7.2: Matching Elements of the Unified Subscriptions Set

ID	Matching Element
1	?event rdf:type dbpedia-owl:Event. ?event (?p){1} dbpedia:England_national_football_team.
2	?event rdf:type dbpedia-owl:Event. ?event (?p){2} dbpedia:Queens_Park_Rangers.F.C..
3	?event rdf:type dbpedia-owl:Event. ?event (?p){3} dbpedia:Loftus_Road.
4	?event rdf:type dbpedia-owl:Event. ?event (?p){4} dbpedia:Fulham.F.C.

most complex subscription being the one with the longest property path. To form the final unified subscriptions, each matching element is concatenated with an enrichment element that consists of the four clauses ENRICH FROM, RETREIVE BY, FIND BY and FUSE BY. The evaluated three strategies are parameters to the FIND BY operator.

### 7.9.3 Minimal Successfully Enriched Events Construction

To generate the event data that can be considered a minimal successfully enriched event for each subscription, the following methodology has been used: For each matching element of a subscription, a SPARQL [245] query is formed and executed against the DBpedia online SPARQL API. The query uses optional joins and filters to match all the events in DBpedia with all possible cases of their associated values or predicates. Example 7.9.1 shows the generated query for subscription 3.

**Example 7.9.1** (A Generated SPARQL Query).

```
{?event a dbpedia-owl:Event.
  OPTIONAL
    {?event dbpedia-prop:team ?team.
      FILTER (!isLiteral(?team))
      OPTIONAL
        {?team dbpedia-prop:team ?club.
          FILTER (!isLiteral(?club))}}}
```

When the SPARQL queries are executed, the result contains all the events with possible values for the specified path. These events with their associated data are minimally complete as a matching decision can be made upon them for the specified subscription.

### 7.9.4 Evaluation Metrics

Given a subscription  $S$  and an event  $e$ . Let  $ee$  be the enriched event of  $e$  according to  $S$ . Let  $e_m$  be the closest minimal successfully enriched event to  $ee$  according to Relations 7.11, 7.12, and 7.13 and their instantiation in Section 7.8.4. The following metrics are defined for evaluating the effectiveness of the enrichment approach:

$$Completeness = \frac{|ee \cap e_m|}{|e_m|} \quad (7.19)$$

$$Precision = \frac{|ee \cap e_m|}{|ee|} \quad (7.20)$$

$$F_5Score = \frac{(1 + 5^2) \times Precision \times Completeness}{5^2 \times Precision + Completeness} \quad (7.21)$$

The intersection is realized via an intersection between the set of triples that form each graph  $ee$  and  $e_m$ . The cardinality of events here is realized through the number of triples in the set that corresponds to each graph. The  $F - Score$  is a composite measure which is useful to summarize the effectiveness of an enrichment approach in one number for a subscription rather than two numbers. I argue that completeness and precision not be equally important. To evaluate an enrichment approach based on information completeness, the completeness measure should be given more weight. That is why the  $F_5Score$  is chosen in this evaluation as it gives much more importance to completeness. Depending on the application domain and constraints, other weightings may be considered.

### 7.9.5 Results

Figure 7.6 illustrates the combined  $F_5Score$  achieved by each enrichment approach for each subscription and averaged over events. The chart shows the superiority of the semantic relatedness-based approach and confirms the hypothesis that an enrichment approach, which makes benefit from the enrichment logic unified with the matching logic, is more effective than enrichment that is solely based on enrichment logic. There is also a trend showing that the enrichment effectiveness decreases for more complex

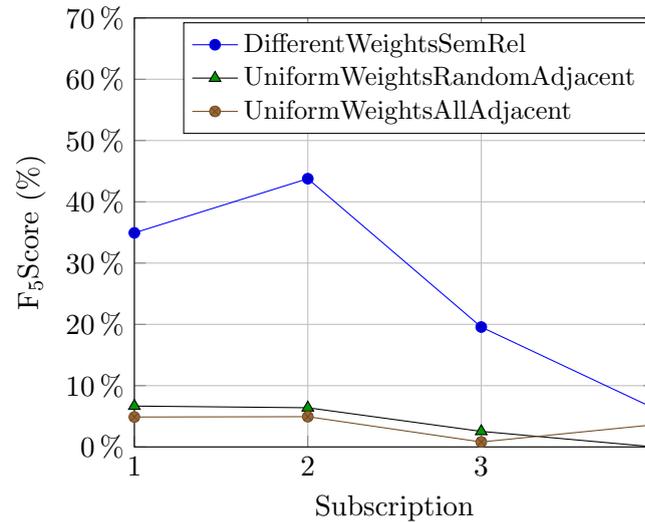


FIGURE 7.6: The combined  $F_5Score$  achieved by the enrichment approaches for each subscription

subscriptions. The decreasing effectiveness is because a longer property path requires more spreading to reach relevant triples while spreading may fade before that. From an empirical perspective, this raises the issue that the evaluation of an enrichment approach shall factor in the effect of the different types and complexities of subscriptions in the results.

The results show the validity of a dynamic enrichment model associated with semantic relatedness, and approximation for matching. It shows that the events completeness could be enhanced without coupling the participants in contextual agreements. Efficiency in terms of time is not measured, but in principle it is partially covered by the precision aspect, as more precision means that within a specific time, more relevant information is returned, i.e. the time is fixed while precision is varied.

Efficiency and effectiveness of the semantic relatedness-based instantiation are up to 44%  $F_5$ Measure, 7 times more than other instantiations of the enrichment model on average. Thus, the effectiveness and efficiency parts of hypothesis  $H3$  are validated with clear identifiable loosening in the pragmatic contextual coupling dimension represented by high-level expressions in the subscriptions to guide the enricher. The fact that the defined model assumes approximation and that the approximate semantic relatedness-based approach is superior also validates the pragmatic part of hypothesis  $H4$  on approximation.

## 7.10 Chapter Summary

This chapter constructs a model that realizes the element of dynamic native enrichment, along with the element of approximation. The rationale for using these elements as suitable models, which relax the effort needed to agree on contextual data, has been discussed. Events are assumed incomplete under an open world assumption. Incompleteness can be described on five dimensions: format, semantics, complementary background knowledge, complementary transformation, and temporal segmentation. This work is concerned with enrichment from complementary background knowledge.

Four elements of enrichment have been identified: determination of the enrichment source, retrieval of information items from the enrichment source, finding complementary information for an event in the enrichment source, and fusion of complementary information with the event. The proposed model recognizes enrichment as a core task of event processing engines. Additionally, the enrichment behaviour of an event processing engine can be dictated to the engine using a uniform and declarative mechanism.

The model proposed that the enrichment logic is described using a set of declarative language constructs similar to the ones used currently for matching purposes. Four language clauses that are mapped to the four enrichment elements were proposed: ENRICH FROM, RETRIEVE BY, FIND BY, and FUSE BY. All the enrichment clauses are described by the event consumer. The resulting subscription, which contains enrichment and matching elements, is called a unified subscription.

The model is formalized using set algebra. Concepts such as a complete event, valid enrichment, a minimal and approximately minimal successfully enriched event, have been defined according to the formal model. The model has been instantiated for Linked Data events and Linked Data enrichment sources. The instantiation uses spreading activation in Linked Data graphs, along with semantic relatedness.

The model has been evaluated with 24,000 events from DBpedia, a Linked Data version of Wikipedia. Results showed up to 44% F<sub>5</sub>Measure of enrichment precision and completeness, 7 times more than other instantiations of the enrichment model on average. The hypotheses *H3* and *H4* about the suitability of dynamic native event enrichment and approximation for efficient and effective loose pragmatic coupling have been validated.

## Chapter 8

# Prototype and Use Cases

“I think IT projects are about supporting social systems - about communications between people and machines.”

— Tim Berners-Lee

### 8.1 Introduction

This chapter discusses how the models in this thesis are combined into a working system called COLLIDER. The chapter details the COLLIDER system design and implementation. It discusses the concept of *thingsonomies*, which is based on the thematic event processing model for the Internet of Things, and how a practitioner can utilize it to build an IoT application. The chapter also details the employment of COLLIDER in two use cases: energy management and water management, to show how the system can work in real-world environments. The COLLIDER system has been demonstrated in the ACM International Conference on Distributed Event-Based Systems (DEBS 2013) [250].

Section 8.2 describes the design of the COLLIDER system and its implementation. Section 8.3 discusses the thingsonomy concept and architecture. Section 8.4 details the use case of COLLIDER for self-configurable energy management while Section 8.5 concerns a water management use case. The chapter ends with reflections on COLLIDER use in practice in Section 8.6, and a summary in Section 8.7.

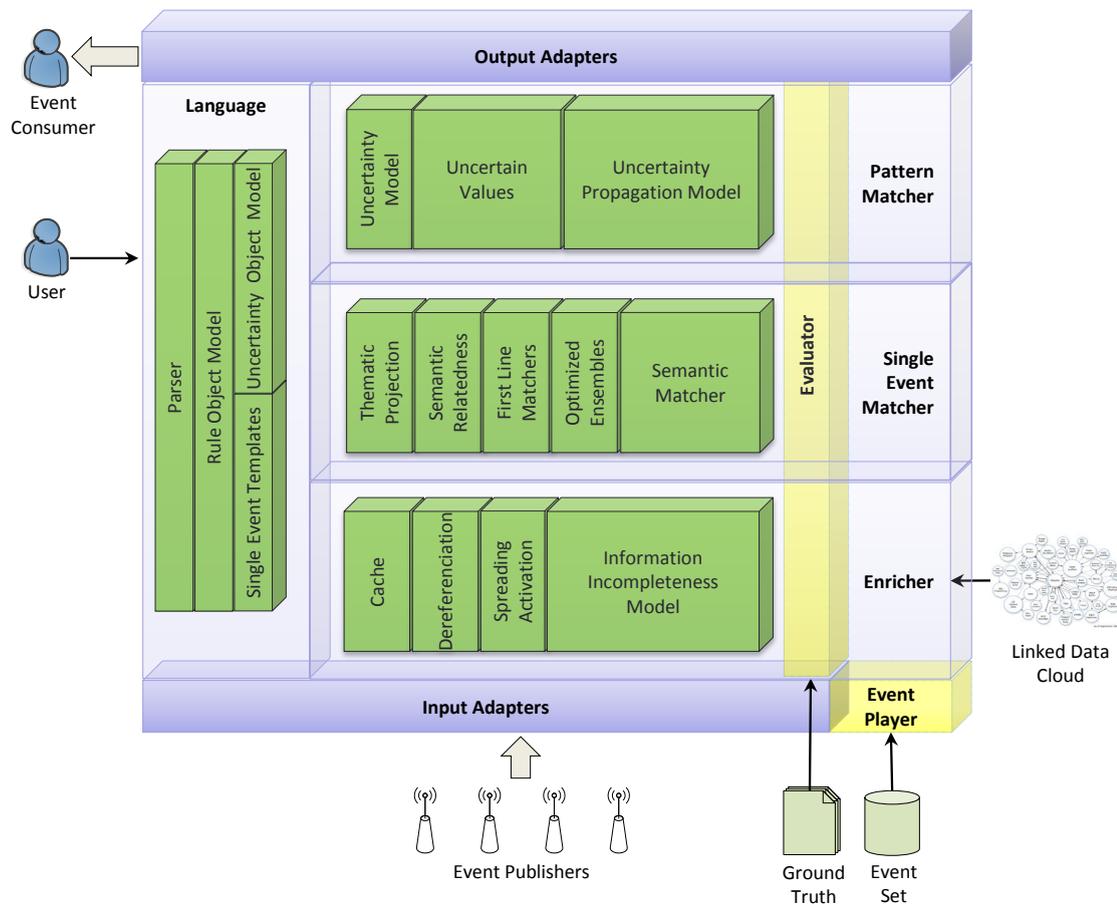


FIGURE 8.1: Internal architecture for the COLLIDER system

## 8.2 Internal Architecture

The main contributions of this thesis are the three models: the approximate semantic event matching model in Chapter 5, the thematic event matching model in Chapter 6, and the dynamic native enrichment model in Chapter 7. These models have been engineered into a system called *COLLIDER*. The architecture of *COLLIDER* is illustrated in Figure 8.1.

The main two layers of the system are the *Single Event Matcher* and the *Enricher*. The former implements the approximate semantic event matching model and the thematic event matching model. The latter implements the dynamic native event enrichment model. The other layers and components are necessary from a system perspective to enable events and rules input/output, evaluation, and to demonstrate the effect of the proposed approach on complex event processing.

The system is designed to be generic to facilitate experimental research with the models and to enable extensibility from a software engineering perspective in a non-laboratory setting. For instance, the element of approximation is realized in the system using a set of abstract interfaces such as `UncertainValue` and `UncertainSingleEvent`. Implementing classes of these interfaces define what uncertainty model is used. For example, an `UncertainValue` can be instantiated by a single numeric value reflecting probabilities, by two values reflecting belief and plausibility, or by another uncertainty model. The probabilistic model has been used throughout this thesis. The components of the internal architecture shown in Figure 8.1 are detailed in the following sections.

### 8.2.1 Input and Output Adapters

The *Input/Output Adapters* are responsible for connecting a COLLIDER agent with the outside environment which includes event producers and consumers. An adapter is instantiated for a particular input/output paradigm or technology such as HTTP adapters, JMS adapters, etc. Adapters also accommodate the necessary syntax level format handling such as dealing with multiple serializations of RDF messages and converting them all into corresponding COLLIDER events. Another task of some adapters is to map an event arriving at a particular URL or JMS topic with the respective internal in-memory channel so it can be considered for matching with rules that take input from this channel. In principle, the engine can be instantiated programmatically and fed with events without the need for adapters, but the programmer would be responsible for handling input/output events and adapt them to the system in this case.

### 8.2.2 Language

The language component is responsible for dealing with a user's input of rules and subscriptions. The *Parser* parses rules from plain text and converts them into their respective object models. A *Rule Object Model* in this context is a syntax tree that can be visited by other parts of the COLLIDER engine for matching purposes and uncertainty propagation. A rule can be just a single event template within the context of single event matching as considered in this thesis. However, more generally, the leaves in a rule syntax tree are single event templates, which are subject to matching by the single event matcher. The *Uncertainty Object Model* gets generated by an *Uncertainty*

*Propagation Model* object from the nodes of the *Rule Object Model*. It governs how uncertainty values resulting from the approximate matching gets propagated through a pattern to a derived event.

### 8.2.3 Enricher

The *Enricher* layer accommodates the dynamic native event enrichment approach proposed in this thesis. The interface `Enricher` can be instantiated using various mechanisms for enrichment, but the one followed in this work is the information incompleteness model which tries to fulfil the information content of an event before it gets matched. The particular instantiation of the `Enricher` illustrated in Figure 8.1 is based on *Spreading Activation* in Linked Data graphs as discussed in Section 7.8. A part of the process is represented by the *Dereferenciation* of URIs into their respect RDF representations. The *Cache* is a performance enhancement mechanism to avoid the high cost of retrieving a contextual piece of data which has already been retrieved.

### 8.2.4 Single Event Matcher

The *Single Event Matcher* layer accommodates the approximate *Semantic Matcher* and its relevant sub-components. *Thematic Projection* is responsible for calculating terms vectors with respect to thematic tags as discussed in Section 6.5. *Semantic Relatedness* and *First-Line Matchers* are responsible for producing similarity and relatedness scores from types, attributes, and values. *Semantic Relatedness* can be parametrized to deal with different measures or different indices of corpora. *First-Line Matchers* are organized into *Optimized Ensembles* to produce a single similarity matrix out of a set of matrices as discussed in Section 5.9.2. The resulting similarity matrix is used to produce top-1 and top-*k* mappings between events and single event templates.

### 8.2.5 Pattern Matcher

The *Pattern Matcher* layer has not been investigated in this thesis, but approximate probabilistic matching, and top-*k* mappings have effects on how pattern matching takes place. The key to this is the use of *Uncertain Values* and an *Uncertainty Model* which

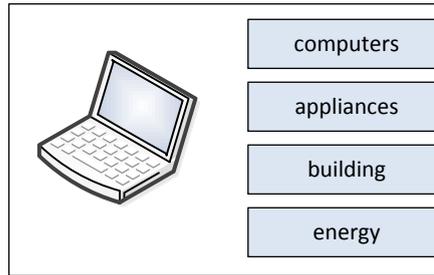


FIGURE 8.2: An example thingsonomy for tagging a device's events

defines the meaning of a matching score, such as being a probability, a two-valued belief-plausibility score, etc. It defines how reasoning over the pattern can take place and how the uncertainty values of single event matching in a pattern are propagated into a derived event according to the *Uncertainty Propagation Model*.

