Generic Application Driven Situation Awareness via Ontological Situation Recognition

Ryan Pearson, Mark P. Donnelly, Jun Liu and Leo Galway
Computer Science Research Institute, School of Computing & Mathematics,
University of Ulster, Newtownabbey, Co. Antrim, UK
pearson-r2@email.ulster.ac.uk; {mp.donnelly, j.liu, l.galway} @ulster.ac.uk

Abstract— **Situation recognition and interpretation based on multisensor data is an important research challenge in the situation awareness field. Existing research has developed techniques concerned with accurate and reliable situation recognition via sensor driven detection of events in an environment. However, real world applications of situation awareness require perception of a situation's meaning, knowledge of expected changes and their relevance to environments inhabitants. Recognizing the significance and implications of situations in complex real world scenarios is challenging, but is essential for designing applications for real world environments. This paper presents a novel knowledge driven approach to situation awareness. Within it we extend established data driven methods of situation recognition by utilizing domain knowledge across the entire situation life cycle. We utilize ontologies for explicit representation of environmental and application context as well as situation modeling. We explore the link between low-level environment context and high-level application knowledge using a generic situation model. We exploit semantic reasoning to provide situation recognition and interpretation and demonstrate delivery of application oriented situation awareness in a smart environment. Finally, a case study-based scenario is utilized in order to demonstrate the system's operation.**

Keywords*— situation awareness; context awareness; ontology; knowledge representation***

I. INTRODUCTION (*HEADING 1*)

Pervasive computing requires agents that act upon an environment intelligently in order to provide unobtrusive support to its inhabitants. In order to achieve this, these agents require knowledge and an understanding of the state of the environment in which they operate. This information is typically provided as an abstract model of the physical environment. This work focuses on modeling knowledge for application oriented pervasive systems, aiming to provide an explicit method of defining ontologies for this purpose that allow interpretation of events within a general domain into specific situations.

The aim of any context aware system is to have a understanding of the environment it is situated in and the ability act intelligently in that environment [1]. Many research interests have developed around this aim, the relevant area to this work is reasoning in knowledge-based systems. Situation awareness is a type of context aware behavior that refers to "knowing what's going on" within an environment [2]. It performs a specific abstraction of context that describes the state of the environment in relation to common goals shared between the system and its user. Therefore it is essential to provide a means of establishing this shared understanding of goals as well as the context relevant to them.

The framework proposed in this paper focuses on situation recognition based on application independent contextual information and providing a contribution of an application oriented method of knowledge based situation inference that is domain independent. The proposed objective of the framework is to provide a generic platform upon which general environment context can be modeled and related to models of application specific situations. The motivation for this study comes from increasing research on how best to provide application oriented situation awareness. Situation awareness has predominantly been studied from the perspective of situation recognition accuracy and performance [3] [4]. A lesser body of work has investigated the optimum way to interpret situations given the requirements of users in the environment [5]. Situations are subjective abstractions of context, meaning the situation recognized by different people or groups, given the same context, can vary widely based on their own goals in the environment as well as their personal characteristics. Adapting the recognition of situations based on the pervasively collected context of the environment's inhabitants, taking into account their own goals and personal characteristics, is an important aspect to providing situation awareness and leveraging a dynamic situation recognition capability.

In order to test and validate the conceptual model a scenario situated within an office-based environment is proposed. The system aims to show that office personnel can utilize situation awareness adapted to their specific needs based upon their own personal characteristics, knowledge of their goals and an understanding of the environment they inhabit. This work provides a description of the conceptual and logical components of the framework, its reasoning capability and the practical implementation of the system. A case study presenting a university based smart office system involving student-to-faculty and faculty-to-faculty interaction has been used to provide a practical description of the system.

This paper is further structured as follows: Section II investigates related work and motivation for our work; Section

III introduces application oriented situation awareness and the ontology supported knowledge based reasoning approach this work employs; Section IV discusses the specification of the situation, rule based reasoning and working process of the Section VI presents the results of the aforementioned system validation, examining the strengths and limitations of the approach; Section VII provides a summary of the work along with a description of the future direction for the research.

II. Related Work

Situations can be considered a subjective interpretation of events, given the particular circumstances in which they occur, and the perspective of the observer. Situation awareness therefore requires more than recognition of events; it requires an understanding of those events within the environment in which they occur in relation to the other events occurring within that environment in real time. Context aware systems, capable of recognizing events within an environment and the context that surrounds them, are typically employed as a means of gathering the type of information required for situation awareness. Conventional situation aware systems of this type have focused on situation recognition in an efficient and accurate way [5]. While these are important aspects of situation awareness, in terms of providing timely and reliable information, it is important to consider how situations interrelate and impact one another, and ultimately their relevance to given inhabitants of an environment.

