Tutorial: Tackling Variety in Event-Based Systems

Souleiman Hasan
Insight Centre for Data Analytics
National University of Ireland, Galway
Ireland
souleiman.hasan@insight-centre.org

Edward Curry
Insight Centre for Data Analytics
National University of Ireland, Galway
Ireland
edward.curry@insight-centre.org

ABSTRACT

Event-based systems follow an interaction model based on three decoupling dimensions: space, time, and synchronization. However, event producers and consumers are tightly coupled by event semantics: types, attributes, and values. That limits scalability in large-scale heterogeneous environments with significant variety such as the Internet of Things (IoT) due to difficulties in establishing semantic agreements at such scales. This paper studies this problem and investigates the suitability of different traditional and emerging approaches for tackling the issue.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*Internet*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems*

General Terms

Theory

Keywords

event processing systems, semantics, coupling, thingsonomy, Internet of Things

1. INTRODUCTION

Recent trends of Big Data and the Internet of Things pose challenges to current computational paradigms such as event processing systems. Three dimensions of Big Data are identified (Volume, Velocity, and Variety) [17]. While Volume and Velocity are active areas of research, we think that more attention needs to be given to the Variety aspects within distributed event based systems. This paper sheds light on the suitability of the semantic assumptions in the current event processing paradigm. The contribution of this paper is the analysis of the event processing paradigm in terms of a communication problem that crosses system boundaries, together with a literature review and analysis of the suitability of various models of semantics such as symbolic, subsymbolic, and non-symbolic models.

In Section 2 we identify some new trends in the data landscape. Section 3 revisits a set of computational paradigms and traces the evolution to event processing systems. Section 4 provides a theory to event exchange based on communication and crossing systems boundaries. In Section 5 we analyze semantic models and their relation to the theory of event exchange and semantic coupling. Section 6 reviews current approaches in event processing to address semantic coupling. Section 7 outlines the thematic event processing approach while Section 8 shows how an architecture to IoT can be built based on the idea of thingsonomies and thematic event processing. We conclude the paper along with future challenges in Section 9.

DEBS'15, June 29 - July 3, 2015, Oslo, Norway.

DOI: http://dx.doi.org/10.1145/2675743.2774215.

2. THE INTERNET OF THINGS AND NEW TRENDS

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 by 2020 there will be about 50 billion devices connected to the Internet [32]. This leads to challenges in information processing solutions as the volume, velocity, and variety of data increase. Smart cities, smart grids, and cyber-physical systems have been the topics of active research throughout the last decade. A technology enabler for such areas is represented by the Internet of Things [2].

2.1 The Internet of Things

From a high-level architectural perspective, IoT can be divided into three tiers [2]:

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 key 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 [2] serve this layer of IoT.

2. **Middleware layer** which encompasses common functionalities and abstracts application developers and users from IoT infrastructure details. Among the technologies to contribute to this layer are Service-Oriented Architectures (SOA) [2] and event processing systems which support functionalities such as early filtering of events, spatio-temporal correlation, sequencing, event enrichment, and complex event processing.

3. **Application layer** which builds upon the middleware to provide direct and potentially domain specific benefits to users. IoT promises new domains of applications in transportation and logistics, healthcare, smart environments, analytics, and more futuristic applications such as robo-taxis and virtual reality.

2.2 IoT and Big Data

While significant efforts in the area of IoT come from communication and networking communities, there is a growing realization that the challenges will be more prevalent at the data level [1] including data collection, management, and analytics. At this level, IoT can be linked with the area of Big Data. In a report by McKinsey [30], it has been estimated that in 2010 alone enterprises and users stored more than 13 exabytes of new data, which is around 50,000 times the size of the Library of Congress. Nonetheless, the Big Data trend should not be understood just in terms of data volume. In fact, one of the most commonly used analysis recognizes three dimensions of the phenomena [17]:

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

2.3 Significant Trends in the Data Landscape

The trends of IoT and Big Data signify a considerable shift in the characteristics of information production, communication, and consumption, as follows:

- *Number of sources* that can create data has been significantly increasing. For example, the International Telecommunication Union (ITU), a United Nations organization, reports that the number of worldwide mobile subscribers has increased from 2,205 millions in 2005 to 6,662 millions in 2013, i.e. more than 200% increase in 8 years [26].
- *Heterogeneity* is increasing in various forms including different types of networks, protocols, devices, data formats and representation [2]. Semantic heterogeneity in IoT follows partially from the number of devices and manufacturers of these devices. In the Semantic Web, for instance, the Falcons search engine could discover in 2008 about 4,000 ontologies on the Web [7], this number has increased to more than 6,400 in 2015 [12], i.e. more than 50% increase in 7 years. Similar phenomena can be assumed to happen in IoT.
- *Number of users* who have access to data has been drastically increasing. The ITU reports an increase in the number of individuals using the Internet from 1,024 millions in 2005, to 2,710 millions in 2013 [26], i.e. more than 160% increase in 8 years. Most of those users come from a non-technical background.
- *Organization of users* in large-scale environments is minimal and users are decoupled and autonomous. An example is the creation of *Wikipedia*, a very comprehensive encyclopedia created by distributed users from all over the world. Studies of the demographics of crowdsourcers reveal a global diversity and geographical distributions of crowdsourcers [36]. We suggest that similar characteristics can be assumed in IoT settings where users will have a minimal organization.
- *Timeliness* or velocity is a challenge due to the large volume of data available in Big Data settings such as IoT. It becomes very important to filter important data items as early as possible.