### 8.2.6 Event Player and Evaluator

The *Event Player* and *Evaluator* components are primarily designed to enable empirical evaluation with COLLIDER. The *Event Player* defines how a set of events, e.g. in a log file, can be streamed into the engine. The *Evaluator* is responsible for contrasting the matcher's and enricher's results with a ground truth to provide evaluation metrics. In a non-empirical setting, the *Evaluator* can still be used to measure the performance of a real-world deployment of the system.

## 8.3 Thingsonomies for the Internet of Things

In this section, I show how an IoT system can be built around the event processing approach described in this thesis. Bob works in the town hall planning department of a smart city. Bob is interested in finding the energy usage of street lights during peak electricity usage in different areas. Such information can be detected using an expression of an Event Processing Language (EPL) such as Esper's language [48] as follows:

```
every a=StreetLightsEvents(  
    a.type= 'energy consumption event'  
    and a.area.consumptionPeak='true')
```

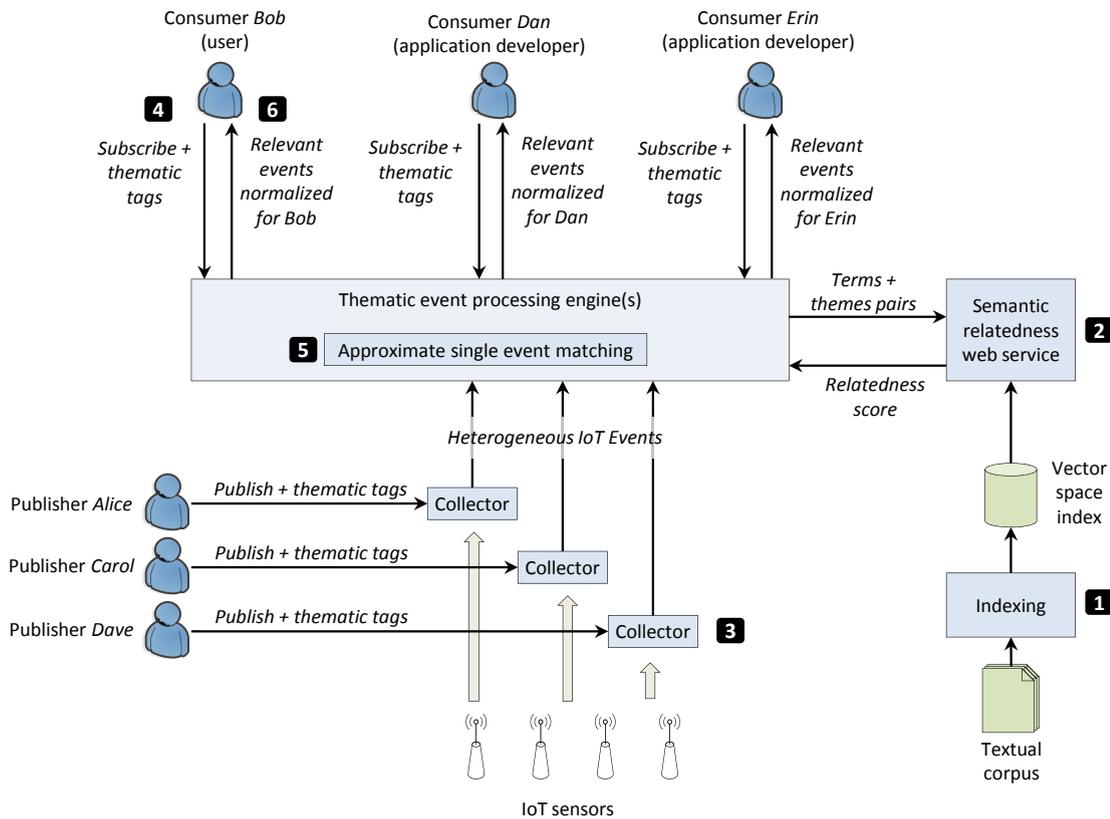


FIGURE 8.3: Architecture for loosely coupled semantic normalization for Internet of Things software

While the sources of required information are available from the street lights, the semantics of the events differ from one area to another due to different sensors manufacturers. For instance, events contain terms such as ‘*energy consumption*’, and ‘*electricity usage*’ to refer to the same thing. The scenario requires a large set of rules with high definition and maintenance costs to cover the semantic heterogeneity of events.

I suggest thematic tags that describe the themes of types, attributes, and values to clarify their meanings as suggested in Chapter 6. I call these tags *thingsonomies* for *things* and *taxonomies* in the context of the Internet of Things. The hypothesis is that associating events and subscriptions with extra tags can improve effectiveness and time efficiency in heterogeneous environments and domain-specific knowledge exchange as validated in Chapter 6. Figure 8.2 shows an example thingsonomy for tagging energy consumption events coming from a laptop.

Figure 8.3 illustrates the main components of the system and the main steps required by the practitioner. The underlying thematic event matching model, which enables this architecture, has been investigated in Chapter 6.

**Step 1** to build the IoT architecture enabled with semantic normalization is to build a semantic model that enables the system to establish automatically relationships between various terms such as ‘*computer*’ vs. ‘*laptop*’. The proposed approach adopts a subsymbolic distributional model of semantics based on statistical indexing of a large corpus of textual documents. Such a model is easy to build automatically as shown in [175], and the main task for the practitioner is the corpus selection. One can start working with an initial documents corpus, e.g. Wikipedia, and incrementally revise it to suit the use cases.

**Step 2** is to avail a semantic relatedness measure based on the built semantic model through a conventional interface such as REST and JSON [175]. For example, a request for relatedness between ‘*electricity*’ and ‘*energy*’ is invoked through API:

```
http://example.com/esa?term1=energy&term2=electricity
```

with the result being returned as a JSON object as follows:

```
{“relatedness” : 0.154}
```

Such a result makes sense only in comparison with the relatedness of other pairs of terms such that ‘*electricity*’ is closer to ‘*energy*’ than to ‘*office*’ for instance.

**Step 3** is for publishers who shall accompany their events with a set of thematic tags at the data collector. Such tags shall represent approximately the domain and meaning of the terms used to describe the event attributes and values. Let an event of an *increased energy consumption* be represented as follows:

```
{type: increased energy consumption event,  
measurement unit: kilowatt-hour,  
device: computer}
```

An example of thingsonomy tags for this event are:

```
{computer, appliances, building, energy}
```

**Step 4** is for subscribers to associate their subscriptions with thingsonomy tags. I propose using the language from Chapters 5 and 6 which leverages the *tilde*  $\sim$  operator to signify that the user wants the matcher to match the term used or any term semantically similar to it. A subscription for *increased energy consumption* can be represented as follows:

$$\{\mathbf{type= increased\ energy\ usage\ event\sim},$$

$$\mathbf{device\sim= laptop\sim}\}$$

Example thingsonomy tags are:

$$\{\mathbf{power, computers}\}$$

**Step 5** is the responsibility of the system to normalize events and match them to the suitable subscriptions. The example event and subscription do not use exactly the same terms to describe the type or the device, hence *'energy consumption'* vs. *'energy usage'*, and *'computer'* vs. *'laptop'*. Nevertheless, the event should not be considered as a negative match to the subscription. For this reason, the system employs a probabilistic matcher that uses a measure to estimate semantic similarity and relatedness between various terms. Functionally, it tries to establish possible mappings between subscription predicates and event tuples. For example, the most probable mapping of previous examples is described as follows:

$$\sigma^* = \{(\mathbf{type=increased\ energy\ consumption\ event}$$

$$\leftrightarrow \text{type:increased\ energy\ usage\ event}),$$

$$(\mathbf{device\sim = laptop\sim} \leftrightarrow \text{device:computer})\}$$

**Step 6** represents the return of events matching a subscription to its initiator. The matcher establishes probabilistic matching and, as a result, forwards the normalized event along with an uncertainty value that reflects the amount of semantic normalization that has been conducted all the way from publishers to subscribers.

The COLLIDER system has been successfully employed in the context of two projects concerned with energy management and water management domains as discussed in the following sections.

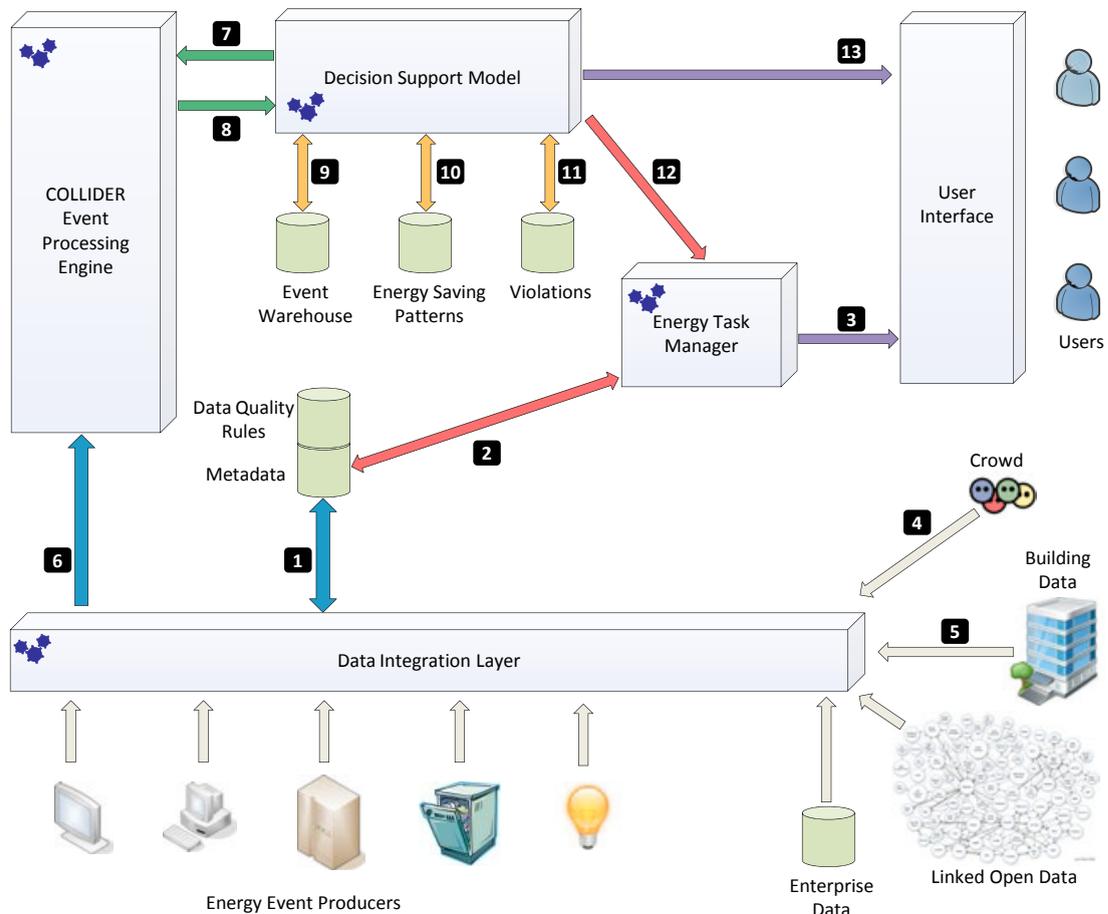


FIGURE 8.4: The role of COLLIDER in the Self-Configurable Energy Management Systems Use Case

## 8.4 Self-Configurable Energy Management Systems Use Case

The COLLIDER system has been used within the SENSE project <sup>1</sup>. The project mission statement is to:

“Deliver technology required to create a self-configuring smart energy management systems for small commercial buildings.”

SENSE has been deployed initially in the Digital Enterprise Research Institute (DERI) at the National University of Ireland, Galway which had hosted the Sustainable DERI and DERI Energy works [218, 251–253]. The building has been equipped with energy consumption sensors. In total there have been over 50 fixed power sensors covering offices,

<sup>1</sup><http://sense-project.com/>

café, data centre, kitchens, meeting rooms, and the computing museum. Additionally, there have been over 20 mobile energy sensors for computing and printing devices, lights and heaters, as well as light detection, temperature, and motion detection sensors.

Figure 8.4 illustrates a high-level architecture of the project. To achieve the goal of self configuration in energy systems, the following objectives have been addressed as shown in Figure 8.4:

- Delivery of a multi-level decision support model for energy management.
- Integrating energy data from sensors, enterprise, and open data through the use of a Linked Data integration layer.
- Design and implementation of persuasive user interfaces for decision support.
- Design and implementation of approximate semantic event processing for energy data.
- Design and implementation of a collaborative crowdsourcing platform for energy data management tasks.
- Validation of the performance of the project output.

Figure 8.4 shows the relative relationship of COLLIDER to the integration layer as the primary source of events and contextual data, and to the energy decision support model, which is the main consumer of energy waste and saving patterns.

#### 8.4.1 Event Processing Requirements in the Use Case

This use case forms a good fit for motivating the use of COLLIDER. This stems from the requirements of self-configurable energy management systems as follows:

- *The self configuration requirement for energy systems* is essential to easily integrate energy-related resources in a building and putting a system up and running with a low effort from building personnel. Consequently, building occupants become increasingly aware of energy usage and become able to detect energy saving opportunities to help the environment and lower the costs. COLLIDER meets

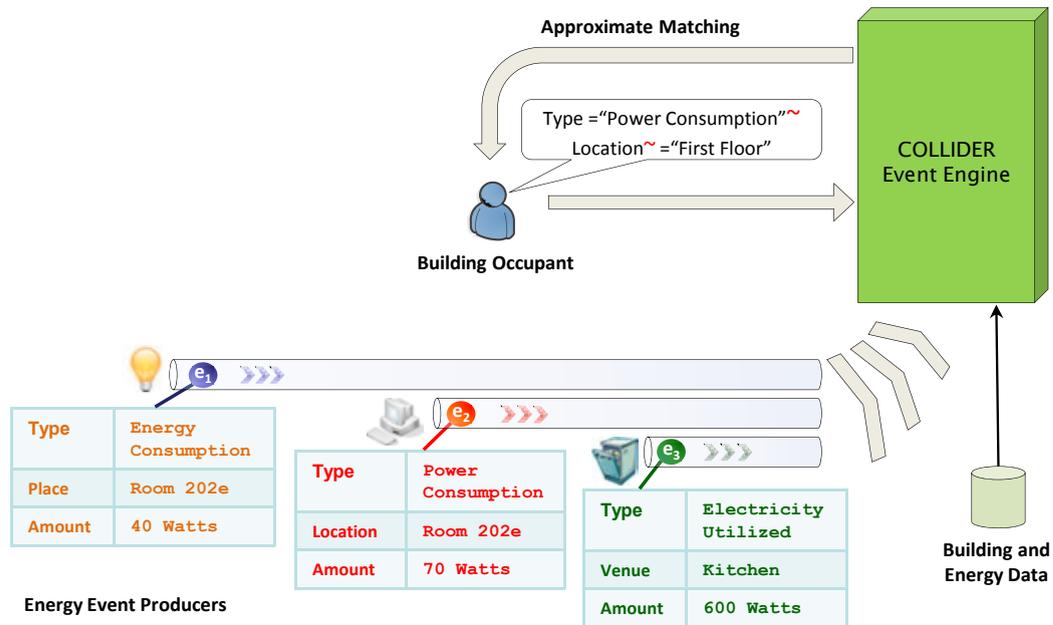


FIGURE 8.5: An energy domain-specific scenario

this requirement through its support of a loosely coupled semantic and pragmatic event processing paradigm. Thus, it helps reduce the efforts by the users and shift the semantic and pragmatic configuration tasks into the event engine.

- *The requirement of decision support* that includes: typical flow of energy, contextual business and building entities for energy management, and energy saving patterns. COLLIDER meets these requirements through its layers and underlying models. The flow of energy can be handled through the single event processing from sensor feeds. The dynamic enricher can handle contextual energy data. Energy saving patterns can be handled by the complex pattern matcher.
- *The real-time detection requirement of energy wasting patterns* which can be tackled by the reactive behaviour of an event engine such as COLLIDER. COLLIDER can meet this requirement as it is built upon efficient models in terms of throughput and latency.

