Social Cloud-Based Cognitive Reasoning for Task-Oriented Recommendation

Dina Hussein and Son N. Han, Institut Mines-Telecom
Gyu Myoung Lee, Liverpool John Moores University
Noel Crespi, Institut Mines-Telecom

An intelligent recommendation framework that applies a cognitive reasoning mechanism in the social Internet of Things is proposed to improve the relevancy of recommendation results.

The Internet of Things (IoT) paradigm covers a diverse range of technologies, including sensing, networking, computing, information processing, and intelligent control technologies.[1] In practice, the large scale, complexity, and highly heterogeneous nature of the IoT are the main challenges facing IoT technologies. The emerging social IoT (SIoT) aims to tackle some of these challenges.[2] The SIoT paradigm represents an ecosystem that allows people and smart devices to interact within a social framework resembling traditional social network services (SNSs). Service providers can use Web technologies on top of this framework to offer applications and services in a user-friendly manner. The SIoT builds on the emerging concept of *social objects*, in which smart objects are being connected to the Web, allowing autonomous and proactive interactions with other people and objects.[3]

Typically, context within the IoT is dealt with reactively. (See the sidebar for other work in this area.) That is, the objective aspects of context, which describes entities' states (location, device type, user identification, and so on), are usually considered for context-aware decision making.[4] However, providing adaptive services to meet users' specific situational needs requires the subjective aspects of context.[5] These subjective aspects describe cognitive states, such as the user's goal, preferences, and mood. Our approach considers both objective

 2325-6095/15/$31.00

and subjective aspects of context in characterizing a user's situation for intelligent recommendation in the SIoT.

A growing number of smart objects and devices are connected to the Internet. Smart spaces and building automation solutions and services that benefit from the connectivity and data generated by these objects have likewise proliferated. However, there's still a huge need to improve the intelligence mechanisms required to make such solutions and smart services more adaptive to users' needs and conditions, especially to aid senior people and those needing specific medical care, such as people dealing with dementia.

We propose a cognitive reasoning mechanism that combines objective and subjective aspects of context in characterizing users' situations. We apply this reasoning mechanism within InRe, a task-oriented intelligent recommendation system that recommends quotidian tasks based on a user's situation in a smart home. InRe is fitted to users' situational goals, which we detect through their schedules, preferences, and daily habits, as well as the devices and environmental conditions in their smart homes. From an architectural viewpoint, InRe is a service built on top of the social cloud (SoC) to benefit from the contextual data extraction, reasoning, and storage capabilities that the SoC provides. From a technological viewpoint, we adopt Web services at the device level to ensure network navigability and direct human-to-object interactions. We developed the application ThingsChat to illustrate InRe's operation in a smart home. We consider the lightweight version of the Device Profile for Web Service (DPWS) specification,[6] which doesn't require devices with powerful capabilities to fit into the system. Our initial experiments show an improved adaptability of recommendation results to users' situations.

Cognitive Reasoning in the SIoT

Our proposed cognitive reasoning mechanism combines objective and subjective contextual elements to characterize users' situations and to infer their situational goals and thus the tasks that would fit such goals. InRe applies this reasoning approach by generating a list of quotidian tasks that are relevant to users' situations in smart homes. Therefore, to achieve intelligence in the SIoT, which entails situation characterization and proactive decision making, we need a detailed contextual model of users' social objects and the surrounding environment. Semantic Web technologies provide a scalable means for context-aware applications and services to access and reuse contextual data available in the SIoT. Although knowledge reuse is one important advantage of using an ontology, in this article we build on domain ontologies, such as Friend of a Friend (FOAF, http://xmlns.com/foaf/0.1) and Semantic Sensor Network (SSN, www.w3.org/2005/Incubator/ssn/ssnx/ssn). These domain ontologies provide generic vocabularies that suit context-modeling requirements. However, we extend these ontologies by adding new vocabularies, aiming to utilize the context model for generating task-oriented recommendation of smart services in smart homes. In our SIoT context model, we suggest two kinds of relationships between people and objects: ownership and authorization to use. Device owners or building managers can authorize users to establish social relationships with objects surrounding them (see Figure 1a).

Context Representations

An ontology is a formal description of concepts that are often conceived as a set of entities, properties, instances, functions, and axioms. The Web Ontology Language (OWL) in this sense defines and instantiates ontologies in a manner that lets Web agents interpret and exchange information based on a common sense vocabulary. Smart spaces typically cover a diverse range of environment types, such as homes and offices. Additionally, given that most smart spaces have limited resources, including limited CPU speed and processing capabilities, we adopted a two-layer hierarchical ontology model:

- a general upper ontology (see Figure 1a) representing general concepts and ontological classes in smart spaces, and
- a domain-specific ontology (see Figure 1b) that represents contextual details about people, objects and items existing in smart homes.