A wide variety of research has been performed within the domain of situation awareness. In this work we investigate situations modeled as a series of events. Similar approaches are taken by Chen *et al.* [6], [7], who utilize ontological context models in situation recognition scenarios using description logic. Work conducted by Cimino *et al.* [8] investigates management of situation awareness by. The situation inference processes involves fuzzy representations of both time and spaces in specific domains. Recommendations are made based upon recognized situations and their associated tasks, however, there is no specific or explicit goal modeling technique used in order to reflect in depth knowledge of user goals within specific domain. In [9], a goal oriented view of situation awareness is envisioned within the human factors domain, specifically based on the work of Endsley [2]. The proposed framework provides a solution aimed at the problem of data confliction and data redundancy in the information integration and interpretation phase of situation recognition. This work addresses some of the pressing issues of refining situation awareness through the use of semantic web technologies and machine learning. However, it still does not address the important issue of situation interpretation and presentation to specific users, or user groups, accounting for their personal context. Alongside the work of Tamea and Cusmai [10], there is no method for handling the usefulness of the interpretation of situations from a user based perspective.

From the analysis of the literature, it is clear there is a large body of work that has investigated situation recognition. However, it is clear that there is a limited amount of existing research that investigates the relevant presentation of situations to users.

III. Application Oriented Situation awareness

Situation awareness involves knowledge of multiple contextual factors existing within an environment, the interrelationship between the contextual factors and their interpretation from the perspective of *actors* in the environment [11]. By application oriented we refer to research problems beyond efficient and accurate situation recognition. As already stated, a large body of research has focused upon improved identification of situations in terms of accuracy and efficiency [6], [11]. In the real world it is obvious that situations undergo a constant change of state; the situations in real world environments are dynamic, interleaved, sequential, have varied duration and can occur across a range of locations [5]. Furthermore, transitions between situations in the real world are instant and occur due to both environmental factors and complex interplay between other situations occurring in the same environment. In order for a system to provide situation awareness it must be capable of perceiving and reflecting this state of environment change in its delivery of situation awareness to its users. Furthermore, a situation aware system should be capable of comprehending the meaning of a situation with respect to the environment in which it occurs and in response to user goals [12].

The framework we proposed and developed has 4 main components, each interacting to form an information flow that establishes situation awareness from general contextual information from the environment. The Ontology Management component is a data collection, storage and management layer of the framework. It receives context information from the environment, creates a representation of that information, which conforms to the context model, and stores it in a database. It provides access to this information upon request as well as modification features. The context, situation and application ontological models, which have previously been created through a knowledge engineering processes, are also stored in this component, along with ontological alignments and semantic rules. The ontological reasoner is also accessed through this component of the framework.

The purpose of the Context Model component (Figure 1) acts as a basis for representing events in a particular environment from which situation recognition can be performed. Specifically, the context model represents events that have been observed by sensors located within the environment. The events are modeled in an application independent, generic format, meaning they are designed to capture the widest possible range of situations within the environment that can potentially occur and be recognized from the modeled events. The aim is to create a general event model that will facilitate modeling of a range of pervasive environments (i.e. smart home, office or hospital). The events represented within the contextual model can then be utilized as a basis to perform situation recognition and inference. The representation of events in the context model is updated in real time in order to reflect new events that occur within the environment. This process is handled by the context management component of the framework.