Figure 8.5 illustrates an energy-specific use case of COLLIDER within SENSE. It shows how several energy consumption devices can use various descriptions of event data such ‘energy’, ‘power’, and ‘electricity’ while they probably mean the same thing which is energy consumption. The user can depend on COLLIDER to resolve this issue and to

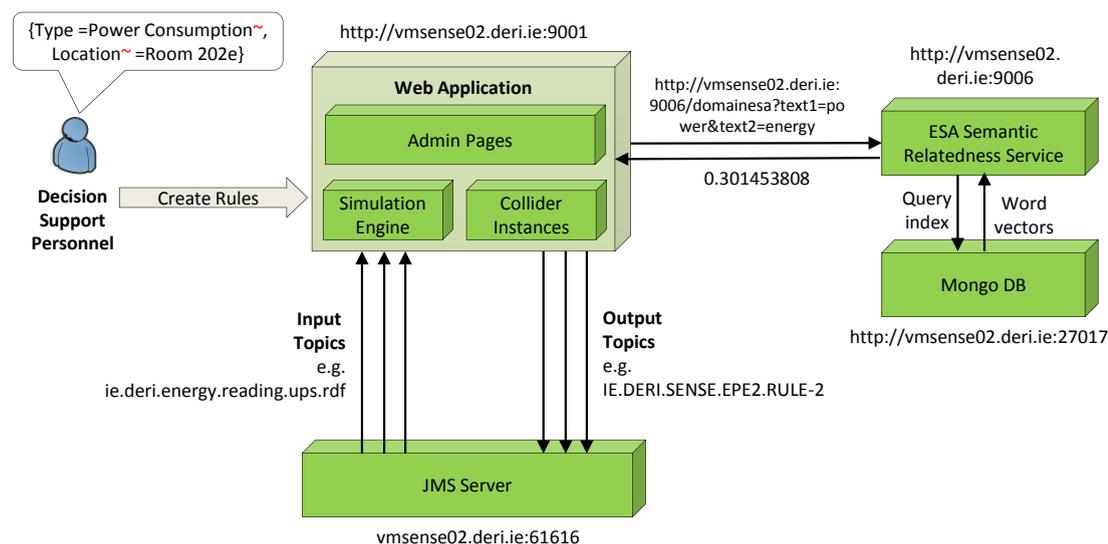


FIGURE 8.6: COLLIDER and semantic relatedness within the SENSE project

enrich events with their ‘*floor*’ for instance in addition to the ‘*room*’. Consequently, users need less *configuration effort* to get benefits from the system.

#### 8.4.2 COLLIDER Implementation for Energy Management

COLLIDER has been designed in SENSE to be deployed as running instances of the engine, which by themselves are contained in a web application server as shown in Figure 8.6. The web application includes besides the COLLIDER instances a set of administration pages and a simulation engine. The administration pages allow the administrator, e.g. the building administrator, to create new engine instances, configure the feeding JMS topics, and the output topics. They also enable the administrator to deploy new rules with the engine instances and configure the default semantic relatedness measure to use. Administrators can also monitor the engines performance constantly.

The simulation engine allows the user to define a set of virtual sensors and zones to simulate a building environment. Virtual sensors can be configured with templates of event data where some parameters such as energy consumption values get generated at random during simulations. The rate of incoming events can also be configured. The simulation engine is also equipped with an interactive user interface as shown in Figure 8.7 which allows the user to drop new sensor instances within zones to emulate the sensor deployment. Real-time readings from sensors are pushed to the web client using web sockets to create an interactive simulation scenario. Users can at any time view the

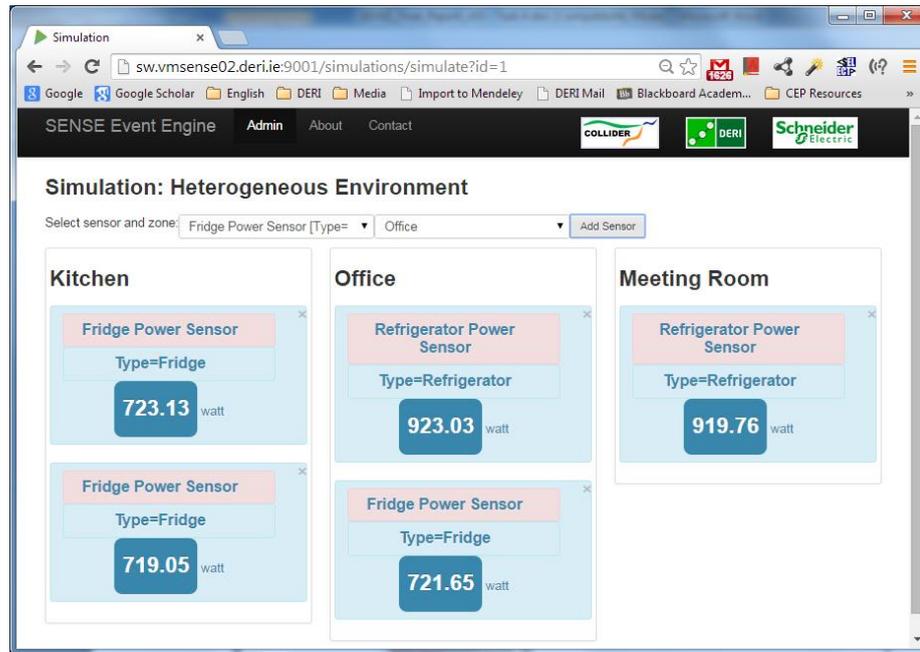


FIGURE 8.7: COLLIDER simulation engine

rules matching results, i.e. the uncertain event matchings, which are also pushed to the JMS server topics.

COLLIDER instances also make use of deployed semantic relatedness web services in the SENSE deployment. The semantic relatedness service works over an indexed corpus of energy domain Wikipedia articles. The index is hosted by a Mongo DB service as described in [175]. The semantic relatedness scores are returned to the COLLIDER instances which then use them for probabilistic matching.

### 8.4.3 Building an Energy Domain Corpus for Semantic Relatedness

A domain-specific measure is achieved by building a domain-specific semantic model. A domain-specific semantic model is built via a domain-specific thesaurus or ontology for thesaurus-based measures, and created by a large set of domain-specific text documents for distributional semantic models. The biomedical domain is one of the most thriving domains with respect to semantic measures.

Pedersen et al. [254] adapt a set of thesaurus-based and distributional-based measures for the biomedical domain. Rada et al. [255] build a semantic distance measure based on the MeSH semantic network. Caviedes and Cimino [256] devise a measure for finding path lengths in a Unified Medical Language System (UMLS) hierarchy. Lord et al.

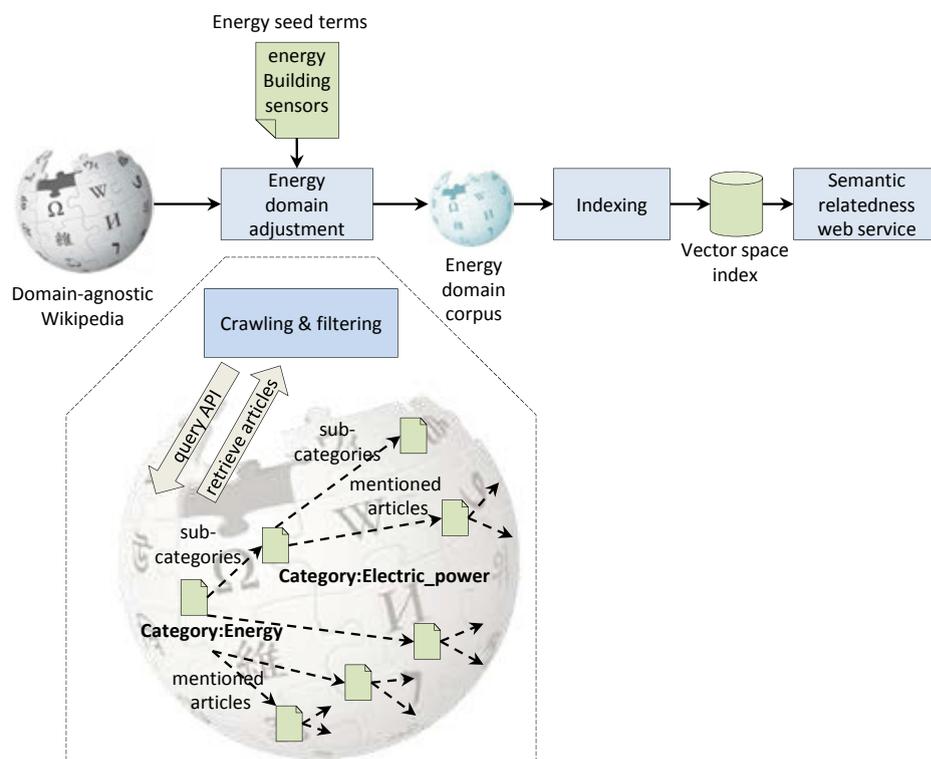


FIGURE 8.8: Building an energy domain-specific corpus and semantic relatedness for energy management

[257] adapt WordNet-based measures to the Gene Ontology. Latent Semantic Analysis (LSA) has been used for indexing clinical records and classifying medical events [258], for example, Pedersen et al. in [254] use LSA to build a vector-based measure.

Reviewing the literature reveals a shortage of works on semantic similarity or relatedness measures for the energy management domain. Nonetheless, semantic *models* for the energy domain exist, namely ontologies for the energy domain. Semantic models for building energy management have been investigated in [242] and [259]. An ontological framework for energy saving in intelligent smart homes has been proposed by Grassi et al. [260]. Devices description plays an significant role in energy applications and has lead to several standards for device description: Composite Capabilities/Preference Profiles (CCPP)<sup>2</sup>, the Generic Station Description Markup Language (GSDML), and the Field Device Configuration Markup Language (FDCML) [261–263].

There has also been work to describe components of power systems. The International Electrotechnical Commission (IEC) has published two standards IEC 61970-301 [264] and IEC 61968-1 [265] which serve as a Common Information Model (CIM). Several

<sup>2</sup><http://www.w3.org/Mobile/CCPP/>

works also use the Web Ontology Language (OWL) to encode CIM models [266, 267]. The work in [268] provides a CIM extension to support photovoltaic and wind power generation in addition to battery energy storage units. Such approaches can be classified under the symbolic family of models as discussed in Section 5.4 which requires an effort to establish granular agreements and thus affect semantic coupling. I am not aware of subsymbolic distributional energy semantic models. Furthermore, there seems to be a lack of semantic similarity and relatedness measures upon the existing energy-specific ontological models.

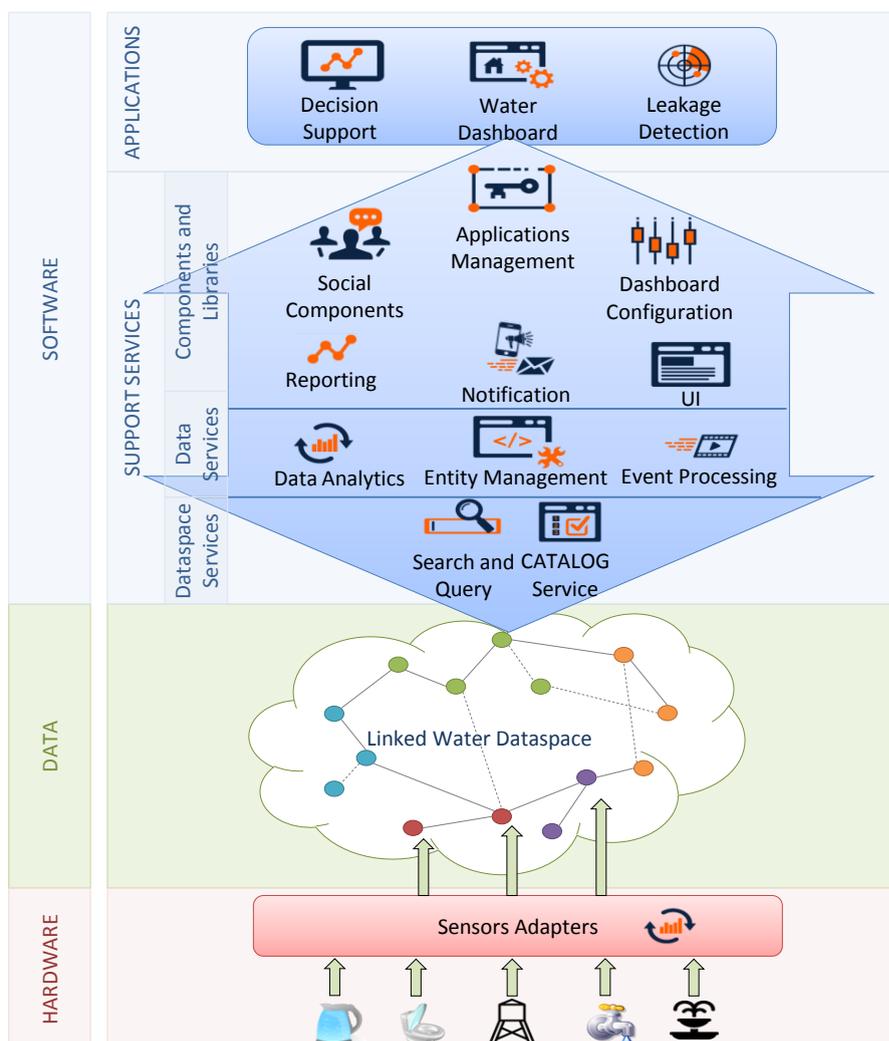
The approach I follow for building an energy domain corpus that works with COLLIDER is by filtering a Wikipedia-based domain-agnostic distributional model, developed by Freitas et al. [174, 175, 200], as shown in Figure 8.8. The domain adjustment process is fed by a set of seed terms that could represent the domain of interest, i.e. ‘*energy*’, ‘*building*’, and ‘*sensors*’ within the context of SENSE. The seed terms are then used to search the Wikipedia API for relevant categories, and then follow recursively their sub-categories and mentioned articles. The resulting visited subset of articles are the ones chosen for indexing to build the ESA model, or filter an existing one for semantic relatedness.

#### 8.4.4 Validation within Self-Configurable Energy Management Systems

Running the COLLIDER system over the energy management deployment has shown that event matching is done with a latency as small as 0.09 millisecond/event. This is a small number to react to energy events within a smart building or enterprise scenario. COLLIDER could get up and running before full ontological agreements on data description were achieved. Thus, it requires a small effort to deploy events and rules where the semantic mediation is left to the engine. This fits the self-configuration goal of the use case.

## 8.5 Water Management Use Case

The COLLIDER system has been employed within the European Waternomics project [269]. Waternomics investigates how information and communication technology can

FIGURE 8.9: Waternomics overall architecture <sup>3</sup>

help households, businesses, and municipalities reduce water wastage and consumption. Building a data platform to collect water usage and contextual data, as well as analysing the integrated data is key to water management from the perspective of Waternomics.

The water management task is designed within this use case into three layers as shown in Figure 8.9. The bottommost layer is hardware, where water sensors such as water consumption and leakage detection hardware are developed, configured, and instrumented. The middle layer is concerned with the water data collected from various sources including sensors and contextual data from systems such as building management systems. The topmost layer is concerned with the software that operates over the data to ultimately help users better manage their water resources.

<sup>3</sup>Adapted from deliverable D3.2 (<http://waternomics.eu/>)

The software layer of the use case is further divided into support services, which contribute to the infrastructure, and applications which give direct decision support tools to users such as dashboards, leakage detection apps, etc. Support services are classified into three classes: dataspace support services, data services, and component libraries. Dataspace support services help to manage the space of data underneath using services such as cataloguing of data sources, searching, and querying. Data services concern the further support of the dataspace at a higher level, including entity management, data analytics, and event processing. Component libraries give domain support for applications using services such as reporting, notifications, social services, applications management, configuration, and user interfaces.

### 8.5.1 Event Processing Requirements in the Use Case

This use case forms a good fit for motivating the use of COLLIDER. This stems from the requirements of water management systems as follows:

- *The requirement of real-time water data processing*, which is crucial to identify water leakage and other waste scenarios. COLLIDER meets this requirement through its underlying efficient models of matching in terms of throughput and latency.
- *The requirement to handle heterogeneity of water sensors*, which results from the large number of parties involved in the water lifecycle. COLLIDER meets this requirement through its underlying support for semantic matching and its ability to deal with semantically loosely coupled parties.
- *The requirement of consuming open water data*, which is useful to make predictions about water consumption. An example of open water data is weather precipitation forecast which could affect water distribution and cost as well as usage scenarios. COLLIDER meets this requirement through its underlying event enrichment model which helps to fuse open contextual data with water events.

FIGURE 8.10: Waternomics Linked Water Dataspace components view <sup>4</sup>

### 8.5.2 Linked Water Dataspace

Waternomics adopts a dataspace approach to data management. Dataspaces have been proposed by Franklin, Halevy, and Maier in [270, 271] as a new abstraction of information management as opposed to databases. Databases reflect a well-controlled environment with a relatively centralized administrative authority of data sources. Dataspaces on the other hand concern the management of diverse and loosely coupled data sources which co-exist but not strictly integrate.

Waternomics emphasizes the role of Linked Data as an enabling technology for the water dataspace. A set of services exists to support the resulting Linked Water Dataspace (LWD) as shown in Figure 8.10. The dataspace represents an incremental view of how water datasets join the computational space targeted by applications. In contrast to the

<sup>4</sup>Adapted from deliverable D3.2 (<http://waternomics.eu/>)

classical one-time integration of datasets that causes a significant overhead, the LWD adopts a pay-as-you-go paradigm. Water datasets join the space in an incremental manner: the more interfaces they expose, the more links they provide; and the more linked dataspace services they support, the more integrated into the dataspace they become.

