Research Article

Semantical Markov Logic Network for Distributed Reasoning in Cyber-Physical Systems

Abdul-Wahid Mohammed,^{1,2} Yang Xu,¹ Ming Liu,¹ and Haixiao Hu¹

¹*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*

²*School of Engineering, University for Development Studies, Tamale, Ghana*

Correspondence should be addressed to Yang Xu; xuyang@uestc.edu.cn

Received 28 May 2016; Accepted 27 November 2016; Published 24 January 2017

Academic Editor: José A. Somolinos

Copyright © 2017 Abdul-Wahid Mohammed et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The challenges associated with developing accurate models for cyber-physical systems are attributable to the intrinsic concurrent and heterogeneous computations of these systems. Even though reasoning based on interconnected domain specific ontologies shows promise in enhancing modularity and joint functionality modelling, it has become necessary to build interoperable cyber-physical systems due to the growing pervasiveness of these systems. In this paper, we propose a semantically oriented distributed reasoning architecture for cyber-physical systems. This model accomplishes reasoning through a combination of heterogeneous models of computation. Using the flexibility of semantic agents as a formal representation for heterogeneous computational platforms, we define autonomous and intelligent agent-based reasoning procedure for distributed cyber-physical systems. Sensor networks underpin the semantic capabilities of this architecture, and semantic reasoning based on Markov logic networks is adopted to address uncertainty in modelling. To illustrate feasibility of this approach, we present a Markov logic based semantic event model for cyber-physical systems and discuss a case study of event handling and processing in a smart home.

1. Introduction

Cyber-physical systems (CPSs) [1–3] represent a novel research field with high prospects for providing synergy between the digital and physical worlds. These systems consist of interconnected components, which collaboratively execute tasks in order to bridge the gap between the digital and physical worlds [4]. With the growing complexity of tasks due to the combination of CPSs and Internet of Things (IoT) [5, 6] however, CPSs have become distributed, and it is necessary developing interoperable CPSs capable of enabling timely delivery of services. In this way, CPSs become real-time distributed with multiscale dynamics and networking for efficient, dependable, safe, and secure management of monitoring and control of objects in the physical domain [7].

Following the integration of CPSs and IoT, innovative concepts and approaches, such as service-oriented architecture (SOA) [8, 9], collaborative systems [10], and cloud computing [11], have become apparent in the development

of CPSs. As devices in CPSs are required to interoperate at both cyber and physical scales, service-oriented CPSs [12, 13] so far look promising. By coupling the need for CPSs to interact with real world objects in real-time with optimal resource consumption however, using only the service-oriented approach is not enough to realise comprehensive distributed real-time CPSs that take into account the strong interdependencies between the cyber and physical components. Therefore, context-aware agent-based approach is more appealing since diverse attributes can be encapsulated within agents, and distributed pervasive computing with better coordination and interoperability among autonomous and heterogeneous agents is achievable. Essentially, context-awareness is indispensable in CPSs since sensing, resource discovery, adaptation, and augmentation are the key drivers of this novel technology [14, 15]. Additionally, with the changing dynamics of distributed CPSs domains, it is natural that partial observability is inherent in CPSs [16, 17]. Using ontology as underlying semantic technology therefore

requires uncertainty modelling techniques towards good model performance.

This paper proposes a context-based multiagent architecture for distributed reasoning in CPSs. Sensor networks in distributed physical environments provide the semantic capabilities of this model, and semantically annotated low-level contextual information merges with domain knowledge in a reasoning engine. Derived implicit knowledge through semantic reasoning and together with the annotated data enables distributed software agents operating in the cyberspace to provide decision support for actuation information. Each agent is capable of interacting with the physical environment and sharing some principal commonalities with the other agents. As such, the agents are capable of exposing, consuming, and even at times processing collaborative services targeting laid down system goals. To incorporate uncertainty in modelling, semantic reasoning on this model incorporates the inferential power of Markov logic networks (MLN) [18] into event recognition to reduce inferential and computational loads. Finally, we discuss a case study of a smart home as a CPS, and results of our experiments show feasibility of this approach in modelling concurrent events in CPSs.

In summary, we describe in this paper our initial work in CPSs, which overcomes some of the major limitations in this research field. We provide four main contributions. First, we propose a multiagent architecture that can bridge the gap between the operations of the cyber and physical components of CPSs. Second, we describe a procedure that can be used to dynamically compose high-level system goals and underlying criteria from low-level contextual information. Third, we introduce a smart home ontology, which incorporates human actors in the physical space of CPSs as computing entities. Finally, we present a methodology based on MLN for event recognition in CPSs.

The rest of the paper is organised as follows: we present the state of the art and the study background and other preliminary information in Sections 2 and 3, respectively; Section 4 gives an agent model for CPSs, followed by our framework for distributed reasoning in CPSs in Section 5; uncertainty-based event recognition in CPSs using MLN is presented in Section 6; experiments and discussions of results are presented in Section 7; and finally, we conclude and propose future research in Section 8.

2. Related Work

A coherent distributed reasoning architecture is one that achieves an interoperable CPS to cope with the requirements of the physical and cyber components. Because of the lack of sound theoretical foundation for CPSs currently [19], most approaches successfully model either the physical component [20] or the cyber component [21], but not both. Salient studies towards comprehensive models for CPSs that take into account both the cyber and physical components include [8, 22, 23], which use service-oriented computing to achieve interoperable CPSs. Service-oriented approach alone, however, is not suitable for modelling real-time distributed CPSs

with multiscale dynamics and networking. In this regard, an agent-based modelling is more appropriate for distributed complex systems such as CPSs [24, 25].

Because of the underlying sensor network in CPSs, semantic agent technologies are closely associated with our approach. As provided in [26], a semantic agent technology has been used to describe a battlefield information system, which uses information fusion processes to dynamically integrate sensor networks towards real-time context-based reasoning. To enable scalable sensor information access, an architecture and programming model for service-oriented sensor information platform has also been proposed [27]. This approach leverages an ontological abstraction of information to optimise use of resources in collecting, storing, and processing data. Timeliness and concurrency in distributed processing environments can also be enhanced using autonomous software agents. As such, use of autonomous semantic agents as a new software paradigm for distributed computing environments has been proposed [28].