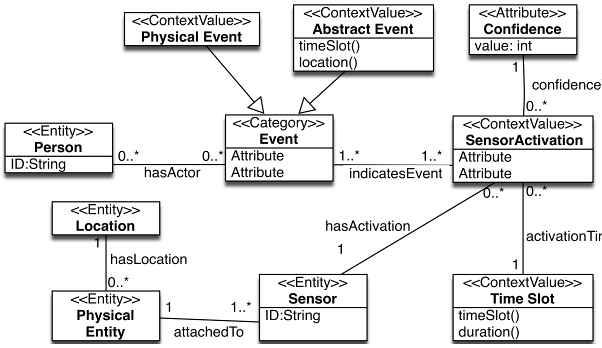

Fig. 1. Context Model Concepts and Relational Structure

The Situation Model, depicted in Figure 2, is a generic ontological model designed to infer the existence of situations as a result of events in the environment. The model captures the generic concepts required to represent situations across a range of application domains. Situations are specified as a sequence of events, which are utilized by the inference process, along with a defined context, to infer the existence of a situation. If a defined set of events does exist under specific context constraints then a situation related to these is deemed to occur. The basic inference process is based upon a generic rule base contained within the primary situation model component. These rules cover basic inference, such as *if event occurring then situation occurring*, to a more intricate inference process covering the time arrangement of events required for specific situations or the vicinity of events to defined location for other situations. The contextual classes related to the event class are equivalent to their counterpart classes in the event model. This allows contextual information (e.g. time of occurrence, location, entities involved) to be linked to and reasoned upon by the situation model. By incorporating context into the situation model, the context surrounding a situation can be inferred, i.e. a situation can be said to be occurring in a specific time and place involving specific entities rather than just occurring. Subsequently, this allows the application to actually interpret the meaning of situations given their particular reference point.

The application model is a representation of application specific use of situation awareness. Figure 3 provides a conceptual overview of its operation and processes. Specifically, it is used to model knowledge of user goals in the environment, situations that are related to those goals and applicable actions that could be taken towards achieving the goals. The inference process is used to decide an appropriate course of action, given a user goal and the situations that influence it. The information represented in the application model is highly specific to a single domain or application area due to the subjective perspective or outlook taken on information such as events that occur in an environment.

Consequently, the situations that are modeled in any individual application model are not intended to be generally applicable, or useable, in a range of applications scenarios but instead are highly specific to the needs of the application user base.

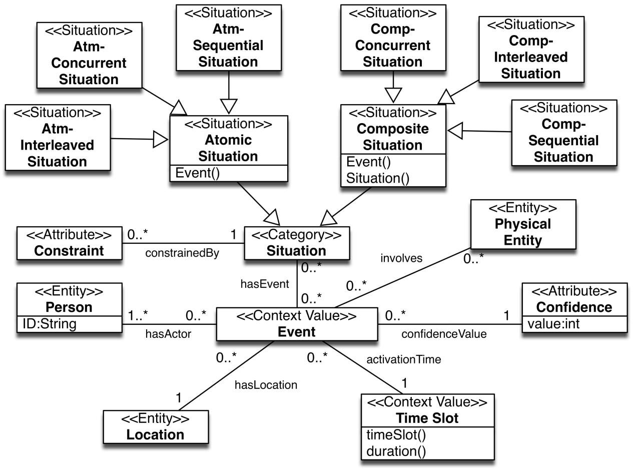

Fig. 2. Situation Model Concepts and Relational Structure

The interpretation of recognized situations is also highly specific to the applications user and their declared goal. This emphasis of depth over breadth of knowledge representation creates the potential for more insightful inference of situations, resulting in a much more effective application. As the model is application oriented, goals are modeled as a functional description of what a user wants to achieve. This practical perspective makes the decision-making required for each goal intelligible using only the experience of a domain expert and intuitively informs the actions and situations that should be specified around goals. Subsequently, goals are tied to specific environment entities that allow a contextual basis to be formed. In addition, a person's goals in an environment practically revolve around the people or objects within it.

Situations modeled within the application model are conceptually any situations of interest to a goal, i.e. they can have an impact upon a goal if they occur. They are also a subjective interpretation of the events within the environment given the context that surrounds those events. Accordingly, the application model interprets a series of events from the perspective of the goals of users in that environment. For this reason one series of events in an application domain, given a specific goal, can represent an entirely different situation given another goal in the same domain. Each situation is related to a distinct set of events, which are formally specified in the primary application model. As the event class of the application model is conceptually equivalent to the event class in the event model, any events modeled in the physical environment can be incorporated into situation specifications within the application model.

Actions are conceptually modled as appropriate actions, given knowledge of a user, that will lead to fulfillment of their goal given the current situation. As such, actions are tied to specific situations and goals and the context in which they occur. The inference process suggests an action to a user based upon a goal, its contextual constraints and the current situation, as depicted in Figure 3 that provides an illustration of this process within the application component of the framework.