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

3. COMPUTATIONAL PARADIGMS

Throughout the last few years, there has been a realization that a new class of information processing systems is needed. The new class, or 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, stock market analysis, RFID-based anomaly detection in inventories, security systems, etc.

Hinze et al. [25] analyze various applications that could justify the need for the new paradigm. They developed a framework that correlates features to application classes. They include for example: spatio-temporal correlation, event sequencing, out of order events, derived events, event enrichment, mobility of event subscriber, etc.

An analysis by Cugola and Margara [9] can be used to complement this picture and grasp the essence to justify a new paradigm. They state that "The concepts of timeliness and flow processing are crucial for justifying the need for a new class of systems." The event processing paradigm has evolved through the work of several communities in whose artifacts elements of the paradigm can be detected.

3.1 Active Databases

Active databases started to appear during the late twentieth century [33]. The term *active* is put in contrast with the term *passive* that was assumed to exist in database systems prior to the appearance of active databases.

3.2 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 [31, p. 2]. Within the context of static networks, middleware systems view data and services stationary in objects of databases allowing an interaction model of request/reply to and from the stationary nodes, e.g. the Remote Procedure Call (RPC) paradigm and its derivative client/server architecture.

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]. Thus, asynchronous and decoupled extensions have been added to the existing middleware systems such as the extension of J2EE and the Real-time CORBA Event Service.

3.3 Event-based Software Engineering

Complex software systems consist of many integrated components that collaborate to achieve the overall system goal. In object-oriented architectures for instance the classical way of components to interact with each other is via *explicit invocation*. An alternative is *implicit invocation* which is widely used now in enterprise application integration, graphical user interfaces, and aspect-oriented programming.

3.4 Message-Oriented Middleware

In the Internet architecture, nodes can communicate via a coupled, synchronous, and end-to-end interaction scheme [11]. The Internet has become a platform for distributed applications that exchange information in a way that the location and behavior of these applications are dynamic. Thus, a decoupled interaction scheme has become crucial to the development of large-scale applications.

Eugster et al. [11] give decoupling high importance with respect to scalability. They recognize three dimensions of decoupling: *space*, *time*, and *synchronization* which are concerned with addresses, activity time, and blocking respectively. Communication paradigms such as remote procedure call and shared spaces are coupled on one or more dimensions. The publish/subscribe paradigm evolved to overcome this issue [11]. Various publish/subscribe schemes exist including: topic-based, content-based, type-based, and concept-based publish/subscribe systems.

3.5 Data Stream Management Systems

Active databases do not scale under high rates of database updates or large number of rules [9]. Thus, the database community developed Data Stream Management Systems (DSMSs) to cope with this issue. Streams form the basic concepts in DSMS as opposed to tables in conventional databases. DSMSs adopt an interaction paradigm based on *continuous queries* which are registered by users.

Example DSMS engines include: TelegraphCQ, OpenCQ, NiagaraCQ, Tribeca, CQL/Stream, Aurora/Borealis, Gigascope, and Stream Mill. Commercial DSMSs also exist such as Sybase Coral8 Engine, SteamBase, and IBM System S. In DSMS data items are homogeneous in a stream, they do not typically have temporal or causal semantics, and languages are typically of a transformation nature.

3.6 Complex Event Processing

Data stream management systems do not associate a 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 a specific semantics to their data items. They are computer objects which 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 would process one event at a time. Events are matched against subscriptions without looking at what previous events occured before in the history. These 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* or *complex* event.

A Complex Event Processing (CEP) engine thus emphasizes the matching of event patterns specifically with ordering conditions such as temporal sequencing and causal relationships. Examples of CEP engines include: Rapide, GEM, Padres, DistCED, Cayuga, NextCEP, Raced, Amit, PB-CED, Sase, Sase+, Peex, and TESLA/T-Rex. Commercial systems also exist such as SAP Event Stream Processor, Oracle Event Processing, Esper, TIBCO Business Events, and IBM WebSphere Business Events. In CEP systems, data items are heterogeneous, they typically have temporal or causal semantics, and languages are typically with a detection nature of clearly distinguished parts for condition and action.

4. A THEORY FOR EVENT EXCHANGE

A theory for event exchange is useful to abstract the characteristics of the discussed computational paradigms. We start this discussion with an analysis of three main technical traits of large-scale event processing systems.

4.1 Traits of Large-Scale Event Processing Systems

We suggest that the following traits are fundamental characteristics for event processing systems at large scales:

- *Distribution*: distribution can be understood from two complementary aspects. The first aspect is the placement of processing workloads on different nodes and thus making use of parallel computing. The second aspect is that large-scale environments are inherently distributed with event production and consumption happening at distributed components. 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.