The contextual model shown in Figures 1a and 1b represents context as ontology instances with their associated properties. We refer to this combination as *context markups*. The upper ontology fragment in

RELATED WORK IN SOCIAL CLOUDS

Cloud computing offers a great opportunity for allowing access to shared computer utilities that can accommodate large-scale and heterogeneous application requirements. It then enables the composition of appropriate cloud utilities that can best fit the needs of given applications.[1] Kyle Chard and his colleagues provide an explicit definition for the social cloud (SoC): "A social cloud is a resource and service sharing framework utilizing relationships established between members of a social network."[2] Hence, the SoC emerged in the literature with a potential to realize the vision of the social Internet of Things (SIoT). That is, the SoC allows platform-independent sharing of physical resources and services based on the trust existing between nodes on the social network of everything.

Luigi Atzori and his colleagues were among the first to view the concept of the SIoT as an evolutionary step following IoT.[3] According to this idea, social relationships among objects and people can be established in a similar way to human relationships. This suggested social structure of people and objects can improve the navigability and discovery of network objects in a manner similar to a traditional social network service (SNS).

Achieving intelligent decision making in SIoT environments is a challenging issue. Intelligent systems should go beyond filtering from a list of prestored services. Rather, they should be equipped with reasoning methods to monitor and model situations to gather the knowledge necessary for situational decision making.[4] In her PhD dissertation, Katharina Rasch provides a thorough study for realizing smart assistants in smart homes.[5] She proposes a collaborative filtering-based recommendation system that filter users' preferences for suggesting a list of actions to be performed at home. One issue from this study that's an interesting topic for future investigation is context awareness, which involves exploiting users' short-term goals, preferences, geographical information, and calendar events extracted from social networks to detect the user's current context and thus provide relevant intelligent recommendation.[5]

Mario Muñoz-Organero and his colleagues provide a collaborative filtering-based recommendation system for IoT smart services[6] that takes into account user location and interaction time to recommend scattered, pervasive context-embedded networked objects. However, collaborative filtering-based recommendation systems rely on a straightforward user model that considers a user as a vector of item ratings where additional profile information, including preferences, location, and status, is considered as an extension to the basic user model.[7] This kind of recommendation, however, ignores situational needs. Because users' preferences, status, and other profile-based information can vary from one context to another, context-aware recommendation systems are more relevant to meet users' situational short-term goals.

References

1. K. Keahey and M. Parashar, "Enabling On-Demand Science via Cloud Computing," *IEEE Cloud Computing,* vol. 1, no. 1, 2014, pp. 21–27.
2. K. Chard et al. "Social Cloud Computing: A Vision for Socially Motivated Resource Sharing," *IEEE Trans. Services Computing*, vol. 5, no. 4, 2012, pp. 551–563.
3. L. Atzori, A. Iera, and G. Morabito, "SIoT: Giving a Social Structure to the Internet of Things," *IEEE Comm. Letters*, vol. 15, no. 11, 2011, pp. 1193–1195.
4. C. Perera et al., "Context Aware Computing for the Internet of Things: A Survey," *Comm. Surveys & Tutorials*, vol. 16, no. 1, 2014, pp. 414–454.
5. K. Rasch, "Smart Assistants for Smart Homes," Trita-ICT-ECS AVH ISSN: 1653-6363; 13:16, PhD Thesis, KTH Royal Institute of Technology.
6. M. Munoz-Organero et al., "A Collaborative Recommender System Based on Space-Time Similarities," *IEEE Pervasive Computing*, vol. 9, no. 3, 2010, pp. 81–87.
7. G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems," *Recommender Systems Handbook*, Springer, 2011, pp. 217–253.

Figure 1a represents context markups with relatively low change rates. For instance, user preferences, relationships with devices, and device-associated services don't change very often. The middle ontology fragment (Figure 1b), on the other hand, shows resources that provide dynamic contextual data, such as loca-