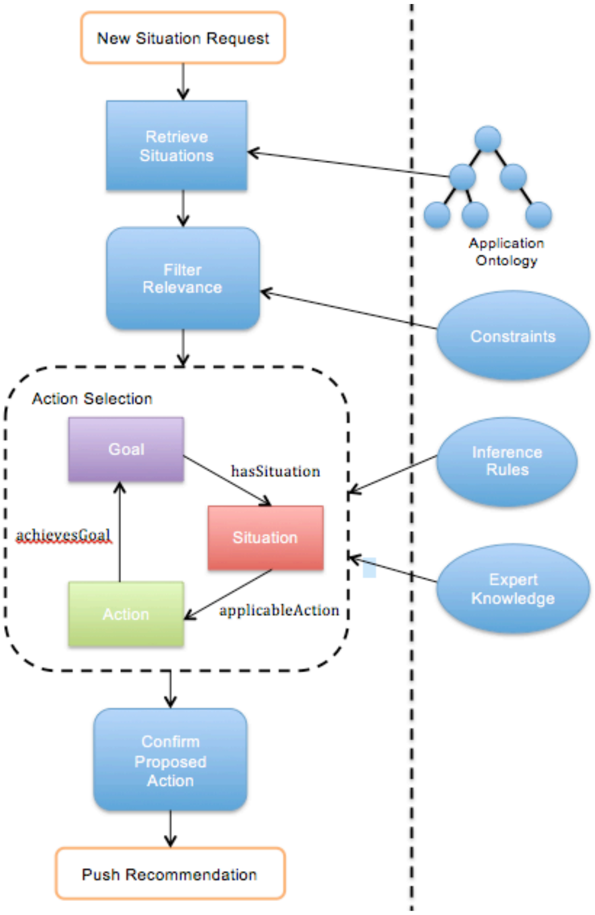

Fig. 3. Application Based Situation Inference Process using Application Model

IV. ONTOLOGICAL SITUATION MODELING

In order to address the requirements outlined in Section V relating to the implementation of the system based on specified technologies and techniques, situations are outlined as a series of events interrelated given their spatio-temporal and semantic characteristics. Our approach aims to be extensible, as it allows both a diverse range of contextual environment ontological models, as well as application ontological models to be integrated with the situation inference ontology previously outlined in Section III.

Temporal constraints are realized as classes of situation representing sequential, interleaved and concurrent situations. Specifically, Allen's representation of temporal intervals [13] as well as the work by Okeyo *et al.* [14] is used. Spatial aspects of situations are represented using the region connection calculus (RCC) [13], more specifically, the eight basic spatial relations between two regions. The work by Skillen *et al.* provides a background in user modeling and service personalization using ontological models [15].

A. Situation and Rule Design

Situation modeling is critical to application oriented situation awareness; situations are typically represented in a data structure called a *model*. The modeling process involved a balance between bottom-up and top-down approaches, as embodied by the general, low-level context model that can support a range of situation aware services within a single environment (e.g. `Physical_Event`, `Open_Door`), along with the application model, which represents high-level application specific concepts (e.g. `In_Meeting`, `At_Lecturer`). Relationships between the concepts are then established, defining the low-level contextual events that are part of the situations in the application model.

The situation and action inference processes used in the system are both rule based; SWRL rules are used to define a cause-effect type relationship between modeled entities. In SWRL [16], each rule is made up of an antecedent and consequent components. While SWRL rules are intuitive to construct, given the nature of situation awareness and a pervasive computing application, a large number of rules must potentially be developed. This is due to an individual rule being required for each possible variation in how an individual type of situation can occur (one rule per situation), as well as an associated rule being required to infer the appropriate action to recommend given the range of situations. Each rule that is modeled represents a real world conjunction of context conditions that result in a particular situation. Contextual conditions are taken from the context model, as mentioned previously. The contexts `Open_Door, Sit_Chair and Use_Computer` are taken from the `Physical_Event` class of the context model. The classes `Location` and `Time Slot` can be used to interrelate each event as co-occurring simultaneously within the same location. Given these contexts it would be reasonable to assume that a person is currently working in that location. If the location is `Office,` then the situation can reasonably be inferred to be `In_Office,` taken from the application model, at this time slot as a consequent to this rule. The rules that must be defined for a real world application are considerably more complex, however, this example serves as a brief explanation of the general principle.

B. Situation Inference Mechanism

To establish the existence of a situation, the reasoning engine is used to decide if the preconditions of a rule (i.e. events and relationships) are met, therefore leading to a consequent situation. The reasoning engine must also decide if the consequent of an already fired rule causes another rule in the system to fire (e.g. the inference of one situation causing the inference of another).