- *Heterogeneity*: heterogeneity occurs in terms of differences in hardware components, protocols, operating systems, middleware, and data. This paper is concerned with data heterogeneity in event systems as described by Mühl et al. [31]: "Syntax and semantics of notifications are likely to vary and there are inevitably different data models in use." We do not deal with syntax heterogeneity within this work. To define heterogeneity, we start by defining semantics first. We draw here on Gärdenfors [16, p. 151], and define semantics 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 found in this definition which are discussed in Section 5. The first is the set of meanings $\mathbb{M}$, and the other 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. We deal mainly with terms in event systems, with very little focus on syntax. Semantic heterogeneity, or variety, can then be defined as the use of different mappings $\mathcal{S}_i$ from $\mathbb{M}$ to $\mathcal{L}$ by different event agents a_i.

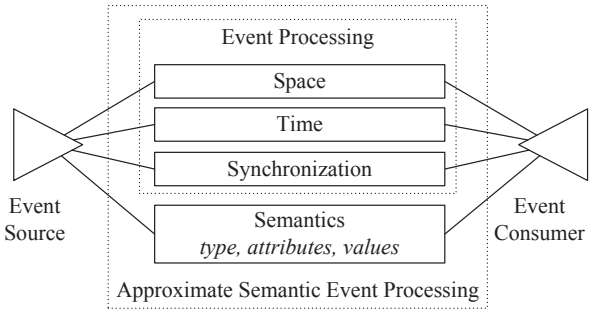

Figure 1: Decoupling dimensions.

- *Openness*: While the term *"open"* has been used frequently in the literature to describe distributed event systems at large scales, it has not been defined precisely. Herein we draw upon the definition used in systems theory [40, p. 139–153] as *a system that has external interactions in form of information, energy, or matter transfer through its boundary.* We define an open event system from the semantics perspective as the event environment where an agent can exchange events with other agents that use different event semantics, i.e. event agent x which has semantics mapping $\mathcal{S}_x$ can in theory exchange events with another event agent y which has semantics mapping $\mathcal{S}_y$.

4.2 The Principle of Decoupling

A fundamental principle to 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]. Eugster et al. [11] recognize three dimensions of decoupling as shown in Figure 1:

- *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* [31, p. 150], where the control over an event-based system has been decentralized into an autonomous version. We 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, this hypothesis can not be accepted in an absolute sense, and needs to take other assumptions into considerations.

The now autonomous events may lead to ambiguities in semantics that requires participants to collaborate again in order to resolve. Because that leads to limitations in scalability, it undermines the fundamental reason why participants are decoupled. Thus, any computational paradigm that tackles event processing in a large-scale, distributed, open, and heterogeneous environment must take into consideration that it has valid assumptions that do not break the principle of decoupling, and thus do not affect scalability.

4.3 The Model of Communication

Large-scale event processing systems are distributed, open, and heterogeneous, with decoupled components which exchange messages. This requires an abstraction which helps better analyze these systems and their challenges. A useful abstraction is a communication model. One of the earliest models is the mathematical model of communication developed by Shannon and Weaver [39]. The model consists of six elements: an information source, a transmitter, a channel, noise, a receiver, and a destination.

Chandler [6] in his work on semiotics, the theory of signs and meanings, analyzes communication models. 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 [6, 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 aspects of communication, which is crucial for communication to succeed. In fact, this has been left out of the model deliberately as stated by Shannon and Weaver [39]:

> "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."

We argue that the current event processing paradigm puts much focus on the transmission aspect of the problem of communication, while it should in the end serve the bigger problem, consequently human agents have to step into the loop considerably in order to complement the event-based paradigm. That leads to issues on the paradigm itself, such as when human agents have to agree on semantics to complement the communication purpose. As a result, they introduce coupling into the paradigm, which in principle contradicts its basic premise of "decoupling for scalability" as discussed in Section 4.2.

4.4 Event Exchange as Crossing Boundaries

Another useful abstraction which we use here is the view of event processing systems from the perspective of a knowledge exchange framework. Such a framework has been proposed by Carlile [3] within the field of organization science. Carlile recognizes three main levels of boundaries that may exist in a given knowledge exchange scenario:

- *Syntactic boundary* between systems focuses on the sharing and establishment of a common syntax across a given boundary. This view has been established with Shannon and Weaver [39]. Carlile [3] sees that crossing such boundaries is synonymous to *transferring* knowledge across those boundaries.
- *Semantic boundary* starts to appear when some meanings become unclear or ambiguous. Even with a syntax, interpretations can be different between the two

boundary sides. The essential premise is that a message conveys meanings rather than mere symbols, which is the emphasis of linguists such as Reddy [35] in his theory of the "metaphor of conduit" stating that language reveals metaphors about communication as meanings are conveyed through language containers. Carlile [3] sees crossing such boundaries as synonymous to *translating* knowledge across those boundaries.