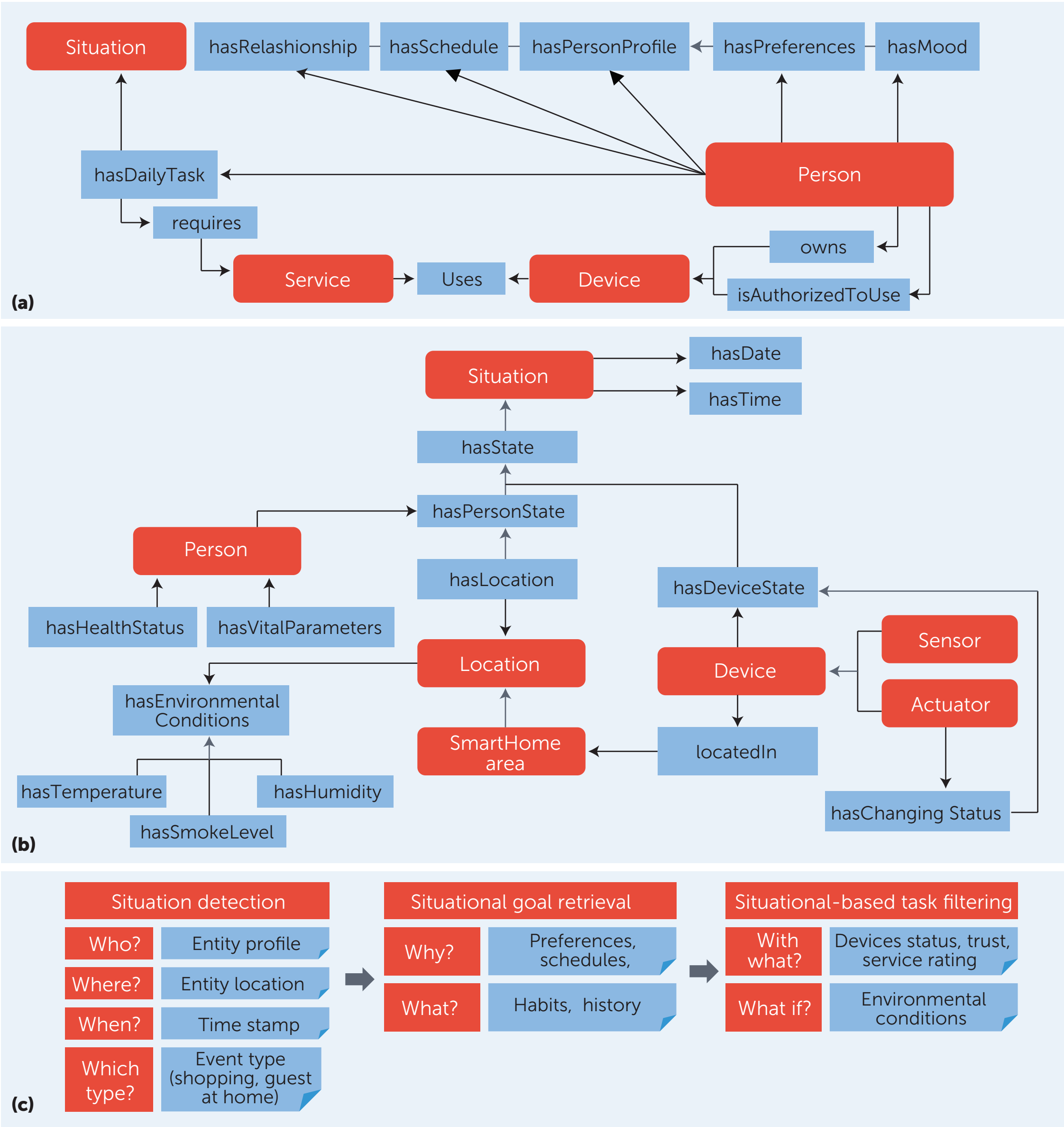

FIGURE 1. Context representation and use in the social Internet of Things (SIoT): (a) SIoT upper-ontology fragment, (b) SIoT lower-ontology fragment, and (c) three-phase situational reasoning in the SIoT following the seven WH basic reasoning questions (what, where, when, who, with what, how, and why).

tion, time, and user status. In this sense, the applications running the ontology model require automation of context markups. Consider, for example, a mobile device application that detects a user's location whenever the user's presence at a certain spot exceeds five minutes. A mobile application could compose the following OWL markup to announce user Nadia's presence at the supermarket:

```
<Person rdf:about="#Nadia"> <hasLocation
rdf:about="#Supermarket01"/> </Person>
```

Each OWL instance has a unique URI that context markups can use to link to other definitions. For instance, the URI "http://www.telecom-sudparis.eu/SIoTData#Nadia" refers to a certain user, and another URI will refer to the supermarket, which is defined elsewhere in our system.