Obviously, the complex dynamics of CPSs, coupled with the need to properly represent embedded computing and communication capabilities, motivate the use of semantics and distributed agents towards interoperable CPSs. As can be found in [29], a multiagent model for CPSs in which a distributed semantic agent model augments data acquisition process with ontological intelligence has been proposed. This model, however, provides no procedure for reasoning locally about individual components and globally about system-wide properties. But such semantics, for instance, event recognition, is very critical for distributed real-time CPSs and can essentially specify components of systems in terms of interfaces and observations [7, 30]. Additionally, ontology forming the underlying layer in the semantic agent-based model primarily supports certainty-based reasoning and as such requires techniques to address uncertainty in modelling.

In our approach therefore, we augment the semantic multiagent architecture with a robust reasoning mechanism, which can support both certainty and uncertainty-based reasoning. To promote concurrency and timeliness in the operations of CPSs, MLN is adopted as an uncertainty modelling framework and can compactly represent heterogeneous computations using a common set of rules. Essentially, this achieves a reasoning procedure for CPSs by leveraging advantages of both ontology and probabilistic graphical models to model both complexity and uncertainty, which reduces limitations of both the ontology and probabilistic graphical models.

3. Background and Preliminaries

We provide in this section aspects of the problem of real-time distributed reasoning in CPSs, Markov logic, and smart home as a case study of a CPS.

3.1. Problem Description. The growing pervasiveness of CPSs further keeps the cyber and physical components apart, and separately managing these components would not allow us to realise fully the benefits of these systems. Appropriate

techniques that would allow interoperability between these components are essential so that monitoring and actuation can be invoked remotely. In this regard, standardised interfaces that can achieve interoperable CPSs are desirable and should be guided by the following:

- (1) Defining an architectural framework that supports interoperability and distributed reasoning in CPSs.
- (2) Applying semantics to explicitly represent contextual information and providing an efficient data storage mechanism.
- (3) Providing distributed reasoning procedure that creates more autonomy and intelligence in operations of CPSs.
- (4) Incorporating uncertainty into modelling.

Following above challenges and the ability of semantic agents to be discoverable and autonomous, we pursue semantically oriented techniques, in which ontological intelligence is used to address the problem of uncertainty-based real-time distributed computing in CPSs.

3.2. Markov Logic Network. MLN [18] is an interface layer in artificial intelligence, which defines a first-order knowledge base in terms of first-order logic formulae and associated weights. Given a set of constants depicting objects of a domain, MLN defines a ground Markov network, which represents a probability distribution over possible worlds. Each world, basically, represents assignment of truth values to all ground atoms, and this distribution is a log-linear model given by

$$P(x) = \frac{1}{Z} \exp \left\{ \sum_i w_i n_i(x) \right\} = \frac{1}{Z} \prod_i \phi_i(x_{\{i\}})^{n_i(x)}, \quad (1)$$

where $n_i(x)$ is the amount of true grounding of first-order formula F_i in x , $x_{\{i\}}$ is the i th state of the predicates appearing in each formula, w_i is the weight of F_i , $\phi_i(x_{\{i\}}) = e^{w_i}$ is the potential function of each clique in the ground Markov network, and Z is a normalizing constant called the *partition function*. Each weight indicates the strength of a constraint that a formula represents and is directly proportional to the difference in the log probability between a world that satisfies the formula and one that does not.

Due to the varying number of constants that can represent the same knowledge base either in part or full, MLN allows the same formulae to be applicable under all circumstances and can be viewed as a template for constructing Markov networks. In this way, different sets of constants can produce different ground Markov networks using a common underlying MLN. This, ideally, is suitable for domains, such as CPSs, where the task of reasoning requires combining separate reasoning chunks, which need to be processed independently.

3.3. Case Study. As intelligence in the home gets more sophisticated, intelligent interconnection of distributed consumer hardware such as consoles, smart home servers, and smart phones running diverse functionalities like assistive

health care and home automation constitutes a CPS. To demonstrate the heterogeneity, concurrency, and sensitivity to timing of CPSs, a case study of temperature event recognition is considered. This presents a scenario of using an ontology-based model to achieve an interoperable CPS that leverages a common event recognition model across different layers. Specifically, using a single computation of a car system, events pertaining to temperature conditions of the car user's home and that of the car engine can be achieved. Typically, this computational platform must support concurrent processing of events since temperature events for the home and the car can concur.

The fact that CPSs need to be sensitive is a challenging task in respect of false positives that can arise if uncertainty is not well managed. Apart from noisy sensor information and incomplete domain knowledge being primary sources of uncertainty, environmental factors are also potential sources of uncertainty that cannot be ignored in CPSs. For instance, a temperature event that considers optimal resource consumption in a smart home may trigger opening of windows on a cool sunny day for comfort of the home. This strategy, though, all things being equal, sounds ideal, but environmental factors such as air population and external noise can present a trade-off between comfortability and minimising resource consumption in the home. As such, the effectiveness of CPSs will be much appreciated if uncertainty, which is unavoidable in nature, is well managed in these systems.

4. Agent Network for CPSs

Building on the foundation of mobile agent network [31], we define a multiagent system residing in the cyberspace of CPSs. This model considers all aspects of agents' communication and operation including issues relating to performance of multiagent systems and CPSs.

4.1. Cyber Agent Model. A cyber agent model is defined by a triple $CAM = \langle A, D, N \rangle$, where A represents a community of agents depicting distributed computational environments in a CPS, D denotes specific domains of agents' services, and N defines networks of agents operating in the cyberspace. Essentially, agents in this definition can be stationary and mobile so as to suit the changing dynamics of CPSs domains. In this way, an agent A_i in a multiagent system, which is defined by $A = (A_1, \dots, A_n)$, represents a specific computational platform and can perform tasks allowed by its domain. This specifies somewhat autonomy in the operations of these agents, and tasks can be encapsulated as agents' capability from the viewpoint of functionality and performance. As such, interactions between agents provide the needed communication and cooperation to bridge the gap between the operations of the cyber and physical components of CPSs.

Contextual reasoning paradigm [32] in which each agent depends on a domain specific ontology and can link with other agents through semantic mapping makes this design distributed. Since each computational platform provides a functionality for service execution, combining these multiple

FIGURE 1: Service assignment graph for agents' domains.

ontologies through semantic mapping allows us to define joint functionalities that can be used in complex task operations. For a nonempty set I of indices used to identify agents associated with domain specific ontologies $\{D_i\}_{i \in I}$, we define the joint functionality of our multiagent system as a set of cross-layer services $JS = \{S_1, \dots, S_n\}$. Thus, a set of services indexed by each domain is defined as $\{\mathcal{S}_i\}_{i \in I}$. Intuitively, $\mathcal{S}_i$ represents a service formalisation of the i th ontology.