- *Pragmatic boundary* appears when assessing the exchanged knowledge needs a bigger picture of the interacting parties' interests and contexts. The origin of the pragmatic view can be traced back to works by semioticians such as Peirce [6]. Carlile [3] sees crossing such boundaries as synonymous to *transforming* knowledge across those boundaries.

4.5 Coupling and Scalability Trade-off

Event agents, i.e. systems, have boundaries that they have to cross in order to communicate with other systems. Such boundaries are syntactic, semantic, and pragmatic. Events are not mere exchange of symbols, but rather meanings signified by symbols (hence the semiotics view [35]).

Events must effectively cross the three levels of boundaries in order to establish communication and collaboration between event agents. We 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 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 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 and heterogeneous environments such as the IoT. This in turn leads to magnifying the problematic nature of *semantic coupling*, as shown in Figure 1, which contradicts the fundamental basis of event systems as decoupled and scalable systems.

5. SEMANTICS AND APPROXIMATION

Semantics generally refer to a relationship between two spaces (or worlds or sets): the meanings, and the symbols. This view is mostly apparent in the field of semiotics [6] where the focus is on signs and sign systems. 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" [16, 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"*. We discuss these notions with respect to the three levels of meaning representation in the following sections.

5.1 Symbolic Representation of Meaning

The key principle of symbolism is that information is represented by *symbols*, and processing of information is by definition a *manipulation of symbols* through *rules* [16, p. 35–36]. 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 Artificial Intelligence (AI). Thus, meanings are purely the result of logical, syntactic relations of symbols, rather than the states they refer to.

This tradition extends beyond AI to other areas such as databases and event-based systems where semantic assumptions are quite derived from those of databases. For instance, in the relational model Codd [8] proposes a "a relational view of data" such that cells generally contain constants, which is the data naming for symbols. Schema is handled similarly. Another indication of the symbolic paradigm in databases comes from the Unique Name Assumption (UNA) which states that two distinct symbols designate two different objects in the universe.

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. There are three directions to tackling the question of what a property, or a concept, is in symbolic semantics [16, p. 60–62]:

- *Extensional Semantics* where a property is defined by the set of objects in the world that have the property.
- *Intensional Semantics* alters the concept of one world to the case of multiple *possible worlds*.
- *Situation Semantics* uses 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*.

In ontologies for instance, properties and concepts are described using *TBox* statements. Objects on the other hand are described using *ABox* statements which are compliant with the TBox terminological description.

Symbolic semantics have been criticized from various aspects, as summarized by Gärdenfors [16, p. 37–40, 62–66]:

1. They do not explain how a person can perceive two properties to be similar.
2. Their limited account for inductive reasoning.
3. The *frame problem* which states that representing all necessary knowledge about the world requires a combinatorial explosion of logical axioms and inferences.
4. The *symbol grounding problem* which states that in the symbolic paradigm the meanings of symbols are actually grounded in symbols themselves [19].

We add to this critique that symbolism does not largely separate the symbolic level from the meaning level. When event agents need to agree on the meanings, the essence of information exchange, they have to agree on symbols due to the tight relationship between meanings and symbols. Agreeing on symbols is a highly costly process and thus hinders the loose semantic coupling requirement. This is a result of the lack of a natural account for similarity in the symbolic models of meanings. The existence of similarity and topology between meanings can lower the amount of information that two parties need to agree on.

5.2 Conceptual Representation of Meaning

At the conceptual level come various alternatives of meaning models that fundamentally leverage topological features

Table 1: Characteristics of Semantic Models

	Symbolic	Conceptual			Connectionist
		Conceptual Spaces [16]	**LSA** [10]	**ESA** [14]	
Loose semantic coupling	-	+	++	++	+
Efficiency	+++	++	++	++	+
Effectiveness	+	++	++	+++	++
Support for similarity	-	+++	+++	+++	++
Easy to build	- -	+	+++	+++	+
Easy to interpret	+++	++	-	+	- -

Legend: the model excellently (+++), moderately (++), slightly (+) has the characteristic
the model moderately (- -), mildly (-) lacks the characteristic

of meanings. What distinguishes these approaches is a geometrical nature of the meaning space. In such a geometry, distances and closeness between meanings can be established.

An example of conceptual representation of meaning is the Conceptual Spaces proposed by Gärdenfors [16]. Gärdenfors starts from the observation that concepts are not independent from each others, but rather are structured into *domains*, e.g. the domain of colors, the spatial domain, etc. Conceptual spaces are then built up from *quality dimensions* which serve the purpose of building the domains. For instance, the colors domain can be built up from three dimensions: *hue*, *chromaticness* or saturation, and *brightness*.

We think that the computational challenge with Gärdenfors' conceptual spaces is building quality dimensions, and to agree on quality dimensions can be hard to achieve at a large scale. A computational implementation of conceptual spaces is needed such that it builds a geometrical space which supports the basic notions of distance and similarity. Let us assume that we have a large number of textual documents, if two terms such as *'water'* and *'fluid'* frequently occur with each other, one can assume that they are close within the meaning space. This particular observation is the tenet of a class of approaches within computational linguistics known as *statistical semantics* or *distributional semantics*. Statistical semantics is based on the distributional hypothesis which states that words that occur in the same contexts tend to have similar meanings [20]. We describe it as subsymbolic to emphasize its relative relationship to the symbolic approach.