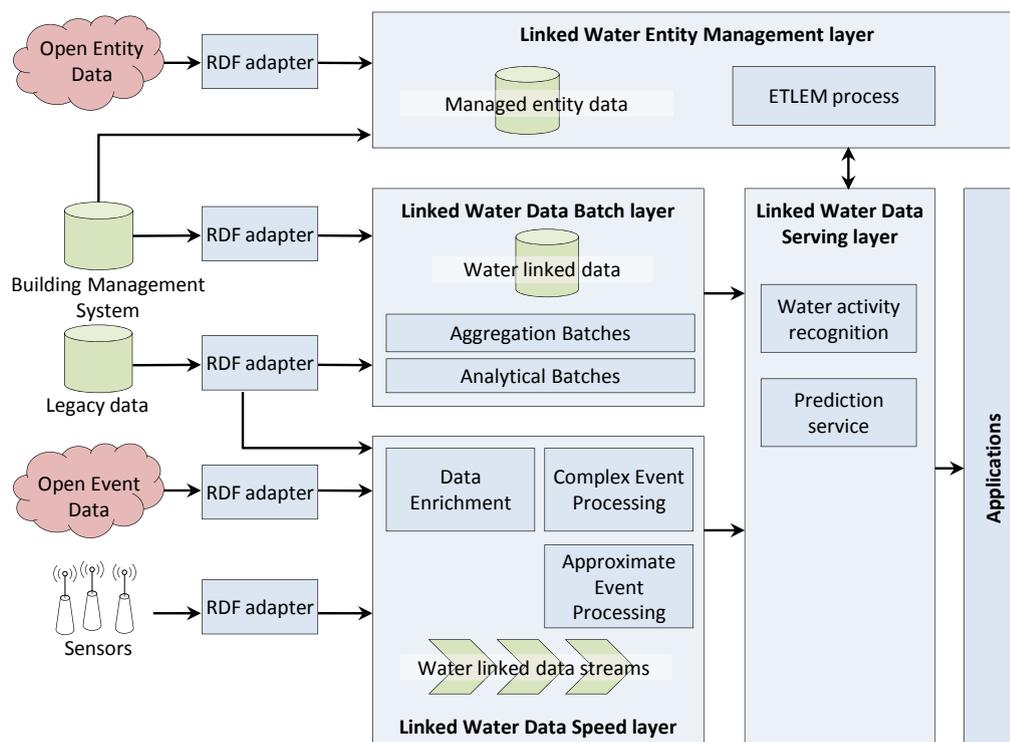
Figure 8.10 illustrates the components view of the LWD witch includes:

- *Datasets* include sources such as weather data, water sensor data, building management system data, etc.
- *Adapters*, or interfaces, are the technical facades of the datasets that other members of the dataspace can talk to. A dataset that provides a JavaScript Object Notation for Linked Data (JSON-LD) interface, for instance, allows structured queries to be executed, and thus it is superior and more integrated into the dataspace than a document that is only exposed by keyword search.
- *Services* are the platform that allows datasets to be visible, query-able, integrable, searchable, monitorable, and curatable. Services include catalogue, query, search, human, monitoring, and event services. Such services form the support platform in dataspace [270].

Beside those three concepts, there is the notion of a *relationship* between two datasets. For example, the Kafka middleware *feeds into* the event service. Finally, applications surround the dataspace and make use of its services to interact with the datasets.

The Linked Water Dataspace is equipped with a real-time aspect to enable the processing of information items within a short period of time. That is done through the adoption of in-flow processing which means that data items are not stored and indexed in order to be processed, but rather they undergo enrichment, matching, aggregation, and pattern detection as they flow in streams. COLLIDER is employed to achieve this goal.

The technical architecture of the dataspace is illustrated in Figure 8.11. It is based on a *Lambda* architecture that aims at the seamless integration of stream processing and batch processing. Lambda architecture emphasizes three main layers: the speed layer, the batch layer, and the serving layer. The speed layer is concerned with manipulating real-time data continuously. The batch layer is concerned with calculating views over

FIGURE 8.11: Waternomics Linked Dataspace architecture <sup>5</sup>

stored data. The serving layer provides a transparent query interface to the user for both real-time and batch data.

COLLIDER is used in the speed layer through three main functionalities: approximate matching, event enrichment, and complex event processing. For instance, COLLIDER can enrich water consumption events with their source ‘room’. It can approximately match those described by ‘bathroom’ and ‘restroom’. It finally can use various events such as ‘tap opened’ followed by ‘kitchen unoccupied’ to detect a water wastage situation.

Additional to the energy management use case, this use case gives more emphasis to unstructured events, the thematic tagging facet of COLLIDER, as well as the complex event detection. Unstructured events such as images, the behaviour of sensor thingsonomy tagging by crowd users, as well as its effect on matching are objectives of investigation. The effect of uncertainty propagation into complex water usage patterns is also an aim of the work in this use case.

<sup>5</sup>Adapted from deliverable D3.1.1 (<http://waternomics.eu/>)

## 8.6 Reflections on COLLIDER in Use

Putting the proposed approach via the COLLIDER system into practice has revealed a number of insights as follows:

- In real-world scenarios of small-to-medium scales, events do not fully occupy the time scale when arriving at the event engine. Thus, throughput does not reflect the full picture of the event engine's performance without being complemented by latency. For this reason, latency has been put into use when building systems for the use cases presented in this chapter.
- Latency could be within the range of milliseconds to seconds in small-to-medium scales. That appears to be sufficient in applications with no security or time-critical requirements.
- Approximate matching by itself results in scored mappings between subscriptions and events. Nonetheless, normalization of events to the subscriptions they matched can be very useful for further processing of matched events by other components in the system where the event engine is deployed.
- Finding a ready-to-use corpus to build a distributional semantics relatedness service may be challenging within corporates. Thus, it is worth investigating the possibilities of indexing structured data available in companies. Another line is also to develop novel methods to adjust domain-agnostic indices to domain-specific indices, similarly to what has been presented in this chapter.
- The COLLIDER system does not incorporate thresholds on scored event matching natively. That is in order to allow propagation of uncertainties for potential further processing. However, the use of thresholds and mechanisms to allow that can be important within real-world applications to cut down the number of matched events and allow users to act upon situations of their interest.
- Tagging has been meant for events mainly in this work which is crucial to improve their understanding. However, users might also find it more intuitive to tag entities such as locations or sensors. Thus, extending the thematic model to account for this can be very beneficial within real-world scenarios.

- Enrichment is a very valuable functionality within real-world scenarios as users would typically match events on high level information not included in events, but rather existent in companies's data assets such as spreadsheets or databases. Thus the access to enrichment sources shall be facilitated by, for instance, serving sources as Linked Data and providing them through appropriate data portals.
- Caching of semantic similarity scores and enrichment data can return significant performance gains in real-world scenarios due to repetitive patterns of use. Thus, this factor shall be more investigated and developed in future work.
- Surrounding components of an event engine such as user interfaces for administration and monitoring, as well as input/output adapters for connecting the engine with sources such as event buses are very important to facilitate the use of the overall system by end users.
- Simulation environments, such as the one presented in this chapter, can reduce the barrier to end users to understand COLLIDER and its approximate paradigm.
- Having an explicit linkage between events and application-specific situations can allow a quantitative evaluation of the impact of the technology on domain-specific application targets such as reducing costs or CO<sub>2</sub> emissions which are important measures within real-world use cases.

## 8.7 Chapter Summary

This chapter shows how the proposed approach can be employed in an Internet of Things architecture. Events arriving from things, and subscriptions of users are tagged using thingsonomies, which are then used to enhance semantic matching. The chapter recognizes the components of a working system that realizes the functionalities of the proposed approach. The resulting system, which is called COLLIDER, is designed through the three main layers: enricher, single event matcher, and pattern matcher. Additionally, the components of language, input, and output adapters are considered.

COLLIDER has been exploited within the use case of self-configurable energy management systems in the SENSE project. It has been supported with an administration application, a simulation engine, and an energy domain-specific distributional semantic

model. Running COLLIDER over the energy management deployment showed a latency for event matching of 0.09 millisecond/event.

COLLIDER has also been employed within the water management domain in the European Waternomics project. The use case emphasizes the concept of Linked Water Dataspace for data management, including events. COLLIDER realizes the event support service in the architecture. The technical implementation adopts a Lambda architecture of speed, batch, and serving layers, with COLLIDER being a part of the speed layer. More emphasis is given to image-based events, crowd tagging, and complex event processing for water usage and saving scenarios.



## Chapter 9

# Conclusions and Future Work

“We can only see a short distance ahead, but we can see plenty there that needs to be done.”

— Alan Turing

### 9.1 Thesis Summary

A significant shift in the data landscape has been taking place through recent years. This shift can be described by a set of characteristics: an increasing number of data sources and users, an increasing heterogeneity, a loose organization of users, a need to process information in time as it flows, and an incompleteness and uncertainty of data. The Internet of Things (IoT) is one area where these characteristics are realized as tens of billions of devices are expected to connect to the Internet in the coming years within smart cities, smart grids, and cyber-physical systems.

Event processing systems represent a computational paradigm to exchange data items by distributed and potentially heterogeneous producers and consumers. Thus, interoperability is a key requirement for such a paradigm. However, the current approach to this problem is done through top-down granular agreements represented by ontologies and taxonomies for semantics. Such approaches are non-scalable and achieving such agreements may be unfeasible under the characteristics of current and future event and data environments.

This thesis analysed this problem using a decoupling versus coupling trade-off framework. The principle of decoupling allows event systems to scale as it removes interdependencies between interacting parties. This thesis viewed events as boundary objects, and event exchange as a model of crossing system boundaries where every event agent is a system in an overall system of systems. Boundaries are syntactic, semantic, and pragmatic and events shall cross them all for effective interoperability and communication.

Current event-based systems are decoupled on three dimensions as they do not hold references to each other (space), they are not active simultaneously (time), and they do not block each other (synchronization). While this decoupling enables scalability, events can still cross syntactic boundaries and achieve syntactic interoperability. Nonetheless, human agents are needed in the loop to cross semantic and pragmatic boundaries and address interoperability through explicit agreements on event types, attributes, values, and contexts, introducing coupling into these systems and limiting scalability. I recognized this problem as two new dimensions of coupling: semantic coupling and pragmatic coupling.

This thesis tackled the trade-off problem between coupling through agreements which is needed for event-based interoperability, and decoupling needed for scalability. The thesis formulated two research questions of how this trade-off can be approached on the semantic and pragmatic dimensions. By analysing the related literature, I found that the current event processing approaches mainly depend on an exact model of matching. This can be intolerant towards semantic and contextual loose agreements and heterogeneity in event environments. Thus, an approximate model has been proposed with probabilities as the main outcome of event matching and enrichment.

Besides, current event processing use symbolic models of semantics such as ontologies and taxonomies. These models require granular agreements on concepts and terms and thus can be coupling and non-scalable. I proposed the use of a statistical vector space model of semantics, which are based on term co-occurrence in large corpora. Such models are geometrical and naturally supports similarity and distance so they can address heterogeneity. This model is accompanied by free tagging, or thingsonomies, which enhance meaning representation and disambiguation. The agreement on a corpus and the use of free tagging is a coarse-grained process that represents a loose semantic coupling.

Regarding event contexts, the current event processing paradigm interprets events under a closed world assumption. That is, when an event lacks a piece of information, it is considered as a negative match. Enrichment is done outside the event engine using dedicated enrichers which depend on full understanding of the required context and its fusion within the event, entailing a non-scalable pragmatic coupling. This thesis proposed a dynamic native event enrichment model which tries to complement events before they are considered for matching. This process is guided by high-level clauses to tell the enricher how to access the context. The dynamic enrichment search the context based on each event and each subscription to decide on what information is used for enrichment.

The ability of the proposed elements and models to answer the research questions has been formulated into hypotheses which have been validated empirically. To evaluate the approximate semantic event matching model, an evaluation event set of 50,000 events has been synthesized out of seed event sets from real-world deployments of IoT smart cities, energy management, building, and relevant datasets. The set has been semantically expanded to reflect a large-scale heterogeneous event environment. Similarly, a set of 14,743 events has been used for evaluating the thematic event matching model, and 20,000 Linked Data events to evaluate the dynamic native enrichment approach.

The experiments have shown, as summarised in the next section, that the proposed approach which is based on approximate semantic event matching, free tagging, and dynamic native event enrichment can loosen agreements on semantics and pragmatics and thus enable scalability. It can at the same time support interoperability, and efficient and effective event-based communication.

## 9.2 Thesis Conclusions

The main conclusion of this thesis is that event exchange which uses subsymbolic statistical event semantics, free tagging, dynamic native event enrichment, and approximation, can effectively and efficiently loosen semantic and pragmatic coupling leading to scalability in open, distributed, an heterogeneous environments. It thus outperforms exact event processing which depends on symbolic semantics and dedicated enrichers, which

require a significant level of semantic and contextual agreements that can limit scalability in large-scale environments. The four elements of subsymbolic event semantics, free event tagging, dynamic native event enrichment, and approximation provide the answer to the research questions set out for this thesis as follows:

- *Q1.* The first research question has been concerned with the case when event producers and consumers do not have exact, granular, and rigid agreements on terms used in events and rules and their meanings but rather a form of statistical loose agreements on the meanings. The question is how to achieve timely event matching with high true positives and negatives in such a loosely semantically coupled environment?
- *Q2.* The second research question has been concerned with the case when event producers and consumers do not have equal assumptions on the amount of contextual information included in events and how much they are complete with respect to evaluating some consumers' rules. The question is how to complement events with context at high precision and completeness needed to meet consumers expectations in such a loosely contextually coupled environment?

This broad conclusion can be broken down into the conclusions drawn from the analytical and empirical testing of hypotheses discussed throughout this thesis as follows:

*Hypothesis H1: Subsymbolic distributional event semantics decreases the cost needed to define and maintain rules with respect to the use of terms, and to build and agree on an event semantic model more than symbolic semantic models; and at the same time it can achieve timely event matching with high true positives and negatives of magnitudes comparable to that of event processing based on semantic models.* This hypothesis has been validated through the investigation of the proposed approximate semantic matcher in Chapter 5 due to the following results:

- *Loose semantic coupling:* the subsymbolic semantic model requires a coarse-grained agreement on semantics rather than granular agreements on symbols. That is also manifested by the result that 100 approximate subscriptions in the proposed model compensate for 74,000 exact subscriptions otherwise needed in a symbolic model-based event engine. The results suggested that the best use cases for the proposed

model are where small-to-medium degrees of approximation are expected with the user having at least a partial knowledge of the event semantics, which would be the case for many IoT applications.

- *Efficiency*: a magnitude of 1,000 events/sec of throughput has been achieved by the proposed matcher.
- *Effectiveness*: over than 95%  $F_1$ Measure of matching quality has been achieved by the matcher.

*Hypothesis H2: Free tagging of events and subscriptions does not add to the cost of defining and maintaining rules with respect to the use of terms, and the cost of building and agreeing on an event semantic model required by subsymbolic event semantics; and at the same time it can achieve timely event matching with high true positives and negatives more than event processing based on non-tagged subsymbolic event semantics.* This hypothesis has been validated through the investigation of the proposed thematic matching model in Chapter 6 due to the following results:

- *Loose semantic coupling*: a lightweight amount of tags to describe events, around 2 – 7, and subscriptions, around 2 – 15 have been required which does not add any semantic coupling.
- *Efficiency*: a magnitude of 800 events/sec of throughput has been achieved by the thematic matcher. The thematic approach outperforms the non-thematic matcher for more than 92% of the sub-experiments, with throughput of 202 – 838 and an average of 320 versus 202 events/sec.
- *Effectiveness*: 85%  $F_1$ Measure of matching quality has been achieved by the thematic matcher as opposed to 62% for non-thematic processing, in the worst case of full approximation.

*Hypothesis H3: Dynamic native event enrichment decreases the cost needed to define and maintain the context parts of rules, and to agree on contextual data that is needed in events more than dedicated enrichers; and at the same time it can achieve high precision integration of event context with high completeness of events comparable to that of event processing based on dedicated enrichers.* This hypothesis has been validated through the

investigation of the proposed dynamic native enrichment model in Chapter 7 due to the following results:

- *Loose pragmatic coupling*: four high-level clauses were added to the subscriptions to guide the enricher which does not induce a coupling to contextual sources as assumed by dedicated enrichers or fusion-based queries which require a granular specification of the contextual data.
- *Loose pragmatic coupling*: four high-level clauses were added to the subscriptions to guide the enricher which does not induce a coupling to contextual sources as assumed by dedicated enrichers or fusion-based queries which require a granular specification of the contextual data.
- *Efficiency and effectiveness*: up to 44% F<sub>5</sub> Measure of enrichment precision and completeness have been achieved, 7 times more than other instantiations of the enrichment model on average.