We must recognise that each service S_i can be provided by one or more agents. To avoid conflicts in accessing services, an agent definition that explicitly specifies computational platforms and services provided is critical. In this way, we make services distinct by encapsulating each agent definition as a property of a computational platform in a CPS using the triple $agent_i = \langle name_i, deployment_i, service_i \rangle$. As we can see, this definition of an agent, apart from a property describing a given computation, also provides information about agents' services and domains for those services. A service domain is specified by a deployment property, which can be a physical address of a distributed environment in CPSs. For instance, given a set of agents' services domains $D = \{D_1, \dots, D_n\}$, the set of services provided by agent $agent_i$ can be described by $S_i = \{S_{i1}, \dots, S_{in}\}$, and each service within this domain can be invoked using the pair $address_i = \langle S_i, agent_i \rangle$. This means, within the cyberspace, the interactions between these agents define an undirected graph $N = (D, E)$, where E denotes an edge between any two domains of agents with overlapping functionalities.

4.2. Agent Cooperation in CPSs. For the multiagent system to ensure high fidelity between the physical and cyber components, each agent domain must trigger requests, which represent implicit knowledge inferred from the domain's low-level contextual information. Typically, these requests may involve complex tasks that can exceed capacity of a single operational domain and may require cross-domain services aggregation. In this case, agents operating in the cyberspace, through their interactions, can communicate and cooperatively assume specific roles to execute tasks.

The idea that agents' domains are distributed and can form a graph with overlapping functionality presents a complex network in which fast information sharing becomes necessary for large agent teams [33]. Relying on information importance as a determinant for service assignment based on performance requirements of agents' domains is ideal for fast information sharing and parallel execution of services. In

this regard, service requests to agents can be either domain specific or across multiple domains. As shown in Figure 1, a special case in the assignment of services to agents is, for example, *Request1*, when a request is domain specific. In this case, all services can be deemed mutually exclusive, and communication and coordination among agents become less important. However, when services overlap or a given request requires a combination of cross-domain services, we face a problem of a multiagent autonomy denoted by *Request2*. But interestingly, instead of employing a planning agent for the assignment of services in this case, agents can combine their inherent intelligence with domain knowledge to negotiate for services in our design.

5. Distributed Reasoning in CPSs

Distributed computing systems in CPSs can employ pervasive computing techniques to provide autonomous, interoperable, computational elements that can be described, discovered, and orchestrated within and across different layers [15]. In this regard, ontology-oriented modelling of contexts offers a lot of advantages [34]. We present next a semantic agent-based architecture for CPSs. Additionally, semantics for formulation of high-level complex tasks with underlying criteria from low-level contextual information is discussed.

5.1. Semantic Multiagent Architecture for CPSs. This is a broker-centric multiagent architecture, which can support cross-layer service collaboration in CPSs. As shown in Figure 2, this architecture captures into perspective the modelling concerns of CPSs raised in [35]. Key components of this architecture are *data management module*, *context ontology module*, *semantic reasoning engine*, and a *confederation of semantic agents*. Detailed descriptions of these components are provided in the following subsections.

(1) Data Management Module. The key functionalities of this module are collection and transmission of data to storage areas. Raw context data are acquired from distributed sensor networks in the physical domain, and the heterogeneity of these data requires *semantic markups* that applications can easily understand. Through semantic annotation, the context data are transformed into semantic markups that can link to external definitions through unique URIs of ontology instances. For example, the semantic annotation of a temperature sensor data can be as in Listing 1.


```

<Device rdf: ID="&obs; TempSensor2">
<owl: sameAs rdf: resource = "&smh; TempSensorR2"/>
<smh: observedPhenomenom rdf: resource = "&smh; Temperature"/>
<smh: qtyValue rdf: datatype = "&xsd; float">24.5</smh: qtyValue>
<smh: qtyUnit rdf: resource = "&smh; Celsius"/>
<smh: timeStamp>2015-07-28 16:52:30</timeStamp>
</Device>

```

LISTING 1

FIGURE 2: Multiagent context architecture for CPSs.

As can be seen clearly, each annotation contains a unique URI, such as *smh*, *TempSensor2*, of an ontological instance, description of the observed phenomenon, measured value of the phenomenon, unit of measurement used, and the timestamp for the observation. Apart from the timestamp and measurement value, which are XML Schema datatypes, the other attributes are resources which exist in an external definition.

Data storage in this architecture is achieved at two levels using different databases that must interoperate with each other. In the first instance, the raw sensor data is stored in such a way that it can be efficiently maintained and exported by disparate sources. Secondly, after the raw sensor data is semantically annotated, a storage mechanism that supports semantics is required for storing this annotated data. Thus, in our architecture, the annotated data are stored in a repository

as ontology instances and associated properties that machines can easily interpret. This repository is updated whenever a new context event occurs and can be augmented with Linked Data techniques to support semantic data integration at the instance level.

In Linked Data research [36, 37], dereferencing the URIs of resources through HTTP protocol can be exploited to incrementally obtain the description of resources. To further support integration of semantic data distributively, OWL axiom *owl:sameAs* has been used successfully at the instance level in Linked Data research. With such success in a distributed setting, this same technique can be adopted into CPSs for semantic integration. Thus, semantic markups in this paper are linked to external definitions using the *owl:sameAs* axiom. As can be seen in the example above, the use of this axiom shows that the two URIs refer to the same instance, thereby providing a mapping between the semantic repository and the context ontology repository.

(2) *Context Ontology Module*. Ontology modelling and processing occur in this module. In CPSs, computing entities and services of distributed intelligent environments can be grouped together, forming service-oriented ecosystems. Contexts in these domains can share some concepts in common, even though their detailed properties can differ significantly. Instead of completely modelling all contexts across different domains, the objective here is to model contexts using a *base ontology* and a *domain specific ontology*. Entities of the base ontology are extensible basic common concepts across different environments. The domain specific ontology, however, represents only those concepts that uniquely exist in each domain.

Specifically, in a smart home domain, the most fundamental concepts we have identified as extensible nodes of the base ontology are *user*, *deployment*, *service*, and *computing entity*. When these entities are linked together, a skeleton of contextual entity is formed, which allows context-based data acquisition. Figure 3 shows the context ontology model we propose for a smart home domain in this paper. The base ontology, which is extended by both a smart home and a smart institution, illustrates the advantage of knowledge reuse using ontology. In both cases, base entities such as Room and AdhocService are extended to meet specifications of the application domain. For instance, whilst we can specify bedroom and living room in a home, an institution can have rooms such as lecture room and conference room.