5.2.1 Vector Space Models

One of the widely used mathematical tools to formalize and deal with distributional semantics are Vector Space Models (VSM). The premise is that a multi-dimentional vector space is built out of some textual corpora that reflect the usage of terms in a domain agnostic of domain specific setting. A term or a meaning becomes a vector in the space with coordinates for each component. VSM has a highly automatic nature to build knowledge.

Matrices are the basic elements used to encode terms statistical occurrences. For instance, a term-document matrix encodes the number of times a term occurs in a document of the corpus. Weighting schemes such as Term Frequency Inverse Document Frequency (TF/IDF) gives more weight to a term if it appears more often in a document and less often in other documents. One example widely used in cognitive science and information retrieval is the Latent Semantic Analysis (LSA) [29, p. 369–383]. LSA builds upon a term-document matrix and reduces space dimensionality using an algebraic approach named Singular Value Decomposition (SVD).

5.2.2 Explicit Semantic Analysis

Gabrilovich and Markovitch [14] 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 judgment. 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('water', 'fluid') > esa('water', 'car')* as the formers appear frequently in common articles. Semantic relatedness between a pair of terms can be measured using cosine distance between their corresponding vectors.

Distributional semantics and similar models are criticized, as discussed by Lenci [28], on the grounds of compositionality. That is they mainly concern lexical meanings, i.e. meanings of individual terms, rather than complex sentences. We argue that the compositionality problem is not an issue for event matching. That is due to the fact that 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 suitable for the problem at hand.

5.3 Subconceptual Representation of Meaning

At this level lie a class of non-symbolic approaches to meaning representation such as connectionism. Connectionist systems are Artificial Neural Networks (ANNs) and consist of a large number of units, the neurons, connected together. The *state* of the network at a specific point could be thought of as a meaning or idea [37]. A geometrical interpretation can be given to ANNs as in conceptual spaces. Critics to connectionist models come from two main aspects [16, p. 42-43]: learnability which states that ANNs need a large training set to learn structure and adjust weights, and the difficulty to interpret what an emerging network means.

The vector space distributional model based on explicit semantic analysis appears to meet the main requirement of loose semantic coupling. It also has the favorable characteristics of efficiency, effectiveness, support of similarity, and ease of building and interpretation as discussed in the previous sections and summarized in Table 1.

Table 2: Current Approaches to Semantic Coupling

	Content-based [5]	**Concept-based** [34, 41]	**Approximate Semantic Event Processing** [21, 24]	**Thematic Event Processing** [22]
Matching	exact string matching	Boolean semantic matching	approximate semantic matching	approximate semantic matching
Semantic coupling	term-level full agreement	concept-level shared agreement	loose agreement	loose agreement
Semantics	not explicit	top-down ontology-based	statistical distributional semantics	statistical distributional semantics
Domain specificity cost	defining a large number of domain rules	defining a domain-specific ontology	indexing a domain-specific corpus	parametrizing the vector space of an open domain corpus
Effectiveness (F_1Score)	100%	depends on the domains and number of concept models	depends on the corpus	depends on the corpus and the themes tags. Outperforms non-thematic approximate approach
Cost	defining a large number of rules and establishing shared agreement on terms	establishing shared agreement on ontologies	minimal agreement on a large textual corpus	minimal agreement on a large textual corpus and associating good themes tags
Efficiency (throughput)	high	medium to high	medium to high	medium to high

5.4 Free Tagging and Thingsonomies

While subsymbolic or non-symbolic communication can be a good solution to semantic coupling, humans are still symbolic in nature. Tagging is a mechanism by which humans behind event producers and consumers can add some information to events and subscriptions to enhance the meaning expressed by subscriptions and exchanged with events via a better symbolic approximation.

In Web 2.0 users can tag content such as images, tweets, blog posts, etc. [18]. Websites supporting social tagging has emerged and become popular, e.g. Delicious, Flicker, Twitter, etc. It has been found within the research on social media that fixed static taxonomies are not a suitable approach within a social tagging context [18]. That is mainly because fixed taxonomies are rigid and centralized, cannot easily keep up with an evolving corpus, a controlled vocabulary is expensive to build and maintain in terms of development time, and they present a steep learning curve to users.

We argue that top-down organization of semantic models increases the problem of semantic coupling, which already exists due to the granularity of such models which are symbolic in nature. Folksonomies, (folk (people) + taxis (classification) + nomos (management)), use terms, freely generated by users, and freely used by users to tag resources. To this end, we argue that bottom-up free tagging of events and subscriptions is a good way to manage their semantics in a loosely coupled way. We proposed in [23] an approach where this concept is applied on things within an IoT context, leading to the concept of *thingsonomies*.