*Hypothesis H4: Approximate event processing can operate in event environments with low-cost agreements on event semantics and pragmatics more than exact event processing; and at the same time achieve timely event matching with high true positives and negatives, and high precision integration of event context with high completeness of events, comparable to that of event processing based on exact models.* This hypothesis has been validated through the investigation of the previous models as approximation complements these models to work properly in loosely coupled environments.

### 9.3 Contributions

The *Core Contributions* of this thesis are as follows:

#### ***A Problem Analysis Framework based on Communication Models, Crossing System Boundaries, and Decoupling***

A new analytical framework of distributed, open, and heterogeneous event systems has been used in this thesis. The problem of the effect of semantic and contextual agreements on scalability has been approached through abstracting event-based systems using a communication model and a model of crossing system boundaries. For event systems

to achieve interoperability, they need to address full event-based communication rather than transmission. The analysis found that this can happen when events are seen as boundary objects that can cross syntactic, semantic, and pragmatic boundaries. A trade-off is found between crossing these boundaries and the decoupling that is assumed on each type of the boundaries. The thesis used this analysis to address this trade-off through loosening semantic and pragmatic coupling, and embedding more subsymbolic semantics and contexts into events and subscription rules.

This analysis has been presented in the ACM international Conference on Distributed Event-Based Systems (DEBS 2015) [47].

### ***An Approximate Semantic Event Matching Model***

An effective and efficient approximate event processing model to address semantic coupling in heterogeneous event environments, such as the Internet of Things, has been proposed. The model employed two main elements: subsymbolic statistical event semantics, and approximation. The model uses a formal framework for semantic event matching based on an ensemble of semantic, top-1, and top- $k$  matchers. A probabilistic model for uncertainty management has been used where the result of matching is a probability score that reflects the uncertainty about a particular mapping between an event and a subscription.

An efficient algorithm to find top- $k$  matchings based on an evolving Pareto frontier in a vector space has been proposed. The overall time complexity of the proposed algorithm is proportional to  $O(n.m.\log(m) + n.\log(n) + k.n^2 + k.\log(k))$ , i.e. polynomial and approximately linear with  $k$  and the number of event's tuples  $m$  while it is polynomial and approximately quadratic with the number of subscription's predicates  $n$ .

This model has been presented in the ACM Transactions on Internet Technology Journal (ToIT 2014) [153], and the ACM International Conference on Distributed Event-Based Systems (DEBS 2012) [155].

### ***Thingsonomies and Thematic Matching***

A new model has been proposed for improving the semantic content of events and subscriptions without entailing further semantic agreements. The model is based on free tags, or thingsonomies, which are added to events and subscriptions and used by the approximate matcher to parametrize the semantic model and get better approximations

of events and subscriptions meaning. An effective and efficient thematic event processing model based on free tagging and thingsonomies has been developed and proved superior to its non-thematic counterpart.

This model has been presented in the IEEE Internet Computing (2015) [151], and presented in the International ACM/IFIP/USENIX Middleware Conference (Middleware 2014) [152].

### ***A Dynamic Native Event Enrichment Model***

A new model for tackling event contextual content has been proposed based on altering the assumption on events from a closed world to an open world. Under the open world assumption, events are considered incomplete; and a unified and native model of event enrichment has been proposed to complement events based on subscriptions on the fly in a loosely pragmatically coupled manner. The model developed a new formalism based on set algebra and information incompleteness, and an instantiation based on spreading activation in Linked Data.

This model has been presented in the ACM International Conference on Distributed Event-Based Systems (DEBS 2013) [154], and the the International Workshop on Semantic Sensor Networks (SSN 2011) at the International Semantic Web Conference (ISWC 2011)[156].

*Additional Contributions* of this thesis are as follows:

### ***Literature Review and Gap Analysis***

Related work to the problem of interoperability in event processing systems have been investigated and projected against the newly defined requirements of loose semantic and pragmatic coupling. This novel analysis revealed a gap in the literature in terms of using approaches, such as symbolic ontologies, to address interoperability but which can also add coupling that limits scalability. The analysis showed that current event systems depend on exact models, top-down symbolic semantics, and dedicated enrichers. That helped define the directions of the hypotheses and proposed models of this thesis.

Some parts of the related work analysis in have been presented to various degrees in the IEEE Internet Computing (2015) [151], the International ACM/IFIP/USENIX Middleware Conference (Middleware 2014) [152], the ACM Transactions on Internet Technology

Journal (ToIT 2014) [153], the ACM International Conference on Distributed Event-Based Systems (DEBS 2015) [47], DEBS 2013 [154], DEBS 2012 [155], and the the International Workshop on Semantic Sensor Networks (SSN 2011) at the International Semantic Web Conference (ISWC 2011)[156].

### ***A Synthetic Evaluation Framework for Event Matching and Enrichment***

A new evaluation framework based on synthetic event loads and approximate subscriptions from real world IoT deployments have been proposed. The framework featured a set of metrics for evaluating event-based systems based on precision, recall, F-measures, and information completeness. The framework tackled the challenge of large-scale ground truth generation via starting with a small ground truth and updating it systematically while expanding events to create a large-scale heterogeneous environment for evaluation.

This framework has been presented in the ACM Transactions on Internet Technology Journal (ToIT 2014) [153], and the ACM International Conference on Distributed Event-Based Systems (DEBS 2013) [154], and DEBS 2012 [155].

## **9.4 Limitations**

The research conducted in this thesis has the following main limitations:

- The approximate event matching model is not suitable in critical scenarios such as security settings or time-critical use cases. The use of semantic relatedness services instead of exact string comparison is costly from a time performance perspective. Thus, applications with hard real-time deadlines may not be the ideal applications. Besides, the relatively lower accuracy of the approximate model compared to an exact model may incur false negatives, which can have serious implications in critical infrastructures. It could be better to afford the cost of establishing semantic and contextual agreements and use an exact event engine rather than leaving semantic approximation to the matcher.
- The subsymbolic distributional event semantic model is limited to short lexical compositions of terms such as one or two words, and may not be able to represent complex syntactic structures which can be the case of natural language events and

subscriptions. The extension of geometrical semantic models to complex syntax is still an area of open research called compositional distributional semantics.

- The approach is limited by the limitations of the underlying semantic models. A semantic model that is more suitable to carry out thematic projection is more effective for decoding symbols into geometrical representations. Advancement in the areas of semantic encoding and decoding can push the limits of the proposed approach, specifically the mechanisms concerned with using tags to parametrize the vector space semantic model.
- The proposed approach is limited to the cases where contextual data is automatically accessible and discoverable from where the enrichment-enabled engine resides. Considerations of networking and accessibility issues can limit the extent to which the enricher can perform its task. Security and privacy considerations can also be restrictive to what data can be used to enrich an event.
- This study has been limited by practical considerations of the size of the experimental parameter space which is of a high dimensionality in some experiments. Particularly in the thematic matching model, the size of the parameters space is (the number of events  $\times$  the number of subscriptions  $\times$  the number of event themes  $\times$  the number of subscription themes  $\times$  the sample size). This poses practical constraints on conducting the experiments on the available machines, and any future work shall use more resources to allow experimentation with higher parameters.

## 9.5 Future Work

The main future directions of this work are as follows:

### *Approximate Semantic Event Matching*

One future work of interest is the extension of the proposed approximate semantic event matcher to other data models of events and subscriptions beyond attribute-value models to encapsulate more types of scenarios into the approach. Such an extension may have impacts on the design of the first line matchers, on the definition of mapping functions, and on the one-to-one and one-to-many constraints of an event tuple versus a subscription's predicate.

Another prospective area is the matcher extensibility to other operators such as Boolean and numeric operators, e.g.  $!$ ,  $=$ ,  $<$ ,  $\leq$ ,  $>$ , and  $\geq$ , to improve the expressiveness of the matching language. Besides, an interesting direction is the investigation of parallelization, optimization and indexing techniques for approximate uncertain matching which can reduce the time delay caused by semantic relatedness computation.

### ***Exact Event Matching***

A future work is the comprehensive study of the relationship between approximate matching and exact matching in partially approximate subscriptions and rules. That can dictate the design of a seamless integration between approximate matching and exact matching and the development of a matcher that can transparently perform better in cases where approximate matching or exact matching is preferred. Exact matching parts of rules may be outsourced into another exact matching-based event engine which is optimized for such scenarios, with the ability of the matcher to combine efficiently exact matching results with approximate results to produce the final matching results.

### ***Complex Event Processing***

A future work of interest is the study of suitable uncertainty models and uncertainty reasoning models to support complex event processing over approximate single event matching and approximate enrichment. Such a direction would need to consider the statistical monotonicity in single event matching and proper methods to propagate top- $k$  probability spaces into pattern matching. It also requires the study of efficient reasoning models including sampling over uncertain matching and other pattern-level approximation models.

### ***Thingsonomies and Thematic Matching***

The thematic event matching model is generic using the idea of free tagging. It can be developed by considering unstructured models of events such as images, video streams, and voice events. The interplay between tagging and unstructured events is an area of future research where models to use tags to improve matching over such events shall be investigated.

### ***Dynamic Native Event Enrichment***

Another area for future research includes the instantiation of dynamic native enrichment for non-Linked Data events and background knowledge and the study of the effect of such other instantiations on the model. That can help improve the applicability of the model into various scenarios, and help develop the model and enrichment clauses according to the investigation of different instantiations. Besides, the dynamic native event enrichment model could be developed to extend into other dimensions of incompleteness such as temporal segmentation. Caching and its effect on the performance of enrichment are also an area of interest for future studies regarding event enrichment.

### ***Semantic Models***

The investigation of the effect of various subsymbolic and non-symbolic event semantic models is an area for future research. Such a direction requires the use of latent semantic indexing, as well as models based on neural networks for semantics. The effect of such models on the efficiency and effectiveness of the matcher shall be considered along with the suitability of these models for parallelization, optimization, and interpretability.

### ***Evaluation Framework***

Future work shall consider the experimentation of the proposed models with larger parameters to investigate more dimensions of the performance. Such a direction requires the consideration of the use of parallelization and powerful computing resources such as cloud-based evaluation to enable experimentation with very large parameters. Evaluation can also extend synthetic methods with real-world crowd behaviour with respect to using tags and thingsonomies to annotate events and subscriptions.

# Bibliography

- [1] Organization for Economic Co-operation and Development (OECD). Machine-to-Machine Communications: Connecting Billions of Devices. 2012.
- [2] Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. Smart cities in europe. *Journal of urban technology*, 18(2):65–82, 2011.
- [3] Thilo Sauter and Maksim Lobashov. End-to-end communication architecture for smart grids. *Industrial Electronics, IEEE Transactions on*, 58(4):1218–1228, 2011.
- [4] Deborah Snoonian. Smart buildings. *Spectrum, IEEE*, 40(8):18–23, 2003.
- [5] Jan Kleissl and Yuvraj Agarwal. Cyber-physical energy systems: Focus on smart buildings. In *Proceedings of the 47th Design Automation Conference, DAC '10*, pages 749–754, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0002-5. doi: 10.1145/1837274.1837464. URL <http://doi.acm.org/10.1145/1837274.1837464>.
- [6] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.
- [7] Charu C Aggarwal, Naveen Ashish, and Amit Sheth. The internet of things: A survey from the data-centric perspective. In *Managing and mining sensor data*, pages 383–428. Springer, 2013.
- [8] Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, 44(3):15:1–15:62, June 2012. ISSN 0360-0300.
- [9] Annika Hinze, Kai Sachs, and Alejandro Buchmann. Event-based applications and enabling technologies. In *Proceedings of the Third ACM International Conference*

- on *Distributed Event-Based Systems*, DEBS '09, pages 1:1–1:15, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-665-6. doi: 10.1145/1619258.1619260. URL <http://doi.acm.org/10.1145/1619258.1619260>.
- [10] O Etzion and P Niblett. *Event Processing in Action*. pages 143–175, 2010.
- [11] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35(2):114–131, 2003.
- [12] Gero Mühl, Ludger Fiege, and Peter Pietzuch. *Distributed event-based systems*, volume 1. Springer, 2006.
- [13] Norman W. Paton and Oscar Díaz. Active database systems. *ACM Comput. Surv.*, 31(1):63–103, March 1999. ISSN 0360-0300. doi: 10.1145/311531.311623. URL <http://doi.acm.org/10.1145/311531.311623>.
- [14] Inc. Sun Microsystems. *Java platform, enterprise edition(Java EE) specification, v1.2*. Sun Microsystems, Inc., 1999.
- [15] Object Management Group. *The Common Object Request Broker (CORBA): Architecture and Specification*. Object Management Group, 1995.
- [16] B. H. Tay and A. L. Ananda. A survey of remote procedure calls. *SIGOPS Oper. Syst. Rev.*, 24(3):68–79, July 1990. ISSN 0163-5980. doi: 10.1145/382244.382832. URL <http://doi.acm.org/10.1145/382244.382832>.
- [17] David Garlan and David Notkin. Formalizing design spaces: Implicit invocation mechanisms. In *VDM'91 Formal Software Development Methods*, pages 31–44. Springer, 1991.
- [18] Gregor Hohpe and Bobby Woolf. *Enterprise integration patterns: Designing, building, and deploying messaging solutions*. Addison-Wesley Professional, 2004.
- [19] Mariano Cilia, Michael Haupt, Mira Mezini, and Alejandro Buchmann. The convergence of aop and active databases: Towards reactive middleware. In *Generative Programming and Component Engineering*, pages 169–188. Springer, 2003.
- [20] Adele Goldberg and David Robson. *Smalltalk-80: The Language and Its Implementation*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1983. ISBN 0-201-11371-6.

- [21] Event Processing Technical Society. Event Processing Glossary – Version 2.0, July 2011, Last accessed January 2015. URL [http://www.complexevents.com/wp-content/uploads/2011/08/EPTS\\_Event\\_Processing\\_Glossary\\_v2.pdf](http://www.complexevents.com/wp-content/uploads/2011/08/EPTS_Event_Processing_Glossary_v2.pdf).
- [22] Antonio Carzaniga, David S Rosenblum, and Alexander L Wolf. Achieving scalability and expressiveness in an internet-scale event notification service. In *Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing*, pages 219–227. ACM, 2000.
- [23] Patrick Th Eugster, Rachid Guerraoui, and Christian Heide Damm. On objects and events. In *ACM SIGPLAN Notices*, volume 36, pages 254–269. ACM, 2001.
- [24] Ludger Fiege, Mariano Cilia, Gero Muhl, and Alejandro Buchmann. Publish-subscribe grows up: support for management, visibility control, and heterogeneity. *Internet Computing, IEEE*, 10(1):48–55, 2006.
- [25] Milenko Petrovic, Ioana Burcea, and Hans-Arno Jacobsen. S-topss: semantic toronto publish/subscribe system. In *Proceedings of the 29th international conference on Very large data bases - Volume 29, VLDB '03*, pages 1101–1104. VLDB Endowment, 2003. ISBN 0-12-722442-4.
- [26] Jinling Wang, Beihong Jin, and Jing Li. An ontology-based publish/subscribe system. In *Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware*, Middleware '04, pages 232–253, New York, NY, USA, 2004. Springer-Verlag New York, Inc. ISBN 3-540-23428-4.
- [27] Liangzhao Zeng and Hui Lei. A semantic publish/subscribe system. In *E-Commerce Technology for Dynamic E-Business, 2004. IEEE International Conference on*, pages 32–39, Sept 2004.
- [28] Gordon S Blair, Amel Bennaceur, Nikolaos Georgantas, Paul Grace, Valérie Isarny, Vatsala Nundloll, and Massimo Paolucci. The role of ontologies in emergent middleware: Supporting interoperability in complex distributed systems. In *Middleware 2011*, pages 410–430. Springer, 2011.
- [29] Weiwei Zhang, Jiangang Ma, and Dan Ye. Fomatch: A fuzzy ontology-based semantic matching algorithm of publish/subscribe systems. In *Proc. CIMCA*, pages 111–117. IEEE, 2008.