It is important to note that human as a computing entity in this model is a novel contribution that is ideal for design of CPSs. This essentially extends the service-oriented paradigm to incorporate human services in CPSs towards transmuting system components and behavioural practices [38]. Specially with our design, the role of humans as both actors and sources of contextual information in the physical domain can be explicitly represented and allows social awareness to be incorporated into CPSs. In this view, CPSs are well positioned to provide emotional intelligence [39] that will respond appropriately to people and situations.

```
<ReasonedData rdf: ID="&obs; TempSensor2">
<smh: domainURI rdf: resource = "&smh; room104"/>
<smh: criticalLevel rdf: resource = "smh; High"/>
<smh: alert>Fire in building</smh: qtyValue>
</ReasonedData>
```

LISTING 2

(3) *Semantic Reasoning Engine*. This is the central component of our design in which high-level implicit knowledge can be inferred from sensed contextual information. Semantically annotated data and the context ontology are aggregated into a coherent model that semantic agents and physical objects can share. Both certainty-based and uncertainty models can be supported by this design. But the focus of this research is uncertain decision support in CPSs. Specifically, uncertainty-based reasoning about resources and events and dynamic formation of collaborative cross-layer services given high-level system goals with underlying criteria are the focus of this design.

It is worth noting that putting the reasoned data to use by the semantic agents requires a data structure that can easily integrate with the underlying semantic repository of this representation. Specifically, the semantics of the reasoned data, when used to feed these agents, must specify among other needs, the referenced domain of the inferred knowledge. The argument here is that since CPSs domains are highly distributed in nature, agents can cooperate effectively across different domains to achieve better computational intelligence if we semantically specify domains of inferred knowledge in the reasoned data. In line with this paper's objective, such a data structure can allow easy mashup of resources to solve complex problems. Thus, Listing 2 is an example of a semantic markup of a reasoned data.

As we can see, this example basically demonstrates that aside the domain of interest referenced using *domain-URI*, other elements fitting a given scenario are allowed. Among these additions that is also unavoidable is the high-level knowledge obtained through semantic reasoning. This knowledge in this example is specified using the element *alert* and mostly what users get as prompts. Obviously, such a data structure combined with the domain knowledge can allow the semantic agents gain enhanced computing capabilities.

(4) *Semantic Agents*. The semantic agents are distributed algorithms executing on multiple distributed computing entities in the physical space. To provide decision support for actuation information, these agents merge semantically annotated data with the reasoned data. By describing these agents as a community, they are ostensibly the control point of this architecture and can advertise their services in the reasoner through interactions and semantic reasoning. Thus, each agent's behaviour is well suited for its environment, and such behaviours are well suited for resource discovery through semantic reasoning.

FIGURE 3: Smart home context model for CPSs.

5.2. Dynamic Composition of High-Level Complex Tasks. Facilitation of dynamic formation of collaborative services towards execution of complex tasks requires elicitation of high-level complex services from low-level information. This process can guarantee better quality CPSs and overcomes common engineering design flaws to provide right actuation information for the needs of physical objects. For instance, through low-level information, we can compose a task, such as *put off the fire*, as an event for handling fire outbreak in a CPS environment. However, this particular task, unlike some tasks, requires complex functionality and needs to be decomposed into primitive level tasks, which can then be serviced by specific resources. This is a challenging process

and therefore requires a dedicated framework on how to figure out complex functionality from low-level contexts.

As shown in Figure 4, this approach is motivated by the established actuation relationship between the cyberspace and distributed CPS environments. For clarity of representation, the physical environment is categorised into *usage* and *context* environments. The usage environment describes processes performed by semantic agents, and how systems can achieve tasks in the environment. Because each agent's behaviour best suits its environment, the usage environment ostensibly contains specific objectives, which describe the activities of agents towards useful output. The context environment, as specified by the context acquisition process, is the

FIGURE 4: The relationships between a distributed CPS environment and the cyberspace.

source of domain knowledge, which underpins the semantic capabilities of this approach.

In the distributed setting of CPSs, it is required that the cyberspace provides specifications that can address requirements of the physical environment. We can see from Figure 2 that these requirements in the cyberspace are models and processes that control objects in the physical environment and vice versa. This brings to the fore issues about *domain-driven* and *user-driven* requirements. Since the domain-driven requirements, which fundamentally hold the underlying semantics of this approach, have been discussed in the previous section, our focus now is the user-driven requirements. These requirements are enshrined in the actuation information and form the underlying idea towards the elicitation of high-level complex services from the domain-driven requirements. Therefore, understanding the relationship that the actuation information establishes between the cyberspace and the physical environment is essential towards mapping contextual information to high-level complex services.

High-level composite context can be derived from low-level contextual information through the semantic reasoner of Figure 2. As shown in Figure 5, the logic flow of the reasoning engine of this architecture consists of three main functional blocks: *models*; *filter*; and *composer*. Reasoned data from reasoning models are passed through a filtering process in which statements are categorised based on a predefined set of rules. All statements through this *filter* are either categorised as *executable* or *nonexecutable* but not both. An executable statement represents a single service phenomenon, such as *high temperature*, which can directly be executed either remotely or centrally by reducing the home's temperature through an air-conditioner. But when a statement requires aggregation of services in order to achieve its objective, it is filtered to be nonexecutable. An example of

FIGURE 5: Logic flow of the semantic reasoner of Figure 2.

Input: in
Output: out
 (1) **if** *executable* **then**
 (2) *out* $\leftarrow$ *statement*
 (3) **else**
 (4) *in* $\leftarrow$ *PlanningProblem(statement)*
 (5) goto step 1
 (6) **end if**
 (7) **return** out

ALGORITHM 1: Algorithm for composition and processing of composite context.

a nonexecutable statement is when sensors detect smoke and high temperature in a building, and the reasoned information is *fire outbreak in building*. Obviously, an appropriate action in this case is to put off the fire, which requires different services wrapped as capabilities of different physical objects in this approach. In this regard, this reasoned information requires further processing in the form of composition of appropriate services and constraints so that tasks can be appropriately scheduled among computing entities.