5.5 Approximation in Event Processing

The need for approximate models stems from loosening the coupling on the semantic level. Coupling is important to cross semantic and pragmatic boundaries, but it limits scalability. Loosening coupling at these levels is a compromise to tackle the tradeoff between decoupling for scalability and crossing the 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.

Approximate computing has been investigated in the literature as a response to various problems: time efficiency such as approximation algorithms where finding an optimal solution can have a combinatorial time [27], and full integration such as uncertain schema matching with the realization that matchers are inherently uncertain [15]. We proposed in previous work an approximate approach to event processing that leverages probabilistic matching of events [21, 24].

6. CURRENT APPROACHES TO SEMANTIC COUPLING

Current approaches to semantic coupling are shown in Table 2. In the *content-based approach*, event sources and consumers use the same event types, attributes and values as assumed in traditional content-based publish/subscribe systems such as SIENA [5]. The approach has high semantic coupling between parties and works well in environments with a low level of data heterogeneity. In the *concept-based approach*, participants can use different terms and still expect event matchers to match them properly thanks to an explicit knowledge representation that encodes semantic relationships between terms. Examples of knowledge representations are thesauri and ontologies as in S-TOPSS [34] and semantic pub/sub [41]. Building such knowledge representations is a time consuming process.

Freitas et al. proposed an approximate query processing approach for databases based on distributional semantics [13]. In our previous work [21, 24], we proposed an *approximate semantic event processing approach* and showed that the model is suitable when participants agree on some event types, attributes, or values while performance decreases when an absolute 100% degree of approximation is required.

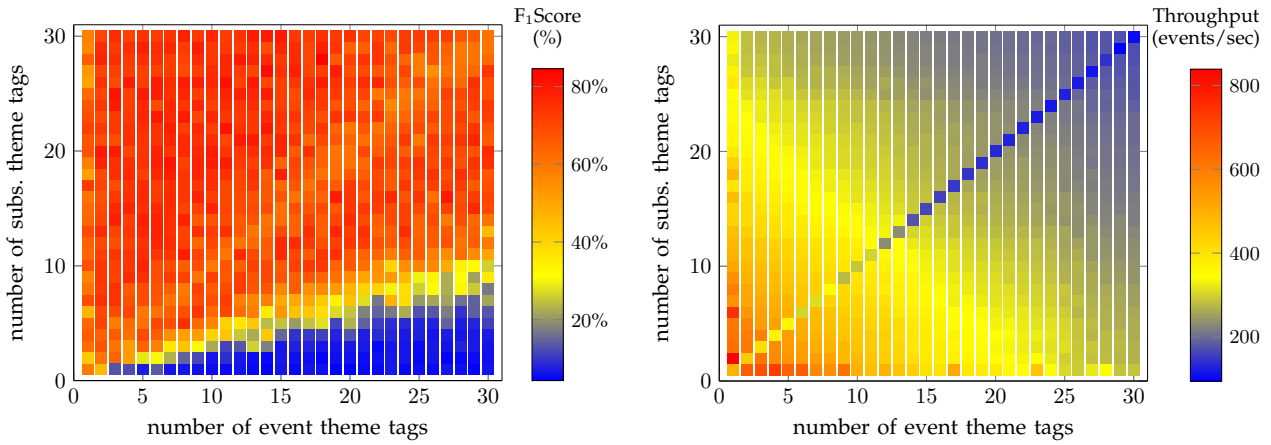

Figure 2: Evaluating IoT semantic normalization: effectiveness (left) and time efficiency (right) [22].

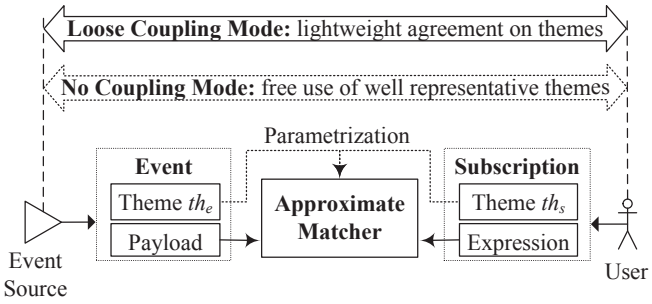

Figure 3: Thematic event matching model.

7. THEMATIC EVENT PROCESSING

In [22] we proposed an approach for loose semantic coupling based on the use of distributional semantics, free tagging, and approximation. The thematic event processing approach suggests associating thematic tags that describe the themes of types, attributes, and values to clarify their meanings as shown in Figure 3. An approximate matcher exploits the associated tags to parametrize the vector space model and improve approximations of the meanings in events and subscriptions.

Evaluation of the matching quality can be achieved by establishing a gold standard set of subscriptions, events, and thematic tags, along with known ground truth of true event matchings. For each subscription, the set of relevant events is identified. *Precision* represents the ratio of correctly matched events versus all the matched events. *Recall* represents the ratio of correctly matched events versus all the relevant ones. The effectiveness of the built software can be measured by precision, recall, and a derivative measure that combines both in one number such as the F_1 *Score*. Efficiency can be measured using *event throughput* which represents the amount of processed events per a unit of time in the IoT middleware layer from the sensors to the applications.