Three-Phase Situational Reasoning

The SIoT context infrastructure lets applications running on top of it retrieve context using queries and supports the inference of higher-level contexts

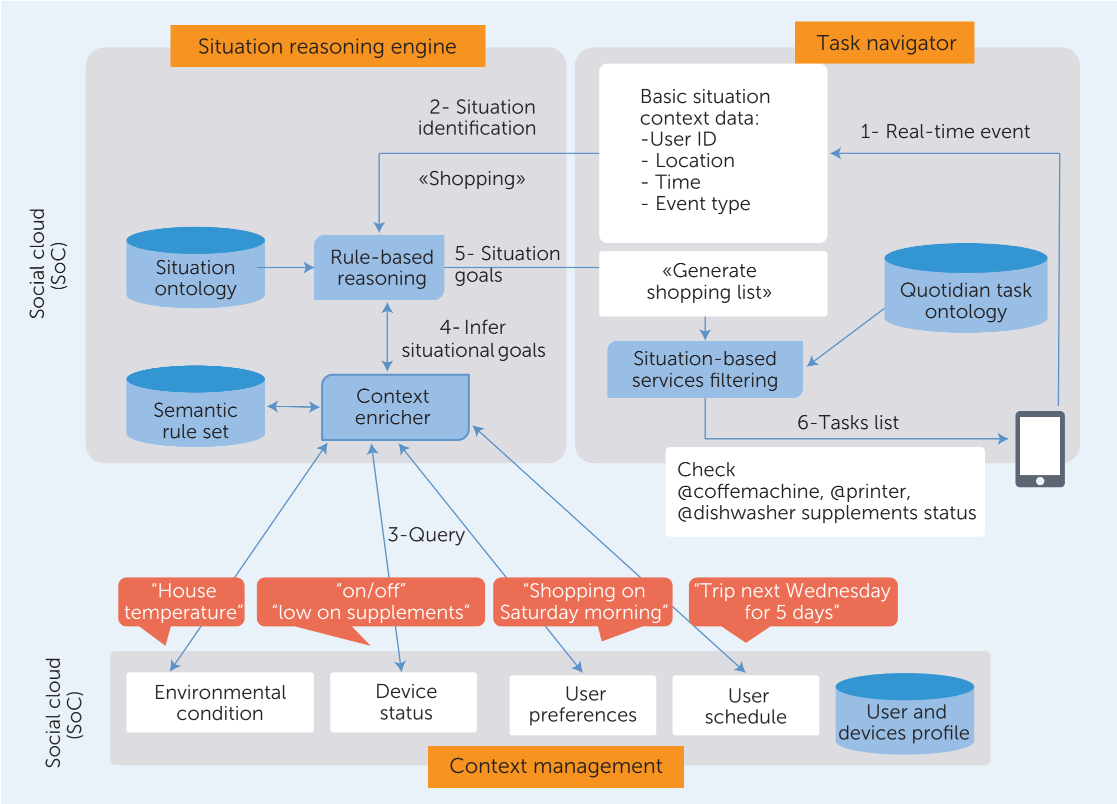

FIGURE 2. The InRe framework. Three main modules contribute to running context queries and reasoning and recommendation tasks. The social cloud (SoC) is proposed as an architectural infrastructure for hosting the InRe modules.

from basic contexts. Cognitive reasoning refers to combining the objective and subjective aspects of context to produce a recommendation list fitted to the situation. The three-phase situational reasoning model represents facts along seven dimensions corresponding to the seven WH questions: what, where, when, who, with what, how, and why (see Figure 1c).[7]

Phase 1: Situation detection. In the first phase, basic contextual data is exploited to identify the main entities involved in a certain situation. The system fetches spatiotemporal data to detect where and when an event is taking place. It then matches this data against relevant event types based on the user's location. For instance, if user Nadia's location is detected at a certain time as in the supermarket, the event is defined as "shopping." Similarly, when a non-household member is detected in a smart home, the event is defined as "guest at home." Finally, the combination of user, location, time, and event type context markups forms a situation.

Phase 2: Situational goal retrieval. In the second phase, the system infers high-level context from the basic context data fetched in the previous step. It represents contextual markups about user habits and history in similar previous situations in addition to user preferences and schedules. For instance, if user Nadia typically shops on Saturday and her schedule says she will be on a trip on Saturday, a reminder to do her shopping when she's near the supermarket would be considered a situational goal.

Phase 3: Situation-based task filtering. In the third phase, and based on the situational goals retrieved in the previous phase, the system generates a list of relevant tasks. It then matches these tasks with smart services available in the smart home. Contextual aspects such as service rating and environmental conditions are exploited to eliminate irrelevant services.

InRe Framework

Figure 2 depicts the overall recommendation system architecture, which relies on the context infrastructure and cognitive reasoning mechanism described earlier. Form an architectural viewpoint, the SoC

```
type(?user, User),locatedIn(?user, ShoppingMall), TimerHasValueGreaterThan(currentTime(), 07:00:00),
TimerHasValuelessThan(currentTime(), 08:30:00)
→ situation(?user,AtShopping)
type(?event,AtShopping),type(?SmartHomeApplianceHasStatus,ApplianceStatus),
CoffeeMachineHasStatus (?ApplianceStatus,ChangeFilter),
WachingMachineHasStatus(?ApplianceStatus,Fine),
DishwashingMachineHasStatus(?ApplianceStatus,ChangeSalt),
PrinterHasStatus(?ApplianceStatus,ChangeInk),
TVHasStatus(?ApplianceStatus,Fine)
→ recommendation(?SmartHomeApplianceHasStatus,SupplimentsNeeded)
```

FIGURE 3. Sample rules to infer users' situation based on context, location, and surrounding objects. When the user is shopping, the reasoning engine checks the status of the appliances at home. If it detects that a device is low on supplies (for example, the coffee machine needs a new filter or the printer needs ink), it sends a list to the user.

stores contextual data as well as reasoning and inference tasks. The framework consists of three main modules: context management, situation reasoning engine, and task navigator.