Whilst the executable statements directly feed the semantic agents, the nonexecutable statements are transformed into high-level composite contexts for further processing through the composer. Following the objective of this paper, these statements are composed into high-level tasks with underlying criteria. Specifically, this stage generates an HTN planning problem, which is passed as an input to the reasoner for automatic composition of collaborative services based on Algorithm 1. As we can see, the filtering process creates a cycle of execution of information whenever a nonexecutable statement is encountered. From lines 4 and 5, a nonexecutable statement results in a planning problem consisting of tasks and constraints. Generation of the planning problem is an instance of Algorithm 2, which takes a nonexecutable statement as an input. With this transformation, collaborative services can then be formed using a specified model and can directly feed the semantic agents. To support the needs of time-criticality in CPSs, this design promotes efficient use of computational resources by ensuring that all nonexecutable statements experience one-off processing. In essence, the

ALGORITHM 2: Planning problem.

set of rules provided in the model block ensures that all composite tasks are reduced to primitive tasks, which can be directly executed.

6. Event Recognition Using Markov Logic

Context-based events are central in initiating activities in CPSs and naturally specify real-time demand responsiveness of systems' components in terms of interfaces and observations [39]. From the viewpoint of our multiagent architecture, events can stimulate services of one or more agents in the network, and it is therefore important to detect events of predefined operations that are desirable to both systems and users of CPSs.

An event ontology is required to augment the proposed context ontology in the previous section towards event-based reasoning. However, ontology in its classical form currently cannot represent and reason under uncertainty. In view of good modelling practice towards best performance of CPSs, we adopt MLN based event recognition to address uncertainty whilst keeping the structure of the underlying ontology intact. This exploits the view of MLN as a template for Markov networks so that only a part of OWL rules applicable to events are considered in the model construction. One obvious advantage is in the compact representation of model complexity, which can guarantee incorporation of rich domain knowledge for high sensitivity and good concurrent processing of events.

MLN allows existing knowledge bases in first-order logic to incorporate uncertainty in knowledge representation by adding weights to logic formulae. Since first-order logic underlines the fundamental theory of ontology, OWL rules for knowledge discovery can therefore be transformed into MLN weighted formulae towards event recognition.

6.1. Rules for Event Recognition. OWL rules form the underlying logical framework of our event recognition process, and the semantics we provide holistically capture the domain's interest phenomena in the rules. As shown in Figure 6, an event is described by a tuple $\langle \text{component}, \text{function} \rangle$, which denotes *event components* and *event semantic functions* [40]. We use event components to compose heterogeneous sensor data to form contextual information that match events' requirements. Thus, $\text{component}_i = \{o_{i1}, \dots, o_{in}\}$, where o_{ij} represents observations driving the occurrence of an event. To be able to propagate logical constraints of events in rules, we use semantics of event functions, $\langle \text{Stop}, \text{Change}, \text{Comparison} \rangle$, to distinguish between categories of events. With this specification, Stop is a predicate

FIGURE 6: Context-based event model in CPSs.

we use to express action of an event that changes spontaneous states of natural phenomena. Thus, the predicate, $\text{Stop}(x, y)$, defined by

$$\text{Active}(y) \wedge (\text{Action}(x, y) \vee \text{Affect}(x, y)) \implies \neg \text{Active}(y) \quad (2)$$

indicates that the state y changes when acted upon by an action of event x . In some cases of CPSs, the event may have effects on the state, as $\text{Affect}(x, y)$ indicates. In a fire scenario, for example, this predicate ensures the right event invocation that will put off the fire. We must note that the same predicate cannot be applied to routine events such as putting off the air-conditioner. Unlike in the fire case whereby the new state of affairs after putting off the fire may be perpetual, devices are only eligible for temporal state changes. As such, we use the unary predicate $\text{Change}(x)$ as the event function for achieving temporal state changes for events in CPSs. $\text{Change}(x)$ is defined as

$$\exists y \quad (x = y \implies x \neq y), \quad (3)$$

where state x changes to new state y using three subfunctions. Specifically, a state change can be rise in degree of something using the predicate $\text{Increase}(x)$; reduction in degree of something using predicate $\text{Decrease}(x)$; and toggling between on and off modes of devices using the predicate $\text{Switch}(x)$. All these subfunctions operate by conditioning the current state against the new state. For instance, to increase the temperature of a room using an air-conditioner, the semantics of the predicate $\text{Increase}(x)$ is defined as

$$\begin{aligned} \exists x \quad \text{Status}(x) \wedge \exists (x, y) \text{Value}(y) \implies \\ \exists (> x, y) \text{Value}(y) \end{aligned} \quad (4)$$

to indicate a change in status value of the air-conditioner when y is greater than x . Obviously, this predicate, like the other two, requires semantics that compares the current and new states in the change process.

The Comparison predicate is used to define conditions that describe state changes. We consider $\text{LessThan}(x, y)$, $\text{LessThanEqual}(x, y)$, $\text{GreaterThan}(x, y)$, $\text{GreaterThanEqual}(x, y)$, and $\text{Equal}(x, y)$ as the predicates for conditions for state changes. For example, the semantics for $\text{LessThan}(x, y)$ is defined as

$$\exists c, \quad \text{Cond}(c) \wedge \text{Value}(x) = c \wedge \text{Value}(y) < c \quad (5)$$

FIGURE 7: A section of ground Markov network constructed from the MLN for event recognition.

to express the condition that the value of x is c , and the value of x is less than the value of y . Therefore, event recognition based on component structures and semantic functions provide logical operations in formulae that form good basis for Markov logic based event recognition.

6.2. Translation of Rules into MLN. The first step towards conversion of OWL rules into MLN requires transformation of OWL rules into first-order logic formulae. As provided in [41], OWL classes and properties, respectively, represent *unary* and *binary* predicates in first-order logic and can be combined using logical connectives to form atomic formulae. For example, the first-order logic translation of a class Room is $Room(x)$, where x denotes instances of the given class. For a property hasDeployment, the equivalent first-order logic formula is

$$hasDeployment(x, y) \implies Device(x) \wedge Room(y), \quad \forall x, y, \quad (6)$$

where x and y , respectively, denote the domain and range classes of this property. Axiomatization of classes and property restrictions can also be translated into first-order logic. For instance, `rdfs:subClassOf` axiom can be translated into first-order logic as

$$Room(x) \implies Bedroom(x) \quad \forall x \quad (7)$$

to indicate that `bedRoom` is a subclass of `Room`. On this basis, we can use logical connectives to compose the first-order logic formula of the concept class `Device` and its property as

$$Device(x) \wedge hasDeployment(x, y) \implies Room(y) \quad \forall x, y. \quad (8)$$

Interestingly, the *Alchemy* [41] tool for MLN provides built-in functions that simplify the translation of logical conditions into MLN. For instance, the predicate *LessThan*(x, y) in MLN is simply represented using the internal predicate of *Alchemy* *lessThan*(*int*, *int*). Basically, this predicate tests if the first argument is less than the second argument.