Test events and subscriptions sets can be chosen based on the use cases. For example, in [22] we have synthesized a set of around $15,000$ events of up to 10 attribute-value pair per event, and around 100 approximate subscriptions from real world smart city deployments in Europe such as the SmartSantander project [38] which employs a set of sensors to monitor temperature, noise, traffic, parking, etc.

Figure 2 illustrates the resulting effectiveness and efficiency of the approximate matcher working with a Wikipedia-based *esa*. Each cell in the figure shows the result that corresponds to a combination of numbers of thematic tags associated with events (the X-axis), and subscriptions (the Y-axis).

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 less terms to describe events, around $2-7$, and more to describe subscriptions, around $2-15$, can achieve a good matching quality, up to 85%, and throughput, up to 800 events/sec, together with lower error rates. That is concentrated in the middle left part of squares in Figure 2 (more red cells). The 100 approximate subscriptions would *cost* users an equivalence of around $48,000$ exact subscription rules. More details on the approach can be found in [22].

8. BUILDING IOT EVENT SYSTEMS

In [23] an architecture for IoT is presented based on the idea of thingsonomies and thematic event processing as shown in Figure 4. The main steps to build the IoT thingsonomies according to this proposal are:

1. Build a distributional semantic model which enables the system to automatically establish relationships between various terms such as *'water'* vs. *'fluid'*.
2. Use a semantic relatedness measure based on the built semantic model through a conventional interface such as REST and JSON [4].
3. Publishers annotate their events with a set of thematic tags at the data collectors.
4. Subscribers annotate subscriptions with thematic tags.
5. The event engine normalizes events and matches them to suitable subscriptions.
6. The event engine returns events matching a subscription to the subscriber.

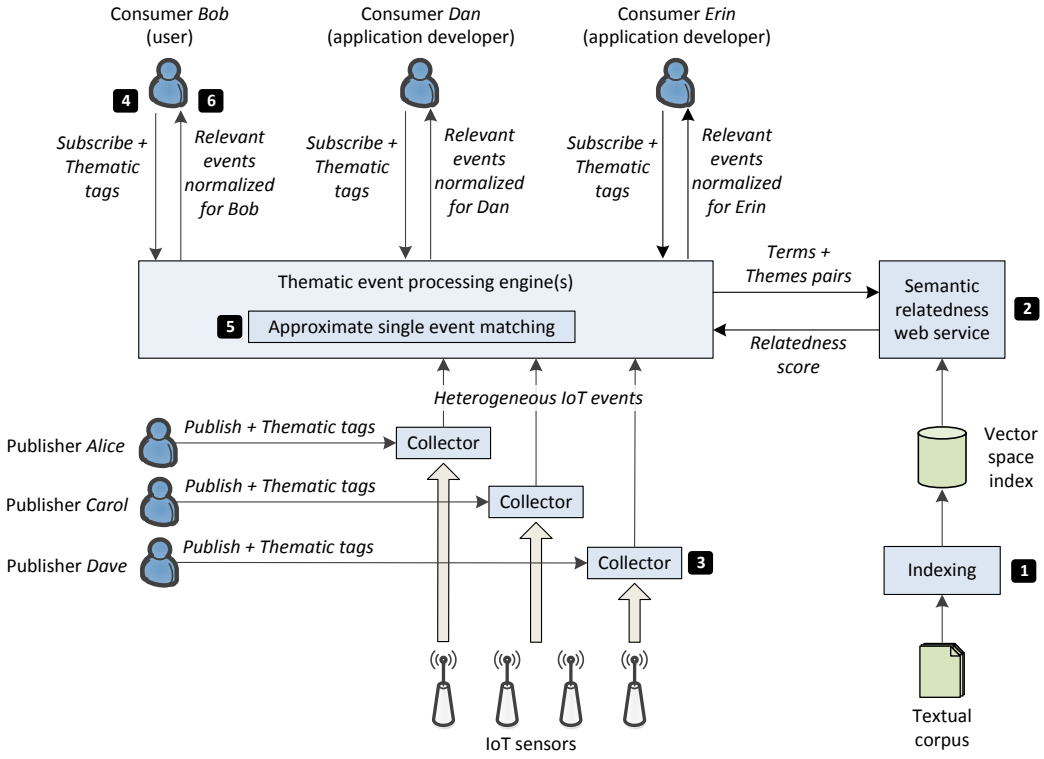

Figure 4: An IoT architecture based on thingsonomies and thematic event processing [23].

9. SUMMARY AND FUTURE WORK

This paper has analyzed the problem of semantic variety in event processing systems at large scales such as in the IoT. It has shown how the symbolic semantics used for events and subscriptions create a coupling of event producers and consumers and thus limits scalability. To loosen semantic coupling, we analyzed the suitability of vector space models of semantics within approximate and thematic event processing. These models are the basis for an IoT architecture that tackles variety using the resulting concept of thingsonomies.