The user's situation is first identified when the system is triggered or a user-initiated event occurs. To characterize users' situations and thus infer situational goals, the context enhancer component collects additional contextual data. Accordingly, the rule-based reasoning component sends queries to gather information relevant to the situation. This information comprises user preferences in relevant situations, schedules, devices status, and environmental conditions. The rule-based reasoning component matches this contextual data semantically against the SIoT situation ontology, which represents common-sense knowledge about typical daily tasks corresponding to particular situations (turning on the robot cleaner when expecting guests, updating a shopping list while the user is shopping, doing laundry before a scheduled trip, and so on). A list of tasks is then sent to the situation-based services filtering module. The services filtering module semantically matches tasks with corresponding smart services using SIoT quotidian-tasks ontology, and generates a recommendation list showing smart services that corresponds to the user's situation. The user's selected services are then activated. Details of InRe's three main modules are listed below.

The *context management module* provides persistent context storage. It stores contextual markups gathered from context wrappers, which are responsible for obtaining objective and subjective context from various sources, such as physical objects and SNS profiles, and transforming them into context markups. These markups are described as OWL representations, allowing other components to access and reuse them. This module also acts as an abstract interface for the situation-reasoning engine module to extract desired context from the context enricher via queries. The reasoning engine can therefore access context in the context management module.

This *situation reasoning engine* is responsible for context processing and ontology parsing based on logic reasoning. Using this module, developers can create their own rules based on a predefined format. Once predefined rules are triggered, the situation reasoning engine can extract facts about the situation and recommend related tasks to the user. Figure 3 is an example of rule-based recommendation.

The *task navigator* acts as an interface for gathering basic context data, which can later help infer more complex context, and displays situation-relevant tasks. It's also responsible for running the situation-based services filtering algorithm. This algorithm semantically matches situational goals against the quotidian-task ontology to determine matching tasks and the smart services that can fulfill these tasks.

Application Prototype

In this article, we build on SNSs to bring together users' social relationships, objects, Web services, and other elements of a user's world. In this sense, an SNS provides an environment in which profiles of people and objects can be built and social relationships established.

We developed an SNS-based platform, ThingsChat, that lets users perform the following functions:

- create relationships with social objects (add objects to a friends list),
- browse social objects,
- receive objects' status, and
- control objects and navigate through quotidian-task recommendations.