Once we obtain the first-order logic translation of rules, the MLN is achieved by adding weights to each formula. The

MLN together with a set of constants define a ground Markov network on which probabilistic reasoning can be performed. Figure 7 shows a section of the ground Markov network based on the MLN of our case study for event recognition. As we can see in this figure, links exist between any two ground terms appearing together in the same formula in the MLN. Hence, given a MLN and a set of constants, arbitrary queries such as the conditional probability that a formula holds given another formula in the MLN can be addressed.

6.3. Fuzzy Markov Logic Network. We recognise that the axiomatic notion of probability as presented in the last subsection is incapable of dealing with vague information in knowledge. This becomes apparent in MLN when multivalued clauses are encountered, and this presents a challenge beyond the classical notion of MLN. In this view, we provide a fuzzy notion of Markov logic called *fuzzy MLN* in which inference to queries requires the inference machinery of fuzzy logic.

The basic idea serving as a point of departure in fuzzy MLN lies in the fact that a formula F in first-order logic can be viewed as a collection of elastic constraints, which restrict the weights W associated with each grounding of its terms. To achieve this, we define a fuzzy membership function in terms of weights and ground terms of MLN clauses and obtain an extension of MLN to fuzzy MLN as

$$F_{MLN} = (\mu : F \longrightarrow W). \quad (9)$$

This represents a mapping of a set F of grounded first-order logic formula into a set of MLN weights W . The idea that different constants refer to different objects in MLN and a formula can contain more than one ground clause allows for separate assignments of weights to each ground clause in MLN. Essentially, this achieves fuzzy membership functions mapping ground MLN clauses into an ordered set of fuzzy pairs from which MLN inferences can be performed. Clearly, this fuzzy set is completely determined by the set of tuples

$$F_c = \{(c, \mu(c)) \mid c \in C\} \quad (10)$$

denoting the assignment of weights to each grounding of F for a set of constants C .

As shown in Figure 8, the membership function in fuzzy MLN, assuming without loss of generality that all weights are

FIGURE 8: Membership function of fuzzy MLN.

positive, is a representation of the magnitude of participation of the weight of each ground term as an input. This associates different weights with the same formula for different ground terms and defines functional overlaps between these ground terms, which determines outcomes of rules. As we can see in this figure, a typical case of an overlap is the temperature value 26°C, which presents a case of a multivalued clause for different grounding of the same formula. This value exists in the interval of the *minimum criterion* of the two fuzzy sets defined by *cold* and *hot*

$$\mu_{cold \cap hot} = \min(\mu_{cold}, \mu_{hot}), \quad (11)$$

where μ_{cold} and μ_{hot} , respectively, define the membership functions of the states cold and hot. Intuitively, this can be described as to what degree a cold temperature 26°C is hot. Obviously, knowledge about true state of this temperature value smacks of vagueness and can be efficiently interpreted as a fuzzy constraint on a collection of ground terms.

In fuzzy MLN, unlike in classical fuzzy logic, the situation described above is easily handled by specifying ground clauses involving both cases in a training set. As Figure 9 depicts, this defines a ground Markov network in which the two outcomes are conditionally independent given the input. Each dotted circle in this figure indicates a weighted ground clause of the same formula, and either clause is a complement of the other, which in the absence of one clause defines the classical notion of MLN. This means learning the weight $w1$ as shown in Figure 8 in this particular case produces two values that define the fuzzy set for reasoning. Thus, fuzzy MLN leveraging these weights and together with a set of constants defines a ground Markov network, which can be reasoned upon using the inference machinery of fuzzy logic without employing any formal fuzzy logic semantics.

7. Results and Discussion

In this section, we present and discuss results of event recognition under uncertainty in a smart home as a CPS. The conditions for all experiments were designed to test performance of this approach using key intrinsic requirements of CPSs such as sensitivity to timing and concurrency. In this regard, thrust of our analyses bothers mainly about precision

as a measure of sensitivity to timing of occurrence of single and concurrent events in CPSs.

To incorporate uncertainty into modelling for our experiments, we considered an MLN based event model rooted in the OWL ontology of this paper's case study. This ontology captures into perspective semantics of our event model, and the key properties include

- (i) *hasDeployment*, which relates the concept *Device* to concepts *Room* and *Engine*,
- (ii) *hasValue*, which relates the concept *Device* to a data value,
- (iii) *hasOutput*, which relates a data value of a device to its semantic interpretation using the concept *Output*,
- (iv) *hasEvent*, which relates the concept *Output* to the concept *event*.

Essentially, *Room* and *Engine*, which are subclasses of *Location*, define heterogeneous computational platforms for devices using the deployment property. Consequently, events associated with different platforms denote effects of interpretations of devices' values on those platforms and are stored in the ontology as type *event*.

For compact modelling towards expedited processing of events, OWL rules provided partial specification of domain concepts relevant for the construction of the MLN event model. With the ability of OWL to support heterogeneous processes, we used the same set of rules to represent different computational platforms in the MLN. Specifically, computations in relation to a home's indoor comfort index and operational safety condition of a car's engine were considered to be two likely synchronous events that a single computation can represent. The ostensible need for a single computation for distributed environments in CPSs can be seen in a case of driving towards home whereby the computation of the car's console monitors both the home condition and the engine temperature of the car. In this way, a distributed sensor network of the home and the vehicle engine provides contextual information for the computational intelligence. With the deployment property of devices in the underlying ontology, contextual information can be accurately filtered and applied according to domain specifications.

As shown in Table 1, five events were defined to represent the two distributed environments considered in our experiments. Essentially, the heterogeneity in these events is enshrined in the different conditions pertaining to temperature measurements in these environments. For instance, whilst we can specify a normal room temperature to be in the range 21°C–27°C, a normal operating temperature range of an engine is 180°C–205°C, which is high temperature in a case of the home and way beyond limits of human survival temperature. Clearly, these two cases represent vagueness in knowledge as both denote a normal temperature, and it is therefore important disambiguating between heterogeneous sensed information in modelling. In this regard, our event model can be described as a composite model designed to precede any recognition process with precise knowledge discovery. Aside event recognition, this same model can be used to perform semantic reasoning towards mashup of

FIGURE 9: An example of ground Markov network of a fuzzy MLN.

TABLE 1: Event specification for home and engine domains.