Future research challenges include the investigation of more accurate approximations of meaning spaces, using thematic projection and variations of it. Research into optimization and indexing techniques of the proposed models are important future work. The extension of the discourse in this paper from semantic coupling into coupling by context can further improve the decoupling of event systems. Propagation of uncertainty values from single event matching into complex event patterns is also a future direction for research.

10. ACKNOWLEDGMENTS

The research leading to these results has received funding under the European Commission's Seventh Framework Programme from ICT grant agreement no. 619660 (WATERNOMICS). It is supported in part by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

[1] C. C. Aggarwal, N. Ashish, and A. Sheth. The internet of things: A survey from the data-centric perspective. In *Managing and mining sensor data*, pages 383–428. Springer, 2013.

[2] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[3] P. R. Carlile. Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organization science*, 15(5):555–568, 2004.

[4] D. Carvalho, C. Calli, A. Freitas, and E. Curry. EasyESA: A low-effort infrastructure for explicit semantic analysis (demonstration paper). In *Proceedings of the 13th International Semantic Web Conference (ISWC 2014)*, pages 177–180, 2014.

[5] A. Carzaniga, D. S. Rosenblum, and A. 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.

[6] D. Chandler. *Semiotics: the basics*. Routledge, 2007.

[7] G. Cheng, W. Ge, H. Wu, and Y. Qu. Searching semantic web objects based on class hierarchies. In *LDOW*, 2008.

[8] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, June 1970.

[9] G. Cugola and A. Margara. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, 44(3):15:1–15:62, June 2012.

[10] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JAsIs*, 41(6):391–407, 1990.

[11] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys (CSUR)*, 35(2):114–131, 2003.

[12] A. Ferrara, D. Lorusso, S. Montanelli, and G. Varese. Towards a benchmark for instance matching. In *International Workshop on Ontology Matching, The 7th International Semantic Web Conference*, page 37, 2008.

[13] A. Freitas, J. G. Oliveira, S. O'Riain, E. Curry, and J. C. P. 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.

[14] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[15] A. Gal. Uncertain schema matching. *Synthesis Lectures on Data Management*, 3(1):1–97, 2011.

[16] P. Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004.

[17] Y. Genovese and S. Prentice. Pattern-based strategy: Getting value from big data. *Gartner Special Report (June 2011)*, 2011.

[18] M. Gupta, R. Li, Z. Yin, and J. Han. Survey on social tagging techniques. *ACM SIGKDD Explorations Newsletter*, 12(1):58–72, 2010.

[19] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346, 1990.

[20] Z. S. Harris. Distributional structure. *Word*, 10:146–162, 1954.

[21] S. Hasan and E. Curry. Approximate semantic matching of events for the internet of things. *ACM Trans. Internet Technol.*, 14(1):2:1–2:23, Aug. 2014.

[22] S. Hasan and E. Curry. Thematic event processing. In *Proceedings of the 15th International Middleware Conference*, Middleware '14, pages 109–120, New York, NY, USA, 2014. ACM.

[23] S. Hasan and E. Curry. Thingsonomy: Tackling Variety in Internet of Things Events. *IEEE Internet Computing*, 19(2):10–18, 2015.

[24] S. Hasan, S. O'Riain, and E. 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.

[25] A. Hinze, K. Sachs, and A. 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.

[26] I. T. U. (ITU). World Telecommunication/ICT Indicators database, 2014, Last accessed January 2015.

[27] D. 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.

[28] A. Lenci. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31, 2008.

[29] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[30] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[31] G. Mühl, L. Fiege, and P. Pietzuch. *Distributed event-based systems*, volume 1. Springer, 2006.

[32] Organization for Economic Co-operation and Development (OECD). Machine-to-Machine Communications: Connecting Billions of Devices. 2012.

[33] N. W. Paton and O. Díaz. Active database systems. *ACM Comput. Surv.*, 31(1):63–103, Mar. 1999.

[34] M. Petrovic, I. Burcea, and H.-A. 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.

[35] M. J. Reddy. The conduit metaphor: A case of frame conflict in our language about language. *Metaphor and thought*, 2:164–201, 1979.

[36] J. Ross, L. Irani, M. Silberman, A. Zaldivar, and B. 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.

[37] D. E. Rumelhart, J. L. McClelland, P. R. Group, et al. Parallel distributed processing. *Explorations in the microstructure of cognition*, 2:216–271, 1986.

[38] L. Sanchez, J. A. Galache, V. Gutierrez, J. Hernandez, J. Bernat, A. Gluhak, and T. Garcia. Smartsantander: The meeting point between future internet research and experimentation and the smart cities. In *Future Network & Mobile Summit, 2011*, pages 1–8. IEEE, 2011.

[39] C. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[40] L. Von Bertalanffy. General system theory. *General systems*, 1(1), 1956.

[41] L. Zeng and H. Lei. A semantic publish/subscribe system. In *E-Commerce Technology for Dynamic E-Business, 2004. IEEE International Conference on*, pages 32–39, Sept 2004.