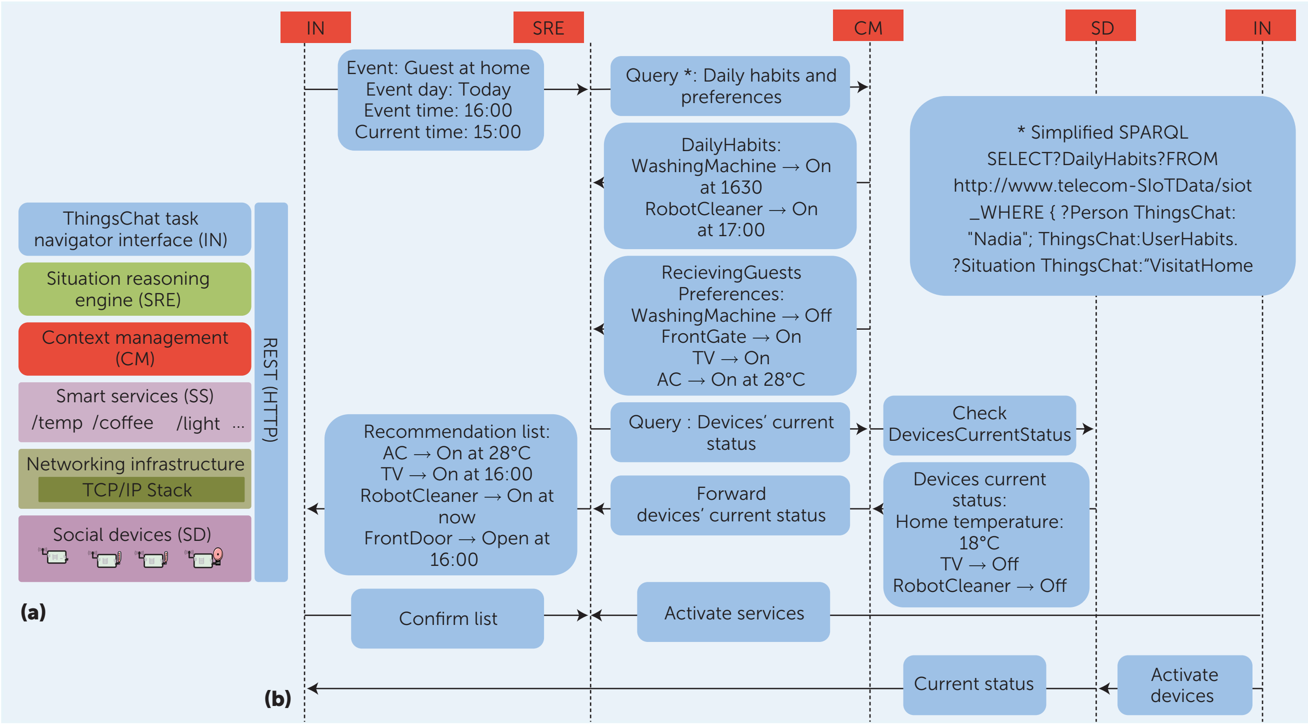

FIGURE 4. ThingsChat application scenario: (a) modules for actuating recommendation services, and (b) InRe sequence diagram.

We adopt the following application scenario to highlight the ThingsChat's main functions (see Figure 4).

Nadia is in her office and receives a text from her mother, Leila, who is near Nadia's house and wants to visit. Nadia sends a message to her smart home virtual group in ThingsChat, informing it about the visit and asking the system to recommend a list of tasks that need to be done to make sure the house is ready to receive her mother. InRe, which is implemented as a module inside ThingsChat, first checks Nadia's preferences and habits when receiving a guest at her home. Next, it checks the condition of the house and status of devices before reasoning about a list of tasks required for intelligent recommendation. After Nadia approves the task list, device actions are activated at home to prepare for the visit. While the services are running (for example, house cleaning, dishwashing, and turning on the heater), Nadia can directly interact with her coffee machine, asking it to prepare her mother's favorite coffee when she arrives. The house is now ready to receive Leila.

ThingsChat: Your Things Are Chatting

As Figure 5 shows, ThingsChat has two modules: DPWSim and ThingsGate, which communicate with the underlying DPWS standards. DPWSim is a Java-based simulation environment for DPWS devices with a graphical interface to animate device operation.[8] It uses Web Services for Devices Java Multi Edition DPWS Stack (WS4D-JMEDS) to handle DPWS protocols and is compatible with the DPWS specification.[8] ThingsGate acts as a wrapper to present DPWS device functionalities in a RESTful style—that is, using HTTP methods (GET, PUT, POST, and DELETE)—to allow Web applications to seamlessly interact with DPWS devices, and to perform social networking tasks. ThingsGate also provides a social device API to meet the requirements of designing an SNS.

The ThingsChat platform is based on the open source phpBB (www.phpbb.com) social network, with Linux, Apache, MySQL, and PHP stacks in the background. It extends the original user profile in the phpBB database to store information about the gateway IP address. This information, unique for each device, allows communication between ThingsChat and social objects (through the ThingsGate social device API). Additionally, the object owner can set the level of authorization for each object, which enables it to be seen within a home or office network, and store it in the object profile. Thus, relationships can be established between users and objects by adding a certain object to the user's contact list as a "friend." Figure 5 shows some ThingsChat usage scenarios.

FIGURE 5. Users access ThingsGate with a smartphone via the mobile Web interface to discover and manage DPWS devices created using DPWSim in a virtual home: (1) user Nadia sends her home appliances group a notification that her mother, Leila, will be arriving and to prepare the house to welcome her changes; (2) InRe accesses profiles for both Nadia and Leila and recommends a list of services to be performed in the home (cleaning, opening the external gate, and so on); and (3) Nadia accepts the recommended tasks, except for switching on the TV. She also chats with the coffee maker to ask it to prepare coffee when her mother arrives.

Figure 6 shows the communication from a ThingsChat user asking her coffee maker to switch on.

We deployed InRe in a separate server running Apache Tomcat, using the Jena library for semantic data manipulation and the integrated reasoner for inference functionalities. InRe collects profile information from the Jena reasoner and the inference engine and matches this information with reasoning rules. InRe also provides a RESTful API for access from other ThingsChat objects.