Domain	Value range	Inferred knowledge	Event
Home	Less than 21°C	Low temperature	Increase temperature
	21°C–27°C	Normal temperature	No event
	Greater than 27°C	High temperature	Reduce temperature
Engine	180°C–205°C	Normal temperature	Normal operation
	Greater than 205°C	Overheat	Switch off engine

FIGURE 10: Precision of a single event recognition with increasing evidence.

resources. For instance, temperature sensors deployed at any of the two distributed environments in our case study can be inferred with this model.

We evaluated the performance of this model by considering both single event and multiple event recognition tasks. In either case, we varied the training set of the MLN from 100 constants to 1,500 constants. As we can see in Figure 10, the precision of a single event recognition improves as more ground terms are introduced into the training set. Looking down the column from left to right, we will notice that the

effect of the number of constants in the training set stabilises at some point. We found this development interesting in our preliminary analysis because if we consider the columns representing the number of constants 100 and 500 or columns representing the number of constants 1,000 and 1,500 for the event of a single constant as evidence, one may be tempted to conclude that any two training sets differing by 500 constants give approximately the same results. This notion, however, is different when we look at the columns representing the number of constants 500 and 1,000. Consequently, we tried varying the number of constants in the evidence set as well to better understand this trend.

By increasing the constants of the evidence set to 50, we observe that the precision of the recognition increases from below 60% to about 70%. Whilst much improvement is achieved between 1 and 10 constants in evidence, the performance stabilises from 20 constants. This gives the intuition that a caveat can be established for the density of sensory information at a given location since more evidence after some point can be inadmissible in the recognition process. From this observation, we hypothesize that a coherent representation of events combining modalities can allow CPSs to efficiently monitor and control environments with multiple sensors.

Concurrent event recognition was also investigated using multiple events representing the two distributed domains under consideration. Similarly, we measured the precision for recognition of multiple events using training sets containing constants 100, 500, 1,000, and 1,500. As shown in Figure 11, the precision for recognition of multiple events follows the trend of the single event recognition. This observation is attributable to the fact that even though different constants

FIGURE 11: Precision of a multiple event recognition with increasing evidence.

applicable to the single event and multiple event cases generate different ground Markov networks with varying structures, the model performance is still guided by the common underlying MLN. However, we recognise that the monotonicity in the precision with increasing constants in evidence is not upheld fully in the multiple event recognition. Whereas the precision increases monotonically in the engine event, the home event does not follow this trend completely. But this new phenomenon is yet another indication that determining a caveat on the size of contextual data towards optimal precision in event recognition is paramount. Obviously, all training data sets hold that the precision of the home event is modal at 20 constants in evidence, which is consistent with the precision of single event recognition. Because the multiple event recognition process contains more combined constants than the single event recognition, the multiple event process performs better. In essence, this approach can support concurrency in operations of CPSs for collaborative processes.

Finally, we investigated a multivalued logic in which MLN clauses are somewhat fuzzified to represent partial knowledge. In mimicking rectangular fuzzy functions, we used the built-in Alchemy predicate `greaterThanEq(x, y)` to define a single MLN clause that represents both normal and high temperature event inputs. Weights of MLN clauses characterise subregions of high and normal temperature measurements. Overlapping regions, as enshrined in fuzzy logic [42], can be treated as a straightforward constants declaration in the training data set. For instance, temperature value 26°C was considered ambiguous in our experiments. So in the training data set, this temperature value was declared a few times to indicate that this same value could be described as normal and slightly warm. As we can see from Figure 12, we obtained impressive results with even least training data set. Using a training data set of 100 constants, the precision for a single fuzzy event recognition also improves with increasing number of constants in the evidence set. Overall, using MLN, event recognition in CPSs can be modelled to handle

FIGURE 12: Precision of a fuzzy event recognition with increasing evidence.

both uncertainty and vagueness whilst modelling domain uncertainty.

8. Conclusion

In this paper, we proposed a context-aware multiagent architecture for distributed reasoning in cyber-physical systems. This architecture is rooted in service-oriented computing, and with the incorporation of services of semantic agents, the seamless integration of the cyber and physical components can be achieved. Ontological intelligence provides the underlying semantics of this approach, and together with Markov logic networks, we defined an uncertainty-based reasoning procedure for event recognition in cyber-physical systems. With the results of our experiments, it is convincing that this framework can be relied upon for concurrent processing of events in cyber-physical systems. Because these semantic agents are thought to be autonomous and intelligent in their operations, future work of this research shall consider agent communication techniques that can ensure good level of cooperation among these agents.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research was sponsored by NSFC 61370151 and 61202211, National Science and Technology Major Project of China 2015ZX03003012, Central University Basic Research Funds Foundation of China ZYGX2014J055, and Huawei Technology Foundation YB2013120141.

References

- [1] R. Baheti and H. Gill, "Cyber-physical systems," *The Impact of Control Technology*, vol. 12, pp. 161–166, 2011.
- [2] E. A. Lee, "Cyber physical systems: design challenges," in *Proceedings of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC '08)*, pp. 363–369, Orlando, Fla, USA, May 2008.
- [3] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: a new frontier," in *Machine Learning in Cyber Trust*, pp. 3–13, Springer, Berlin, Germany, 2009.
- [4] Y. Tan, M. C. Vuran, and S. Goddard, "Spatio-temporal event model for cyber-physical systems," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS '09)*, pp. 44–50, Montreal, Canada, June 2009.
- [5] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [6] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the internet of things," *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, 2010.
- [7] J. Singh, O. Hussain, E. Chang, and T. Dillon, "Event handling for distributed real-time cyber-physical systems," in *Proceedings of the 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC '12)*, pp. 23–30, Shenzhen, China, April 2012.
- [8] S. Karnouskos, A. W. Colombo, T. Bangemann et al. et al., "The imcaesop architecture for cloud-based industrial cyber-physical systems," in *Industrial Cloud-Based Cyber-Physical Systems*, pp. 49–88, Springer, Berlin, Germany, 2014.
- [9] A. Boyd, D. Noller, P. Peters et al., *Soa in Manufacturing Guide Book*, MESA International, IBM Corporation and Capgemini Co-Branded White Paper, 2008.
- [10] R. Harrison and A. W. Colombo, "Collaborative automation from rigid coupling towards dynamic reconfigurable production systems," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 184–192, 2005.
- [11] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [12] K.-J. Lin and M. Panahi, "A real-time service-oriented framework to support sustainable cyber-physical systems," in *Proceedings of the 8th IEEE International Conference on Industrial Informatics (INDIN '10)*, pp. 15–21, IEEE, Osaka, Japan, July 2010.
- [13] D. D. Hoang, H.-Y. Paik, and C.-K. Kim, "Service-oriented middleware architectures for cyber-physical systems," *International Journal of Computer Science and Network Security*, vol. 12, no. 1, pp. 79–87, 2012.
- [14] J. Pascoe, "Adding generic contextual capabilities to wearable computers," in *Proceedings of the Digest of Papers. Second International Symposium on Wearable Computers*, pp. 92–99, Pittsburgh, PA, USA, Oct. 1998.
- [15] S. Noor and H. N. Minhas, "Context-aware perception for cyberphysical systems," in *Computational Intelligence for Decision Support in Cyber-Physical Systems*, pp. 149–167, Springer, 2014.
- [16] Z. Song, Y. Chen, C. R. Sastry, and N. C. Tas, *Optimal Observation for Cyber-Physical Systems*, Springer, London, UK, 2009.
- [17] L.-A. Tang, X. Yu, S. Kim et al., "Trustworthiness analysis of sensor data in cyber-physical systems," *Journal of Computer and System Sciences*, vol. 79, no. 3, pp. 383–401, 2013.