- [30] Haifeng Liu and H-A Jacobsen. Modeling uncertainties in publish/subscribe systems. In *Data Engineering, 2004. Proceedings. 20th International Conference on*, pages 510–521. IEEE, 2004.
- [31] Haifeng Liu and Hans-Arno Jacobsen. A-topss: A publish/subscribe system supporting imperfect information processing. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 1281–1284. VLDB Endowment, 2004.
- [32] M Drosou, K Stefanidis, and E Pitoura. Preference-aware publish/subscribe delivery with diversity. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, pages 6:1—6:12. ACM, 2009.
- [33] Segev Wasserkrug, Avigdor Gal, Opher Etzion, and Yulia Turchin. Efficient processing of uncertain events in rule-based systems. *Knowledge and Data Engineering, IEEE Transactions on*, 24(1):45–58, 2012.
- [34] Björn Schilling, Boris Koldehofe, Udo Pletat, and Kurt Rothermel. Distributed heterogeneous event processing: Enhancing scalability and interoperability of cep in an industrial context. In *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, DEBS '10*, pages 150–159, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-927-5. doi: 10.1145/1827418.1827453. URL <http://doi.acm.org/10.1145/1827418.1827453>.
- [35] Arvind Arasu, Shivnath Babu, and Jennifer Widom. The cql continuous query language: Semantic foundations and query execution. *The VLDB Journal*, 15(2):121–142, June 2006. ISSN 1066-8888. doi: 10.1007/s00778-004-0147-z. URL <http://dx.doi.org/10.1007/s00778-004-0147-z>.
- [36] Kia Teymourian, Malte Rohde, and Adrian Paschke. Fusion of background knowledge and streams of events. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems, DEBS '12*, pages 302–313, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1315-5. doi: 10.1145/2335484.2335517. URL <http://doi.acm.org/10.1145/2335484.2335517>.
- [37] D Le-Phuoc, M Dao-Tran, J Xavier Parreira, and M Hauswirth. A native and adaptive approach for unified processing of linked streams and linked data. *The Semantic Web-ISWC 2011*, pages 370–388, 2011.

- [38] Darko Anicic, Paul Fodor, Sebastian Rudolph, and Nenad Stojanovic. Ep-sparql: A unified language for event processing and stream reasoning. In *Proceedings of the 20th International Conference on World Wide Web, WWW '11*, pages 635–644, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963495. URL <http://doi.acm.org/10.1145/1963405.1963495>.
- [39] Tobias Freudenreich, Stefan Appel, Sebastian Frischbier, and Alejandro P Buchmann. Actress: automatic context transformation in event-based software systems. In *DEBS*, pages 179–190, 2012.
- [40] M Cilia, M Antollini, C Bornhvd, and A Buchmann. Dealing with heterogeneous data in pub/sub systems: The concept-based approach. In *Proceedings of International Workshop on Distributed Event-Based Systems (DEBS 2004)*, Edinburgh, UK, pages 24–25. IET, 2004.
- [41] Mariano Cilia, Christof Bornhövd, and Alejandro P Buchmann. Cream: An infrastructure for distributed, heterogeneous event-based applications. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pages 482–502. Springer, 2003.
- [42] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.
- [43] Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. Survey on social tagging techniques. *ACM SIGKDD Explorations Newsletter*, 12(1):58–72, 2010.
- [44] H. V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Commun. ACM*, 57(7):86–94, July 2014. ISSN 0001-0782. doi: 10.1145/2611567. URL <http://doi.acm.org/10.1145/2611567>.
- [45] Aniket Kittur. Crowdsourcing, collaboration and creativity. *ACM Crossroads*, 17(2):22–26, 2010.
- [46] Paul R Carlile. Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organization science*, 15(5): 555–568, 2004.

- [47] Souleiman Hasan and Edward Curry. Tackling variety in event-based systems. In *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*, DEBS '15, pages 256–265, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3286-6. doi: 10.1145/2675743.2774215. URL <http://doi.acm.org/10.1145/2675743.2774215>.
- [48] EsperTech. Esper Complex Event Processing Engine, Last accessed January 2015. URL <http://esper.codehaus.org/>.
- [49] Feng Gao, Edward Curry, and Sami Bhiri. Complex event service provision and composition based on event pattern matchmaking. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, pages 71–82. ACM, 2014.
- [50] Luis Sanchez, José Antonio Galache, Veronica Gutierrez, JM Hernandez, J Bernat, Alex Gluhak, and Tomás Garcia. Smartsantander: The meeting point between future internet research and experimentation and the smart cities. In *Future Network & Mobile Summit (FutureNetw), 2011*, pages 1–8. IEEE, 2011.
- [51] Felipe Gil-Castineira, Enrique Costa-Montenegro, Francisco J Gonzalez-Castano, Cristina López-Bravo, Timo Ojala, and Raja Bose. Experiences inside the ubiquitous oulu smart city. *Computer*, 44(6):48–55, 2011.
- [52] German Telekom and City of Friedrichshafen. Friedrichshafen Smart City, 2012, Last accessed January 2015. URL [http://www.gsma.com/connectedliving/wp-content/uploads/2012/11/cl\\_tcity\\_web\\_10\\_12.pdf](http://www.gsma.com/connectedliving/wp-content/uploads/2012/11/cl_tcity_web_10_12.pdf).
- [53] Rohan Narayana Murty, Geoffrey Mainland, Ian Rose, Atanu Roy Chowdhury, Abhimanyu Gosain, Josh Bers, and Matt Welsh. Citysense: An urban-scale wireless sensor network and testbed. In *Technologies for Homeland Security, 2008 IEEE Conference on*, pages 583–588. IEEE, 2008.
- [54] METRO AG. future store, Last accessed January 2015. URL <http://www.future-store.org/>.
- [55] José María Cavanillas, Edward Curry, and Wolfgang Wahlster, editors. *New Horizons for a Data-Driven Economy*. Springer International Publishing, first edition, 2016.

- [56] Yvonne Genovese and S Prentice. Pattern-based strategy: Getting value from big data. *Gartner Special Report (June 2011)*, 2011.
- [57] International Telecommunication Union (ITU). ICT Facts and Figures, 2015, Last accessed October 2015. URL <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf>.
- [58] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.
- [59] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.
- [60] Ora Lassila and Ralph R Swick. Resource description framework (RDF) model and syntax specification. 1999.
- [61] Dan Brickley and Ramanathan V Guha. RDF vocabulary description language 1.0: RDF schema. 2004.
- [62] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.
- [63] Gong Cheng, Weiyi Ge, and Yuzhong Qu. Falcons: searching and browsing entities on the semantic web. In *Proceedings of the 17th international conference on World Wide Web*, pages 1101–1102. ACM, 2008.
- [64] Gong Cheng, Weiyi Ge, Honghan Wu, and Yuzhong Qu. Searching semantic web objects based on class hierarchies. In *LDOW*, 2008.
- [65] Alfio Ferrara, Davide Lorusso, Stefano Montanelli, and Gaia Varese. Towards a benchmark for instance matching. In *International Workshop on Ontology Matching, The 7th International Semantic Web Conference*, page 37, 2008.
- [66] National Center for Science US National Science Foundation and Engineering Statistics. National Survey of Recent College Graduates, 2010, 2010, Last accessed January 2015. URL [http://ncesdata.nsf.gov/recentgrads/2010/html/RCG2010\\_DST1\\_1.html](http://ncesdata.nsf.gov/recentgrads/2010/html/RCG2010_DST1_1.html).

- [67] US Department of Health and National Center for Health Statistics Human Services, Centers for Disease Control. Vital Statistics of the United States, 1988. Volume I, Natality, 1988, Last accessed January 2015. URL [http://www.cdc.gov/nchs/data/vsus/nat88\\_1.pdf](http://www.cdc.gov/nchs/data/vsus/nat88_1.pdf).
- [68] Umair ul Hassan, Murilo Bassora, Ali H Vahid, Sean O’Riain, and Edward Curry. A collaborative approach for metadata management for internet of things: Linking micro tasks with physical objects. In *The 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom)*, pages 593–598. IEEE, 2013.
- [69] Andrew Lih. *The Wikipedia revolution: How a bunch of nobodies created the world’s greatest encyclopedia*. Hyperion, 2009.
- [70] Joel Ross, Lilly Irani, M Silberman, Andrew Zaldivar, and Bill Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI’10 Extended Abstracts on Human Factors in Computing Systems*, pages 2863–2872. ACM, 2010.
- [71] Ovidiu Vermesan, Peter Friess, Patrick Guillemin, Harald Sundmaeker, Markus Eisenhauer, Klaus Moessner, Franck Le Gall, and Philippe Cousin. Internet of things strategic research and innovation agenda. *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*, page 7, 2013.
- [72] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.
- [73] Krysia Broda, Keith Clark, Rob Miller, and Alessandra Russo. Sage: A logical agent-based environment monitoring and control system. In Manfred Tscheligi, Boris de Ruyter, Panos Markopoulos, Reiner Wichert, Thomas Mirlacher, Alexander Meschterjakov, and Wolfgang Reitberger, editors, *Ambient Intelligence*, volume 5859 of *Lecture Notes in Computer Science*, pages 112–117. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-05407-5. doi: 10.1007/978-3-642-05408-2\_14. URL [http://dx.doi.org/10.1007/978-3-642-05408-2\\_14](http://dx.doi.org/10.1007/978-3-642-05408-2_14).
- [74] Alan Demers, Johannes Gehrke, Mingsheng Hong, Mirek Riedewald, and Walker White. Towards expressive publish/subscribe systems. In *Proceedings of the 10th*

- International Conference on Advances in Database Technology*, EDBT'06, pages 627–644, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-32960-9, 978-3-540-32960-2. doi: 10.1007/11687238\_38. URL [http://dx.doi.org/10.1007/11687238\\_38](http://dx.doi.org/10.1007/11687238_38).
- [75] Fusheng Wang and Peiya Liu. Temporal management of rfid data. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pages 1128–1139. VLDB Endowment, 2005. ISBN 1-59593-154-6. URL <http://dl.acm.org/citation.cfm?id=1083592.1083723>.
- [76] Massimo Ficco and Luigi Romano. A generic intrusion detection and diagnoser system based on complex event processing. In *Data Compression, Communications and Processing (CCP), 2011 First International Conference on*, pages 275–284. IEEE, 2011.
- [77] Thomas Heinze, Zbigniew Jerzak, Gregor Hackenbroich, and Christof Fetzer. Latency-aware elastic scaling for distributed data stream processing systems. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, pages 13–22. ACM, 2014.
- [78] Radu Tudoran, Olivier Nano, Ivo Santos, Alexandru Costan, Hakan Soncu, Luc Bougé, and Gabriel Antoniu. Jetstream: Enabling high performance event streaming across cloud data-centers. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, DEBS '14, pages 23–34, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2737-4. doi: 10.1145/2611286.2611298. URL <http://doi.acm.org/10.1145/2611286.2611298>.
- [79] Beate Ottenwälder, Boris Koldehofe, Kurt Rothermel, Kirak Hong, and Umakishore Ramachandran. Recep: Selection-based reuse for distributed complex event processing. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, DEBS '14, pages 59–70, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2737-4. doi: 10.1145/2611286.2611297. URL <http://doi.acm.org/10.1145/2611286.2611297>.
- [80] Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, and Roman Vitenberg. Distributed event aggregation for content-based publish/subscribe systems. In *Proceedings of the 8th ACM International Conference*

- on *Distributed Event-Based Systems*, DEBS '14, pages 95–106, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2737-4. doi: 10.1145/2611286.2611302. URL <http://doi.acm.org/10.1145/2611286.2611302>.
- [81] Michael Stonebraker, Uğur Çetintemel, and Stan Zdonik. The 8 requirements of real-time stream processing. *SIGMOD Rec.*, 34(4):42–47, December 2005. ISSN 0163-5808. doi: 10.1145/1107499.1107504. URL <http://doi.acm.org/10.1145/1107499.1107504>.
- [82] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772777. URL <http://doi.acm.org/10.1145/1772690.1772777>.
- [83] Kang G Shin and Parameswaran Ramanathan. Real-time computing: A new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1): 6–24, 1994.
- [84] U. Dayal, B. Blaustein, A. Buchmann, U. Chakravarthy, M. Hsu, R. Ledin, D. McCarthy, A. Rosenthal, S. Sarin, M. J. Carey, M. Livny, and R. Jauhari. The hipac project: Combining active databases and timing constraints. *SIGMOD Rec.*, 17(1):51–70, March 1988. ISSN 0163-5808. doi: 10.1145/44203.44208. URL <http://doi.acm.org/10.1145/44203.44208>.
- [85] Dennis McCarthy and Umeshwar Dayal. The architecture of an active database management system. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*, SIGMOD '89, pages 215–224, New York, NY, USA, 1989. ACM. ISBN 0-89791-317-5. doi: 10.1145/67544.66946. URL <http://doi.acm.org/10.1145/67544.66946>.
- [86] Narain H. Gehani and H. V. Jagadish. Ode as an active database: Constraints and triggers. In *Proceedings of the 17th International Conference on Very Large Data Bases*, VLDB '91, pages 327–336, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1-55860-150-3. URL <http://dl.acm.org/citation.cfm?id=645917.672167>.

- [87] Daniel F Lieuwen, Narain Gehani, and Robert Arlein. The ode active database: Trigger semantics and implementation. In *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*, pages 412–420. IEEE, 1996.
- [88] Stella Gatzui and KlausR. Dittrich. Events in an active object-oriented database system. In NormanW. Paton and M.Howard Williams, editors, *Rules in Database Systems, Workshops in Computing*, pages 23–39. Springer London, 1994. ISBN 978-3-540-19846-8. doi: 10.1007/978-1-4471-3225-7\_2. URL [http://dx.doi.org/10.1007/978-1-4471-3225-7\\_2](http://dx.doi.org/10.1007/978-1-4471-3225-7_2).
- [89] S. Chakravarthy and D. Mishra. Snoop: An expressive event specification language for active databases. *Data Knowl. Eng.*, 14(1):1–26, November 1994. ISSN 0169-023X. doi: 10.1016/0169-023X(94)90006-X. URL [http://dx.doi.org/10.1016/0169-023X\(94\)90006-X](http://dx.doi.org/10.1016/0169-023X(94)90006-X).
- [90] Inc. Sun Microsystems. *Java platform, enterprise edition(Java EE) specification, v5*. Sun Microsystems, Inc., 2006.
- [91] Timothy H. Harrison, David L. Levine, and Douglas C. Schmidt. The design and performance of a real-time corba event service. In *Proceedings of the 12th ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications, OOPSLA '97*, pages 184–200, New York, NY, USA, 1997. ACM. ISBN 0-89791-908-4. doi: 10.1145/263698.263734. URL <http://doi.acm.org/10.1145/263698.263734>.
- [92] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994.
- [93] Andrew D. Birrell and Bruce Jay Nelson. Implementing remote procedure calls. *ACM Trans. Comput. Syst.*, 2(1):39–59, February 1984. ISSN 0734-2071. doi: 10.1145/2080.357392. URL <http://doi.acm.org/10.1145/2080.357392>.
- [94] Kai Li and Paul Hudak. Memory coherence in shared virtual memory systems. *ACM Trans. Comput. Syst.*, 7(4):321–359, November 1989. ISSN 0734-2071. doi: 10.1145/75104.75105. URL <http://doi.acm.org/10.1145/75104.75105>.
- [95] Marcel Altherr, Martin Erzberger, and Silvano Maffei. ibus-a software bus middleware for the java platform. In *Proceedings of the International Workshop on Reliable Middleware Systems*, pages 43–53, 1999.