We also developed a small Natural Language Processor (NLP) module inside the InRe module to detect and process conversation-based events such as chatting. The NLP module fetches the basic contextual data in each devices' profile at the setup phase in order to get the keywords of the device's functions. It also includes a set of rules for obtaining the meanings in English. InRe converts output from the NLP into a semantic format, after which InRe matches it with the list of events stored in the context knowledge base and generates a recommended list of tasks. Figure 7 illustrates a "guest at home" event.

```
POST /mention.tsp HTTP/1.1
Host: http://157.159.103.10:8080
device_name=CoffeeMaker&post_id=135&text=switch%20on
```

FIGURE 6. To get her coffee machine to switch on, a user sends a POST message to the device by simply mentioning the device in a status update.

InRe Performance Evaluation

We can incorporate context at various stages of the recommendation process. Given the richness of contextual data in the SIoT from cyber, physical, and social worlds, it's essential to scale down the amount needed prior to decision making. InRe, which is based on the prefiltering context paradigm,[9] addresses this need. Depending on the context, InRe

```
@prefix : <http://www.telecom-sudparis.eu/siot_data#> .
@prefix siot: <http://www.telecom-sudparis.eu/siot_ont#> .
:Visiting
    a siot:ReceptionEvent ;
    siot:hasVisitor "Leila" ;
    siot:locatedAt :NadiaHome ;
    siot:time "15:30" .
```

FIGURE 7. Data about Leila's visit obtained from the Natural Language Processor (NLP) is stored in N3 semantic model format. The situation reasoning engine (SRE) processes this data to get a specific list of recommended tasks to perform at Nadia's home to serve Leila based on the predefined visiting event.

performs the semantic rules for recommendation generation only against services with matching properties, that is, the service input and output.

We evaluated InRe's task filtering effectiveness, which indicates recommendation quality. To do this, we calculated precision, recall, and F-measure values for some event-based recommendation scenarios. As Equations 1 through 3 show, precision () is the ratio of the number of relevant services to the total number of recommended services, and recall () is the ratio of the number of recommended relevant services to the total number of relevant services. The F-measure is the measure of the testing accuracy considering both and :

$$Precision = \frac{Recommended\ services\ \cap\ Relevant\ services}{Recommended\ services} \tag{1}$$

$$Recall = \frac{Recommended\ services \cap Relevant\ services}{Relevant\ services} \tag{2}$$

$$F - measure = 2.\frac{p.r}{p+r}. \tag{3}$$

To highlight the significance of including both subjective and objective context in the process of recommendation prefiltering, we calculated and F values in four different scenarios for smart home events:

1. having a guest at home,
2. shopping,
3. high temperature in the kitchen, and
4. having a conference call at home.

We used a context synthetic dataset with 3,057 triples (or 600 OWL classes and instances). We evaluated the matched recommendation result of each scenario by comparing two context prefiltering methods via

- the user's daily habits, preferences, and environmental conditions (Figure 8a); and
- time, location, and environmental conditions (Figure 8b).

As Figure 5 shows, the prefiltering method in Figure 8a performed better than the method in Figure 8b.

Additionally, we tested the ThingsChat platform's responsiveness in an experiment using the following elements: a virtual home created using DPWSim that consists of several DPWS devices, including robot cleaners, TV, coffee maker, and floor lamps; the ThingsGate gateway; the social network ThingsChat; and InRe.

We implemented ThingsChat and InRe on application servers using an Intel Core i5-2540M CPU at 2.60 GHz with 6 Gbytes of RAM. DPWSim runs on a Windows 7 computer, and ThingsGate is implemented on a virtual machine on the same computer with one CPU, execution cap of 50 percent, and 512 Mbytes of RAM. All servers are deployed in the same local network. We performed 25 tests focusing on the scenario in which a user sends various messages to devices via ThingsChat asking them to accomplish tasks. For each message sent, we analyzed the users' text and matched it with semantic rules to convert it into a set of commands to be executed on the device. We achieved encouraging results with a response time stable at 2 to 3 seconds for each message sent.

The initial experiments reported here show stable results, highlighting the capabilities that could be achieved when incorporating the SoC as an infrastructure for running required reasoning processes as well as storing and managing the huge amount of contextual data available in the SIoT. In the future, we plan to investigate and incorporate dynamic ranking of recommendation results to produce an adaptive ranked list of recommendations on the fly according to daily situations in various smart spaces. We also plan to investigate the use of trustworthiness in terms of security, privacy, and usability as contextual aspects in the recommendation process with the integration of the SIoT and the SoC.