- [18] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.
- [19] J. Wan, H. Yan, H. Suo, and F. Li, "Advances in cyber-physical systems research," *KSI Transactions on Internet and Information Systems*, vol. 5, no. 11, pp. 1891–1908, 2011.
- [20] G. Quan, "An integrated simulation environment for cyber-physical system co-simulation," in *Proceedings of the National Workshop on High-Confidence Automotive Cyber-Physical Systems*, Troy, Mich, USA, April 2008.
- [21] T. W. Hnat, T. I. Sookoor, P. Hooimeijer, W. Weimer, and K. Whitehouse, "MacroLab: a vector-based macroprogramming framework for cyber-physical systems," in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08)*, pp. 225–238, ACM, Raleigh, NC, USA, November 2008.
- [22] H. J. La and S. D. Kim, "A service-based approach to designing cyber physical systems," in *Proceedings of the 9th IEEE/ACIS International Conference on Computer and Information Science (ICIS '10)*, pp. 895–900, Potsdam, Germany, August 2010.
- [23] C.-F. Lai, Y.-W. Ma, S.-Y. Chang, H.-C. Chao, and Y.-M. Huang, "OSGi-based services architecture for cyber-physical home control systems," *Computer Communications*, vol. 34, no. 2, pp. 184–191, 2011.
- [24] P. Pederson, D. Dudenhofer, S. Hartley, and M. Permann, *Critical Infrastructure Interdependency Modeling: A Survey of Us and International Research*, Idaho National Laboratory, 2006.
- [25] P. P. Datta, M. Christopher, and P. Allen, "Agent-based modelling of complex production/distribution systems to improve resilience," *International Journal of Logistics Research and Applications*, vol. 10, no. 3, pp. 187–203, 2007.
- [26] G. Jiang, W. W. Chung, and G. Cybenko, "Semantic agent technologies for tactical sensor networks," in *Proceedings of the SPIE Conference on AeroSense*, pp. 311–320, International Society for Optics and Photonics, 2003.
- [27] J. Liu and F. Zhao, "Towards semantic services for sensor-rich information systems," in *Proceedings of the 2nd International Conference on Broadband Networks (BROADNETS '05)*, pp. 44–51, Boston, Mass, USA, October 2005.
- [28] A. Elci and B. Rahnema, "Considerations on a new software architecture for distributed environments using autonomous semantic agents," in *Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC '05)*, vol. 2, pp. 133–138, IEEE, July 2005.
- [29] J. Lin, S. Sedigh, and A. Miller, "Modeling cyber-physical systems with semantic agents," in *Proceedings of the 34th Annual IEEE International Computer Software and Applications Conference Workshops (COMPSACW '10)*, pp. 13–18, Seoul, Korea (South), July 2010.
- [30] C. Talcott, "Cyber-physical systems and events," in *Software-Intensive Systems and New Computing Paradigms*, pp. 101–115, Springer, Berlin, Germany, 2008.
- [31] I. Lovrek, "Context awareness in mobile software agent network," *Tehničke Znanosti*, vol. 513, no. 15, pp. 7–28, 2012.
- [32] L. Serafini and A. Tamilin, "Drago: distributed reasoning architecture for the semantic web," in *The Semantic Web: Research and Applications*, pp. 361–376, Springer, 2005.
- [33] Y. Zhang, Y. Xu, H. Hu, and X. Liu, "Semantical information graph model toward fast information valuation in large team work," in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI '14)*, August 2014.
- [34] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology based context modeling and reasoning using OWL," in *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerCom '04)*, pp. 18–22, Orlando, Fla, USA, March 2004.
- [35] P. Derler, E. A. Lee, and A. Sangiovanni Vincentelli, "Modeling cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, 2012.
- [36] L. Ding, J. Shinavier, T. Finin, and D. L. McGuinness, *Owl: Sameas and Linked Data: An Empirical Study*, 2010.
- [37] H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness, and H. S. Thompson, "When owl: sameas isn't the same: an analysis of identity in linked data," in *The Semantic Web—ISWC 2010*, vol. 6496 of *Lecture Notes in Computer Science*, pp. 305–320, Springer, Berlin, Germany, 2010.
- [38] N. Bieberstein, S. Bose, L. Walker, and A. Lynch, "Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals," *IBM Systems Journal*, vol. 44, no. 4, pp. 691–708, 2005.
- [39] N. S. Schutte, J. M. Malouff, L. E. Hall et al., "Development and validation of a measure of emotional intelligence," *Personality and Individual Differences*, vol. 25, no. 2, pp. 167–177, 1998.
- [40] K. Kaneiwa, M. Iwazume, and K. Fukuda, "An upper ontology for event classifications and relations," in *AI 2007: Advances in Artificial Intelligence: 20th Australian Joint Conference on Artificial Intelligence, Gold Coast, Australia, December 2–6, 2007. Proceedings*, Lecture Notes in Computer Science, pp. 394–403, Springer, Berlin, Germany, 2007.
- [41] S. K. P. S. M. Richardson, P. Domingos, and M. S. H. Poon, *The Alchemy System for Statistical Relational AI: User Manual*, 2007.
- [42] L. A. Zadeh, G. J. Klir, and B. Yuan, *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*, vol. 6, World Scientific, 1996.