- [96] TIBCO. TIBCO Rendezvous Messaging Middleware, Last accessed August 2015. URL <http://www.tibco.com/products/automation/enterprise-messaging/rendezvous>.
- [97] David S. Rosenblum and Alexander L. Wolf. A design framework for internet-scale event observation and notification. In *Proceedings of the 6th European SOFTWARE ENGINEERING Conference Held Jointly with the 5th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ESEC '97/FSE-5, pages 344–360, New York, NY, USA, 1997. Springer-Verlag New York, Inc. ISBN 3-540-63531-9. doi: 10.1145/267895.267920. URL <http://dx.doi.org/10.1145/267895.267920>.
- [98] Bill Segall and David Arnold. Elvin has left the building: A publish/subscribe notification service with quenching. In *Proceedings of AUUG97*, pages 3–5. Brisbane, Australia, 1997.
- [99] Gianpaolo Cugola, Elisabetta Di Nitto, and Alfonso Fuggetta. The jedi event-based infrastructure and its application to the development of the opss wfms. *Software Engineering, IEEE Transactions on*, 27(9):827–850, 2001.
- [100] Mark Hapner, Rich Burrige, Rahul Sharma, Joseph Fialli, and Kate Stout. Java message service. *Sun Microsystems Inc., Santa Clara, CA*, 2002.
- [101] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *Communications magazine, IEEE*, 40(8):102–114, 2002.
- [102] Shivnath Babu and Jennifer Widom. Continuous queries over data streams. *SIGMOD Rec.*, 30(3):109–120, September 2001. ISSN 0163-5808. doi: 10.1145/603867.603884. URL <http://doi.acm.org/10.1145/603867.603884>.
- [103] Daniel J. Abadi, Don Carney, Ugur Çetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Aurora: A new model and architecture for data stream management. *The VLDB Journal*, 12(2):120–139, August 2003. ISSN 1066-8888. doi: 10.1007/s00778-003-0095-z. URL <http://dx.doi.org/10.1007/s00778-003-0095-z>.
- [104] Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong, Sailesh Krishnamurthy, Samuel R. Madden,

- Fred Reiss, and Mehul A. Shah. Telegraphcq: Continuous dataflow processing. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pages 668–668, New York, NY, USA, 2003. ACM. ISBN 1-58113-634-X. doi: 10.1145/872757.872857. URL <http://doi.acm.org/10.1145/872757.872857>.
- [105] Jianjun Chen, David J. DeWitt, Feng Tian, and Yuan Wang. Niagaraqc: A scalable continuous query system for internet databases. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 379–390, New York, NY, USA, 2000. ACM. ISBN 1-58113-217-4. doi: 10.1145/342009.335432. URL <http://doi.acm.org/10.1145/342009.335432>.
- [106] Ling Liu and C. Pu. A dynamic query scheduling framework for distributed and evolving information systems. In *Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on*, pages 474–481, May 1997. doi: 10.1109/ICDCS.1997.603389.
- [107] Mark Sullivan and Andrew Heybey. Tribeca: A system for managing large databases of network traffic. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ATEC '98, pages 2–2, Berkeley, CA, USA, 1998. USENIX Association. URL <http://dl.acm.org/citation.cfm?id=1268256.1268258>.
- [108] Chuck Cranor, Yuan Gao, Theodore Johnson, Vlaidslav Shkapenyuk, and Oliver Spatscheck. Gigascope: High performance network monitoring with an sql interface. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD '02, pages 623–623, New York, NY, USA, 2002. ACM. ISBN 1-58113-497-5. doi: 10.1145/564691.564777. URL <http://doi.acm.org/10.1145/564691.564777>.
- [109] Yijian Bai, Hetal Thakkar, Haixun Wang, Chang Luo, and Carlo Zaniolo. A data stream language and system designed for power and extensibility. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 337–346, New York, NY, USA, 2006. ACM. ISBN 1-59593-433-2. doi: 10.1145/1183614.1183664. URL <http://doi.acm.org/10.1145/1183614.1183664>.

- [110] Sybase. Coral8 programmer's guide, Last accessed January 2015. URL <http://infocenter.sybase.com/archive/topic/com.sybase.infocenter.dc01029.0200/pdf/cep-ProgrammersGuide.pdf>.
- [111] StreamBase. StreamSQL guide, Last accessed January 2015. URL <http://streambase.com/developers/docs/latest/streamsql/index.html>.
- [112] Lisa Amini, Navendu Jain, Anshul Sehgal, Jeremy Silber, and Olivier Verscheure. Adaptive control of extreme-scale stream processing systems. In *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on*, pages 71–71. IEEE, 2006.
- [113] Marcos K. Aguilera, Robert E. Strom, Daniel C. Sturman, Mark Astley, and Tushar D. Chandra. Matching events in a content-based subscription system. In *Proceedings of the Eighteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '99, pages 53–61, New York, NY, USA, 1999. ACM. ISBN 1-58113-099-6. doi: 10.1145/301308.301326. URL <http://doi.acm.org/10.1145/301308.301326>.
- [114] Guoli Li and Hans-Arno Jacobsen. Composite subscriptions in content-based publish/subscribe systems. In *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*, Middleware '05, pages 249–269, New York, NY, USA, 2005. Springer-Verlag New York, Inc. URL <http://dl.acm.org/citation.cfm?id=1515890.1515903>.
- [115] David C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001. ISBN 0201727897.
- [116] Asaf Adi and Opher Etzion. Amit - the situation manager. *The VLDB Journal*, 13(2):177–203, May 2004. ISSN 1066-8888. doi: 10.1007/s00778-003-0108-y. URL <http://dx.doi.org/10.1007/s00778-003-0108-y>.
- [117] David C. Luckham and James Vera. An event-based architecture definition language. *Software Engineering, IEEE Transactions on*, 21(9):717–734, 1995.
- [118] Masoud Mansouri-Samani and Morris Sloman. Monitoring distributed systems. *Network, IEEE*, 7(6):20–30, 1993.

- [119] Peter R. Pietzuch, Brian Shand, and Jean Bacon. A framework for event composition in distributed systems. In *Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware*, Middleware '03, pages 62–82, New York, NY, USA, 2003. Springer-Verlag New York, Inc. ISBN 3-540-40317-5. URL <http://dl.acm.org/citation.cfm?id=1515915.1515921>.
- [120] Lars Brenna, Alan Demers, Johannes Gehrke, Mingsheng Hong, Joel Ossher, Biswanath Panda, Mirek Riedewald, Mohit Thatte, and Walker White. Cayuga: A high-performance event processing engine. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 1100–1102, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-686-8. doi: 10.1145/1247480.1247620. URL <http://doi.acm.org/10.1145/1247480.1247620>.
- [121] Nicholas Poul Schultz-Møller, Matteo Migliavacca, and Peter Pietzuch. Distributed complex event processing with query rewriting. In *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, DEBS '09, pages 4:1–4:12, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-665-6. doi: 10.1145/1619258.1619264. URL <http://doi.acm.org/10.1145/1619258.1619264>.
- [122] Mert Akdere, Uğur Çetintemel, and Nesime Tatbul. Plan-based complex event detection across distributed sources. *Proc. VLDB Endow.*, 1(1):66–77, August 2008. ISSN 2150-8097. doi: 10.14778/1453856.1453869. URL <http://dx.doi.org/10.14778/1453856.1453869>.
- [123] Gianpaolo Cugola and Alessandro Margara. Raced: An adaptive middleware for complex event detection. In *Proceedings of the 8th International Workshop on Adaptive and Reflective Middleware*, ARM '09, pages 5:1–5:6, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-850-6. doi: 10.1145/1658185.1658190. URL <http://doi.acm.org/10.1145/1658185.1658190>.
- [124] Eugene Wu, Yanlei Diao, and Shariq Rizvi. High-performance complex event processing over streams. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 407–418, New York, NY, USA, 2006. ACM. ISBN 1-59593-434-0. doi: 10.1145/1142473.1142520. URL <http://doi.acm.org/10.1145/1142473.1142520>.

- [125] Daniel Gyllstrom, Jagrati Agrawal, Yanlei Diao, and Neil Immerman. On supporting kleene closure over event streams. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, ICDE '08*, pages 1391–1393, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-1-4244-1836-7. doi: 10.1109/ICDE.2008.4497566. URL <http://dx.doi.org/10.1109/ICDE.2008.4497566>.
- [126] Nodira Khoussainova, Magdalena Balazinska, and Dan Suciu. Probabilistic event extraction from rfid data. In *ICDE*, volume 8, pages 1480–1482, 2008.
- [127] Gianpaolo Cugola and Alessandro Margara. Tesla: A formally defined event specification language. In *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, DEBS '10*, pages 50–61, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-927-5. doi: 10.1145/1827418.1827427. URL <http://doi.acm.org/10.1145/1827418.1827427>.
- [128] Gianpaolo Cugola and Alessandro Margara. Complex event processing with t-rex. *J. Syst. Softw.*, 85(8):1709–1728, August 2012. ISSN 0164-1212. doi: 10.1016/j.jss.2012.03.056. URL <http://dx.doi.org/10.1016/j.jss.2012.03.056>.
- [129] SAP. SAP event stream processor, Last accessed January 2015. URL <http://www.sap.com/pc/tech/database/software/sybase-complex-event-processing/index.html>.
- [130] Oracle. Oracle event processing, Last accessed January 2015. URL <http://www.oracle.com/technetwork/middleware/complex-event-processing/overview/index.html>.
- [131] TIBCO. TIBCO BusinessEvents, Last accessed January 2015. URL <http://www.tibco.com/products/event-processing/complex-event-processing/businessevents>.
- [132] IBM. WebSphere Business Events, Last accessed January 2015. URL <http://www.ibm.com/software/integration/wbe/>.

- [133] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, pages 1–16, New York, NY, USA, 2002. ACM. ISBN 1-58113-507-6. doi: 10.1145/543613.543615. URL <http://doi.acm.org/10.1145/543613.543615>.
- [134] Lukasz Golab and M Tamer Özsu. Issues in data stream management. *ACM Sigmod Record*, 32(2):5–14, 2003.
- [135] Merriam-Webster's. Merriam-Webster Online Dictionary, Last accessed January 2015. URL <http://www.merriam-webster.com/dictionary/>.
- [136] Oxford. Oxford British and World English Online Dictionary, Last accessed January 2015. URL <http://www.oxforddictionaries.com/>.
- [137] Peter C. Bates. Debugging heterogeneous distributed systems using event-based models of behavior. *ACM Trans. Comput. Syst.*, 13(1):1–31, February 1995. ISSN 0734-2071. doi: 10.1145/200912.200913. URL <http://doi.acm.org/10.1145/200912.200913>.
- [138] Fabio Kon, Fabio Costa, Gordon Blair, and Roy H. Campbell. The case for reflective middleware. *Commun. ACM*, 45(6):33–38, June 2002. ISSN 0001-0782. doi: 10.1145/508448.508470. URL <http://doi.acm.org/10.1145/508448.508470>.
- [139] Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004.
- [140] Mariano Cilia, Christof Bornhövd, and Alejandro P Buchmann. Moving active functionality from centralized to open distributed heterogeneous environments. In *Cooperative Information Systems*, pages 195–211. Springer, 2001.
- [141] Ludwig Von Bertalanffy. General system theory. *General systems*, 1(1), 1956.
- [142] Claude Shannon and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
- [143] Daniel Chandler. *Semiotics: the basics*. Routledge, 2007.
- [144] Paul R Carlile. A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization science*, 13(4):442–455, 2002.

- [145] Michael J Reddy. The conduit metaphor: A case of frame conflict in our language about language. *Metaphor and thought*, 2:164–201, 1979.
- [146] Charles Sanders Peirce and Kenneth Laine Ketner. *Reasoning and the logic of things: the Cambridge conferences lectures of 1898*. Harvard University Press, 1992.
- [147] Susan Leigh Star and James R Griesemer. Institutional ecology, translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. *Social studies of science*, 19(3):387–420, 1989.
- [148] André B. Bondi. Characteristics of scalability and their impact on performance. In *Proceedings of the 2Nd International Workshop on Software and Performance*, WOSP '00, pages 195–203, New York, NY, USA, 2000. ACM. ISBN 1-58113-195-X. doi: 10.1145/350391.350432. URL <http://doi.acm.org/10.1145/350391.350432>.
- [149] Chi Zhang, Arvind Krishnamurthy, Randolph Y. Wang, and Jaswinder Pal Singh. Combining flexibility and scalability in a peer-to-peer publish/subscribe system. In *Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*, Middleware '05, pages 102–123, New York, NY, USA, 2005. Springer-Verlag New York, Inc. URL <http://dl.acm.org/citation.cfm?id=1515890.1515896>.
- [150] Ayelet Biger, Opher Etzion, and Yuri Rabinovich. Stratified implementation of event processing network. *Fast abstract on DEBS*, 2008.
- [151] Souleiman Hasan and Edward Curry. Thingsonomy: Tackling variety in internet of things events. *Internet Computing, IEEE*, 19(2):10–18, 2015.
- [152] Souleiman Hasan and Edward Curry. Thematic event processing. In *Proceedings of the 15th International Middleware Conference*, pages 109–120. ACM, 2014.
- [153] Souleiman Hasan and Edward Curry. Approximate semantic matching of events for the internet of things. *ACM Trans. Internet Technol.*, 14(1):2:1–2:23, August 2014. ISSN 1533-5399. doi: 10.1145/2633684. URL <http://doi.acm.org/10.1145/2633684>.
- [154] Souleiman Hasan, Sean O'Riain, and Edward Curry. Towards unified and native enrichment in event processing systems. In *Proc. The 7th ACM international*

- conference on Distributed event-based systems*, DEBS '13, pages 171–182, 2013. ISBN 978-1-4503-1758-0.
- [155] Souleiman Hasan, Sean O’Riain, and Edward Curry. Approximate semantic matching of heterogeneous events. In *Proc. The 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 252–263, 2012. ISBN 978-1-4503-1315-5.
- [156] Souleiman Hasan, Edward Curry, Mauricio Banduk, and Seán O’Riain. Toward situation awareness for the semantic sensor web: Complex event processing with dynamic linked data enrichment. *SSN*, 839:69–81, 2011.
- [157] Hannes Obweger, Josef Schiefer, Martin Suntinger, Peter Kepplinger, and Szabolcs Rozsnyai. User-oriented rule management for event-based applications. In *Proceedings of the 5th ACM international conference on Distributed event-based system*, pages 39–48. ACM, 2011.
- [158] Antonio Carzaniga. *Architectures for an event notification service scalable to wide-area networks*. PhD thesis, Politecnico di Milano, 1998.
- [159] Patrick Th. Eugster and Rachid Guerraoui. Content-based publish/subscribe with structural reflection. In *Proceedings of the 6th Conference on USENIX Conference on Object-Oriented Technologies and Systems - Volume 6*, COOTS'01, pages 10–10, Berkeley, CA, USA, 2001. USENIX Association. URL <http://dl.acm.org/citation.cfm?id=1268241.1268251>.
- [160] Gero Muhl, Ludger Fiege, Felix C Gartner, and Alejandro Buchmann. Evaluating advanced routing algorithms for content-based publish/subscribe systems. In *Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on*, pages 167–176. IEEE, 2002.
- [161] Ludger Fiege. *Visibility in Event-Based Systems*. PhD thesis, TU Darmstadt, 2005.
- [162] Dan Connolly, Frank Van Harmelen, Ian Horrocks, Deborah L McGuinness, Peter F Patel-Schneider, and Lynn Andrea Stein. Daml+ oil (march 2001) reference description. 2001.

- [163] Bruno T Messmer and Horst Bunke. Efficient subgraph isomorphism detection: a decomposition approach. *Knowledge and Data Engineering, IEEE Transactions on*, 12(2):307–323, 2000.
- [164] Segev Wasserkrug, Avigdor Gal, Opher Etzion, and Yulia Turchin. Complex event processing over uncertain data. In *Proc. DEBS '08*, pages 253–264, 2008. ISBN 978-1-60558-090-6.
- [165] Shivnath Babu, Kamesh Munagala, Jennifer Widom, and Rajeev Motwani. Adaptive caching for continuous queries. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 118–129. IEEE, 2005.
- [166] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, and Michael Grossniklaus. An execution environment for c-sparql queries. In *Proceedings of the 13th International Conference on Extending Database Technology, EDBT '10*, pages 441–452, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-945-9. doi: 10.1145/1739041.1739095. URL <http://doi.acm.org/10.1145/1739041.1739095>.
- [167] Darko Anicic, Paul Fodor, Sebastian Rudolph, Roland Stühmer, Nenad Stojanovic, and Rudi Studer. Etalis: Rule-based reasoning in event processing. In *Reasoning in Event-Based Distributed Systems*, pages 99–124. Springer, 2011.
- [168] Bruce Snyder, Dejan Bosnanac, and Rob Davies. *ActiveMQ in action*. Manning, 2011.
- [169] Erhard Rahm and Philip A Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350, 2001.
- [170] David A Maluf and Peter B Tran. Netmark: A schema-less extension for relational databases for managing semi-structured data dynamically. In *Foundations of Intelligent Systems*, pages 231–241. Springer, 2003.
- [171] Alon Y Halevy, Naveen Ashish, Dina Bitton, Michael Carey, Denise Draper, Jeff Pollock, Arnon Rosenthal, and Vishal Sikka. Enterprise information integration: successes, challenges and controversies. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 778–787. ACM, 2005.

- [172] Avigdor Gal. Managing uncertainty in schema matching with top-k schema mappings. In Stefano Spaccapietra, Karl Aberer, and Philippe Cudré-Mauroux, editors, *Journal on Data Semantics VI*, volume 4090 of *Lecture Notes in Computer Science*, pages 90–114. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-36712-3.
- [173] Avigdor Gal. Uncertain schema matching. *Synthesis Lectures on Data Management*, 3(1):1–97, 2011.
- [174] André Freitas, Joao Gabriel Oliveira, Seán O’Riain, Edward Curry, and João Carlos Pereira Da Silva. Querying linked data using semantic relatedness: a vocabulary independent approach. In *Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 40–51. Springer, 2011.
- [175] Danilo Carvalho, Cagatay Calli, André Freitas, and Edward Curry. EasyESA: A low-effort infrastructure for explicit semantic analysis (demonstration paper). In *Proceedings of the 13th International Semantic Web Conference (ISWC 2014)*, page 177–180, 2014.
- [176] Ferdinand De Saussure. *Course in general linguistics*. Columbia University Press, 2011.
- [177] Charles Sanders Peirce, Charles Hartshorne, and Paul Weiss. *Collected papers of charles sanders peirce*, volume 5. Harvard University Press, 1935.
- [178] Stevan Harnad. Category induction and representation. *Cognition and Brain Theory*, 5:535–565, 1987.
- [179] Jean M Mandler. How to build a baby: Ii. conceptual primitives. *Psychological review*, 99(4):587, 1992.
- [180] FJ Radermacher. Cognition in systems. *Cybernetics & Systems*, 27(1):1–42, 1996.
- [181] Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988.
- [182] Allen Newell and Herbert A Simon. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3):113–126, 1976.