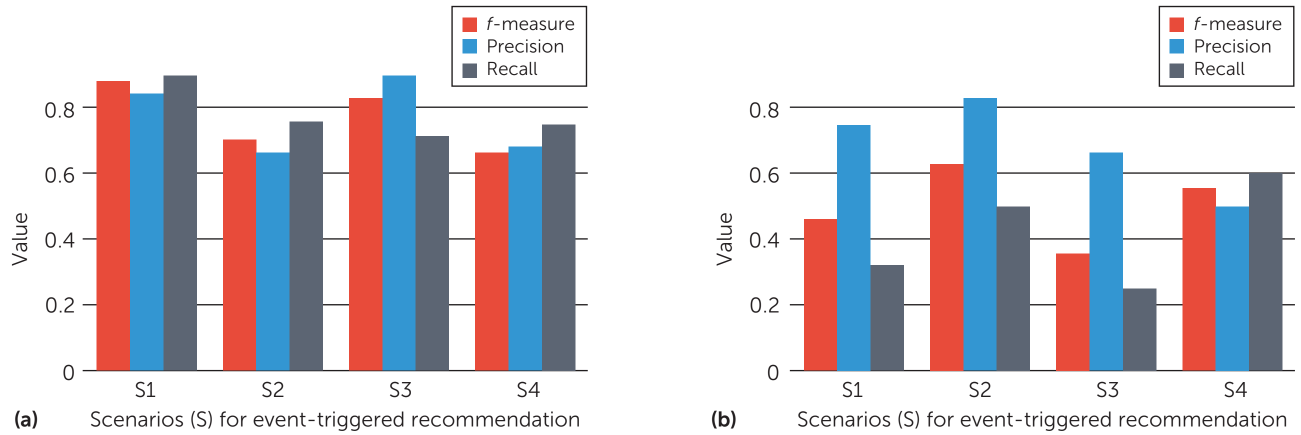

FIGURE 8. Results for applying two prefiltering methods on event-based recommendation for scenarios 1 through 4, where the prefiltering method considered (a) the user's habits and preferences as well as environmental conditions, and (b) time, location, and environmental conditions.

References

1. O. Vermesan et al., "Internet of Things Strategic Research Roadmap," *Internet of Things: Global Technological and Societal Trends*, O. Vermesan and P. Friess, eds., River Publishers, pp. 9–52.
2. L. Atzori et al., "The Social Internet of Things (SIoT): When Social Networks Meet the Internet of Things: Concept, Architecture, and Network Characterization," *Computer Networks*, vol. 56, no. 16, 2012, pp. 3594–3608.
3. D. Guinard, M. Fischer, and V. Trifa, "Sharing Using Social Networks in a Composable Web of Things," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. Workshops* (PERCOM), 2010, pp. 702–707.
4. C. Perera et al., "Context Aware Computing for the Internet of Things: A Survey," *Comm. Surveys & Tutorials*, vol. 16, no. 1, 2014, pp. 414–454.
5. J. Hong, E. Suh, and S.-J. Kim, "Context-Aware Systems: A Literature Review and Classification," *Expert Systems with Applications*, vol. 36, no. 4, 2009, pp. 8509–8522.
6. *Devices Profile for Web Services Version 1.1*, Organization for the Advancement of Structured Information Standards (OASIS), 2009; http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01.
7. D.W. Robertson, "A Note on the Classical Origin of 'Circumstances' in the Medieval Confessional," *J. Studies Philology*, vol. 43, no. 1, 1946, pp. 6–14.
8. S.N. Han, G.M. Lee, and N. Crespi, "Semantic Context-Aware Service Composition for Building Automation System," *IEEE Trans. Industrial Informatics*, vol. 10, no. 1, 2014, pp. 752–761.
9. G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender Systems," *Recommender Systems Handbook*, Springer, 2011, pp. 217–253.

DINA HUSSEIN *is currently with the Networks and Mobile Multimedia Services Department at the Institut Mines-Telecom, Telecom SudParis. Her research interests include socially enhanced services and networks, Semantic Web technology and context-aware computing. Hussein has a PhD from the Institut Mines-Telecom. She's a student member of IEEE. Contact her at dina.hussein@it-sudparis.eu.*

SON N. HAN *is currently with the Networks and Mobile Multimedia Services Department at the Institut Mines-Telecom, Telecom SudParis. Han has a PhD from the Institut Mines-Telecom. He's a student member of IEEE. Contact him at son.han@it-sudparis.eu.*

GYU MYOUNG LEE *is a senior lecturer at the Liverpool John Moores University and an adjunct professor at the Korea Advanced Institute of Science and Technology (KAIST). His research interests include future networks and services. Lee has a PhD from the Korea Advanced Institute of Science and Technology (KAIST). He's a senior member of IEEE. Contact him at G.M.Lee@ljmu.ac.uk.*

NOËL CRESPI *is a professor in the Networks and Mobile Multimedia Services Department at the Institut Mines-Telecom, Telecom SudParis. His research interests include service architecture and the Internet of Things, people, and services. Crespi has a PhD from Paris VI University. He's a senior member of IEEE. Contact him at noel.crespi@mines-telecom.fr.*