- [183] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, June 1970. ISSN 0001-0782. doi: 10.1145/362384.362685. URL <http://doi.acm.org/10.1145/362384.362685>.
- [184] Ullrich Hustadt. Do we need the closed world assumption in knowledge representation? In *Working Notes of the KI'94 Workshop: Reasoning about Structured Objects: Knowledge Representation Meets Databases (KRDB'94)*, 1994.
- [185] Alfred Tarski. The concept of truth in formalized languages. *Logic, semantics, metamathematics*, 2:152–278, 1956.
- [186] Stig Kanger. Provability in logic. 1957.
- [187] Saul A Kripke. A completeness theorem in modal logic. *The journal of symbolic logic*, 24(01):1–14, 1959.
- [188] Richard Montague. Formal philosophy; selected papers of richard montague. 1974.
- [189] Jon Barwise and John Perry. Situations and attitudes. 1983.
- [190] Hilary Putnam. *Reason, truth and history*, volume 3. Cambridge University Press, 1981.
- [191] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346, 1990.
- [192] Lars Sivik and Charles Taft. Color naming: A mapping in the imcs of common color terms. *Scandinavian Journal of Psychology*, 35(2):144–164, 1994.
- [193] Daniel N Osherson and Edward E Smith. On the adequacy of prototype theory as a theory of concepts. *Cognition*, 9(1):35–58, 1981.
- [194] Zellig S Harris. Distributional structure. *Word*, 10:146–162, 1954.
- [195] G W. Furnas, T K. Landauer, L M. Gomez, and S T. Dumais. Human factors in computer systems. chapter Statistical Semantics: Analysis of the Potential Performance of Keyword Information Systems, pages 187–242. Ablex Publishing Corp., Norwood, NJ, USA, 1984. ISBN 0-89391-146-1. URL <http://dl.acm.org/citation.cfm?id=818.826>.

- [196] Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.
- [197] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
- [198] Douwe Kiela and Stephen Clark. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) at EACL*, pages 21–30, 2014.
- [199] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JAsIs*, 41(6):391–407, 1990.
- [200] André Freitas, João Gabriel Oliveira, Seán O’riain, João CP Da Silva, and Edward Curry. Querying linked data graphs using semantic relatedness: A vocabulary independent approach. *Data & Knowledge Engineering*, 88:126–141, 2013.
- [201] Alessandro Lenci. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31, 2008.
- [202] David Hume. *Philosophical essays concerning human understanding*. A. Millar, 1748.
- [203] David E Rumelhart, James L McClelland, PDP Research Group, et al. Parallel distributed processing. *Explorations in the microstructure of cognition*, 2:216–271, 1986.
- [204] Ravi Nair. Big data needs approximate computing: Technical perspective. *Commun. ACM*, 58(1):104–104, December 2014. ISSN 0001-0782. doi: 10.1145/2688072. URL <http://doi.acm.org/10.1145/2688072>.
- [205] David S. Johnson. Approximation algorithms for combinatorial problems. In *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, STOC ’73*, pages 38–49, New York, NY, USA, 1973. ACM. doi: 10.1145/800125.804034. URL <http://doi.acm.org/10.1145/800125.804034>.
- [206] Swarup Acharya, Phillip B. Gibbons, Viswanath Poosala, and Sridhar Ramaswamy. Join synopses for approximate query answering. *SIGMOD Rec.*, 28

- (2):275–286, June 1999. ISSN 0163-5808. doi: 10.1145/304181.304207. URL <http://doi.acm.org/10.1145/304181.304207>.
- [207] Viswanath Poosala, Peter J Haas, Yannis E Ioannidis, and Eugene J Shekita. Improved histograms for selectivity estimation of range predicates. In *ACM SIGMOD Record*, volume 25, pages 294–305. ACM, 1996.
- [208] Kaushik Chakrabarti, Minos Garofalakis, Rajeev Rastogi, and Kyuseok Shim. Approximate query processing using wavelets. *The VLDB Journal—The International Journal on Very Large Data Bases*, 10(2-3):199–223, 2001.
- [209] Nesime Tatbul, Uğur Çetintemel, Stan Zdonik, Mitch Cherniack, and Michael Stonebraker. Load shedding in a data stream manager. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*, VLDB '03, pages 309–320. VLDB Endowment, 2003. ISBN 0-12-722442-4. URL <http://dl.acm.org/citation.cfm?id=1315451.1315479>.
- [210] Luis Gravano, Panagiotis G Ipeirotis, Hosagrahar Visvesvaraya Jagadish, Nick Koudas, Shanmugaelayut Muthukrishnan, Divesh Srivastava, et al. Approximate string joins in a database (almost) for free. In *VLDB*, volume 1, pages 491–500, 2001.
- [211] Hong-Hai Do, Sergey Melnik, and Erhard Rahm. Comparison of schema matching evaluations. In *Web, Web-Services, and Database Systems*, pages 221–237. Springer, 2003.
- [212] Françoise Fabret, François Llirbat, Joao Pereira, I Rocquencourt, and Dennis Shasha. Efficient matching for content-based publish/subscribe systems. In *Proc. CoopIS*, 2000.
- [213] Z Liu, S Parthasarathy, A Ranganathan, and H Yang. Near-optimal algorithms for shared filter evaluation in data stream systems. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 133–146. ACM, 2008.
- [214] Zohra Bellahsene, Angela Bonifati, Fabien Duchateau, and Yannis Velegarakis. On evaluating schema matching and mapping. In *Schema matching and mapping*, pages 253–291. Springer, 2011.

- [215] Yoonkyong Lee, Mayssam Sayyadian, AnHai Doan, and Arnon S. Rosenthal. etuner: Tuning schema matching software using synthetic scenarios. *The VLDB Journal*, 16(1):97–122, January 2007. ISSN 1066-8888. doi: 10.1007/s00778-006-0024-z. URL <http://dx.doi.org/10.1007/s00778-006-0024-z>.
- [216] Bogdan Alexe, Wang-Chiew Tan, and Yannis Velegrakis. STBenchmark: towards a benchmark for mapping systems. *Proceedings of the VLDB Endowment*, 1(1): 230–244, 2008.
- [217] Merriam-Webster’s. Merriam-Webster’s Collegiate® Thesaurus, 2012. URL <http://www.dictionaryapi.com/products/api-collegiate-thesaurus.htm>.
- [218] Edward Curry, Souleiman Hasan, and Sean O’Riain. Enterprise energy management using a linked dataspace for energy intelligence. In *Proc. SustainIT*, pages 1–6. IEEE, 2012.
- [219] Wassim Derguech, Souleiman Hasan, Sami Bhiri, and Edward Curry. Organizing Capabilities Using Formal Concept Analysis. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 260–265, 2013.
- [220] Wassim Derguech, Sami Bhiri, Souleiman Hasan, and Edward Curry. Using Formal Concept Analysis for Organizing and Discovering Sensor Capabilities. *The Computer Journal*, 2014. doi: 10.1093/comjnl/bxu088. URL <http://comjnl.oxfordjournals.org/content/early/2014/09/11/comjnl.bxu088.abstract>.
- [221] Yahoo! Yahoo! Directory: Automotive - Makes and Models, 2013. URL [http://dir.yahoo.com/recreation/automotive/makes\\_and\\_models/](http://dir.yahoo.com/recreation/automotive/makes_and_models/).
- [222] Kyle Anderson, Adrian Ocneanu, Diego Benitez, Derrick Carlson, Anthony Rowe, and Mario Berges. BLUED: a fully labeled public dataset for Event-Based Non-Intrusive load monitoring research. In *Proc. SustKDD*, August 2012.
- [223] Richard Cyganiak. Rooms in the DERI building, 2013. URL <http://lab.linkeddata.deri.ie/2010/deri-rooms>.
- [224] George A Miller. WordNet: a lexical database for English. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782.
- [225] John G Breslin, Alexandre Passant, and Stefan Decker. Social tagging. In *The social semantic web*, pages 137–158. Springer, 2009.

- [226] John G Breslin, Alexandre Passant, and Stefan Decker. Introduction to the social web (web 2.0, social media, social software). In *The Social Semantic Web*, pages 21–44. Springer, 2009.
- [227] Shengliang Xu, Shenghua Bao, Ben Fei, Zhong Su, and Yong Yu. Exploring folksonomy for personalized search. In *Proceedings of the 31st Annual International ACM SIGIR Conference*, pages 155–162. ACM, 2008.
- [228] Roberto Navigli. A quick tour of word sense disambiguation, induction and related approaches. In *SOFSEM 2012: Theory and practice of computer science*, pages 115–129. Springer, 2012.
- [229] Jeff Z Pan, Stuart Taylor, and Edward Thomas. Reducing ambiguity in tagging systems with folksonomy search expansion. In *The Semantic Web: Research and Applications*, pages 669–683. Springer, 2009.
- [230] Jose Antollini, Mario Antollini, Pablo Guerrero, and Mariano Cilia. Extending rebecca to support concept-based addressing. In *Proceedings of the Argentinean Symposium on Information Systems (ASIS'04)*, 2004.
- [231] Wenfei Fan and Floris Geerts. Capturing missing tuples and missing values. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 169–178. ACM, 2010.
- [232] Alon Y Levy. Obtaining complete answers from incomplete databases. In *VLDB*, volume 96, pages 402–412. Citeseer, 1996.
- [233] Simon Razniewski and Werner Nutt. Checking query completeness over incomplete data. In *Proceedings of the 4th International Workshop on Logic in Databases*, pages 32–32. ACM, 2011.
- [234] Amir Parssian, Sumit Sarkar, and Varghese S Jacob. Assessing information quality for the composite relational operation join. In *IQ*, pages 225–237, 2002.
- [235] Edward Curry. Message-oriented middleware. *Middleware for communications*, pages 1–28, 2004.
- [236] Wenfei Fan and Floris Geerts. Relative information completeness. *ACM Trans. Database Syst.*, 35(4):27:1–27:44, October 2010. ISSN 0362-5915. doi: 10.1145/1862919.1862924. URL <http://doi.acm.org/10.1145/1862919.1862924>.

- [237] Edgar F Codd. Extending the database relational model to capture more meaning. *ACM Transactions on Database Systems (TODS)*, 4(4):397–434, 1979.
- [238] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-76297-3. doi: 10.1007/978-3-540-76298-0\_52. URL [http://dx.doi.org/10.1007/978-3-540-76298-0\\_52](http://dx.doi.org/10.1007/978-3-540-76298-0_52).
- [239] Tim Berners-Lee. Linked data-design issues (2006). URL <http://www.w3.org/DesignIssues/LinkedData.html>, 2011.
- [240] Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.
- [241] Allan M Collins and Elizabeth F Loftus. A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407, 1975.
- [242] Edward Curry, James O’Donnell, Edward Corry, Souleiman Hasan, Marcus Keane, and Seán O’Riain. Linking building data in the cloud: Integrating cross-domain building data using linked data. *Advanced Engineering Informatics*, 27(2):206–219, 2013.
- [243] Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks. The semantic web: The roles of xml and rdf. *Internet Computing, IEEE*, 4(5):63–73, 2000.
- [244] Stefan Decker and Martin R Frank. The networked semantic desktop. In *WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, volume 105, pages 1613–0073, 2004.
- [245] Eric Prud’Hommeaux, Andy Seaborne, et al. Sparql query language for rdf. *W3C recommendation*, 15, 2008.

- [246] Cristiano Rocha, Daniel Schwabe, and Marcus Poggi Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web*, pages 374–383. ACM, 2004.
- [247] Xing Jiang and Ah-Hwee Tan. Ontosearch: A full-text search engine for the semantic web. In *AAAI*, volume 6, pages 1325–1330, 2006.
- [248] Fabio Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.
- [249] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.
- [250] Souleiman Hasan, Kalpa Gunaratna, Yongrui Qin, and Edward Curry. Demo: approximate semantic matching in the collider event processing engine. In *Proceedings of the 7th ACM international conference on Distributed event-based systems*, pages 337–338. ACM, 2013.
- [251] Edward Curry, Souleiman Hasan, Umair ul Hassan, Micah Herstand, and Sean O’Riain. An Entity-Centric Approach to Green Information Systems. In *The 19th European Conference on Information Systems (ECIS)*, page Paper 194, Helsinki, Finland, 2011. AIS Electronic Library (AISeL).
- [252] Edward Curry, Souleiman Hasan, Mark White, and Hugh Melvin. An Environmental Chargeback for Data Center and Cloud Computing Consumers. *First International Workshop on Energy-Efficient Data Centers*, pages 117–128, 2012.
- [253] Souleiman Hasan, Richard Medland, Marcus Foth, and Edward Curry. Curbing resource consumption using team-based feedback. In *Persuasive Technology*, pages 75–86. Springer, 2013.
- [254] Ted Pedersen, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G Chute. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of biomedical informatics*, 40(3):288–299, 2007.
- [255] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30, 1989.

- [256] Jorge E Caviedes and James J Cimino. Towards the development of a conceptual distance metric for the umls. *Journal of biomedical informatics*, 37(2):77–85, 2004.
- [257] Phillip W. Lord, Robert D. Stevens, Andy Brass, and Carole A. Goble. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283, 2003.
- [258] Christopher G Chute and Yiming Yang. An evaluation of concept based latent semantic indexing for clinical information retrieval. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 639. American Medical Informatics Association, 1992.
- [259] James O'Donnell, Edward Corry, Souleiman Hasan, Marcus Keane, and Edward Curry. Building performance optimization using cross-domain scenario modeling, linked data, and complex event processing. *Building and Environment*, 62:102–111, 2013.
- [260] Marco Grassi, Michele Nucci, and Francesco Piazza. Towards an ontology framework for intelligent smart home management and energy saving. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on*, pages 1753–1758. IEEE, 2011.
- [261] M Kazuyuki, K Mishina, F Ren, and S Kuroiwa. Industrial automation systems and integration-open systems application integration framework part 3: Reference description for iec 61158-based controlsystems, 2003.
- [262] Konstantinos Togias, Christos Goumopoulos, and Achilles Kameas. Ontology-based representation of upnp devices and services for dynamic context-aware ubiquitous computing applications. In *Communication Theory, Reliability, and Quality of Service (CTRQ), 2010 Third International Conference on*, pages 220–225. IEEE, 2010.
- [263] Sebastian Hegler and Martin Wollschlaeger. The semantic web in action: semantically enabled device descriptions. In *Industrial Informatics, 2007 5th IEEE International Conference on*, volume 2, pages 1013–1018. IEEE, 2007.
- [264] IEC IEC. 61970-301: Energy management system application program interface (ems-api)-part 301: Common information model (cim) base. Technical report, Technical report, IEC-International Electrotechnical Commission, 2003.

- [265] International Electrotechnical Commission et al. Iec 61968-1 application integration at electric utilities—system interfaces for distribution management part 1: Interface architecture and general requirements. *IEC Reference number IEC*, pages 61968–1, 2003.
- [266] Marta Majewska, Bartosz Kryza, and Jacek Kitowski. Translation of common information model to web ontology language. In *Computational Science–ICCS 2007*, pages 414–417. Springer, 2007.
- [267] Stephen Quiroigico, Pedro Assis, Andrea Westerinen, Michael Baskey, and Ellen Stokes. Toward a formal common information model ontology. In *Web Information Systems–WISE 2004 Workshops*, pages 11–21. Springer, 2004.
- [268] Ming Ding, Zhengkai Zhang, and Xuefeng Guo. Cim extension of microgrid energy management system. In *Power and Energy Engineering Conference, 2009. APPEEC 2009. Asia-Pacific*, pages 1–6. IEEE, 2009.
- [269] Edward Curry, Viktoriya Degeler, Eoghan Clifford, Daniel Coakley, A Costa, SJ van Andel, N van de Giesen, C Kouroupetroglou, T Messervey, J Mink, et al. Linked water data for water information management. In *11th International Conference on Hydroinformatics (HIC), New York, New York, USA*, 2014.
- [270] Michael Franklin, Alon Halevy, and David Maier. From databases to dataspace: a new abstraction for information management. *ACM Sigmod Record*, 34(4):27–33, 2005.
- [271] Alon Halevy, Michael Franklin, and David Maier. Principles of dataspace systems. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–9. ACM, 2006.