The reasoning processes used in this work is forward chaining. The context module takes observations from the environment and creates a representation of them within the context modules ontology, which represents environment events using a corresponding ontology class, as previously described in Section III. The application module holds the situation specifications relevant to the application's service(s). The application module ontology maintains a conceptual representation of each situation in the form of a class description with relationships to relevant events. The situation module sits between the application and context modules, as described in Section III. When a user makes a request of the system it is parsed by the application module, which performs a filtering process in order to decide upon the situations relevant to this request. The rules representing each situation in the filtered situation list are then imported into the working memory of the reasoner along with application specific constraints and user contextual information. At the same time, the current context is imported from the context module into the reasoner as well as the generic situation ontology and the ontology alignments are added to allow the flow of information between each of the framework modules. The context information and application specific constraints are subsequently used as variables in the logical operation of the inference rules. The reasoning engine will check to see if any of the antecedent conditions of each rule in the rule base are valid before firing a rule. Consequents of fired rules can then be used to fire other rules. In this way ground level contextual events as well as knowledge of application specific constraints can be used to perform application specific situation inference.

V. Course of Action Inference

The action inference process is similar to that for situation inference. Figure 4 describes the interplay between the goal, situation and action components of the application model. To infer the appropriate action to recommend to a user, knowledge of the user's goal must be shared with the system. A representation of this goal is established in the application model under the appropriate class. Goal classes are predefined during the model development phase given the service the application must provide. A range of situation classes are related to each goal based upon their ability to impact upon it. Action classes are also predefined in the model and related to specific goal-situation combinations, and conceptually represent the appropriate action to take for the specified combination. A rule is constructed to perform action inference given the currently occurring situation, inferred using the process described above. The rule base for performing actions incorporates knowledge of the goal, relevant situation and user information, along with preferences, which are both statically modeled in the application component of the framework as well as solicited at query time by the user. Based on this information, an appropriate recommendation of how best to pursue a goal, given the current state of the environment, can reliably be provided to the user. In this section the implementation of the developed situation aware application is described. The core aspects of the application's functionality are highlighted, followed by some discussion of the completed testing.

The current implementation took place across 5 stages: (1) development of the context management platform; (2) development of the environment context model and generic situation model; (3) creation of the application model, including user modeling and application specific situation modeling; (4) creation of the SWRL rules used to perform situation inference; (5) the simulated application scenario.

A. *Context, Situation and Application Model Development*

The context management platform was developed based upon the Apache Jena framework utilizing a Java implementation. The Jena Triple Database (TDB) was used for the implementation of the triple store. OWL-2 [17] was used as the ontology development specification due to expressiveness and support under the Jena framework, along with its ability to integrate with description logic reasoners such as Pellet. Ontology alignments were also developed at this stage and are managed under this component of the system framework [18]. In the second stage the ontological models were created for the specific application scenario (i.e. an office environment), with validation using the OWL2 specification. For the purposes of this paper, the application scenario is office based situation awareness.

B. *Framework Validation*

In order to simulate the chosen application scenario, data was generated using finite-state machines that simulated the behavior of inhabitants within the environment, in keeping with the case study outlined below.

1) *Case study*

a) Michael and Emma are lecturers at the same university. Michael has a project meeting that both he and Emma are scheduled to attend at 2:10pm. The time is currently 2:12pm and Emma has still not arrived for the meeting. Michael is unsure whether to delay the meeting, as Emma will arrive shortly, or to begin the meeting as she is currently delayed.

b) Jane is requesting an unscheduled meeting with her lecturer Michael. Michael will be available in his office for the next 15 minutes before he must leave for a scheduled appointment. Jane is given a recommendation on how to proceed with the meeting based on Michael's current as well as near future situations.

c) Emma is late for a meeting and only wishes to attend if she can make it on time. She is informed of the current situation in the meeting room in regards to if the meeting has begun without her.

d) Kyle wishes to make a phone call to Jane. As her current situation is in a lecture the system delays recommending he make the call until a break in the lecture occurs or it ends.

Specifically finite-state machines were used to simulate behavior within a smart environment and to generate appropriate sensor events for each of the application scenarios. States are used to represent individual behaviors that inhabitants of the environment are likely to perform. Data is generated based upon transitions into these states, which is then used to populate instances in the context model. The information generated corresponds to general context that contributes to recognition of an event.

VI. RESULTS AND DISCUSSION

The application was used to simulate four scenarios as part of a case study, as described in Section V. The situation aware system aims to allow its users to make an appropriate decision on how to pursue their goal based on the current state of the environment. Specifically, the inference of situations under an evolving set of contextual conditions is investigated. Upon testing each of these scenarios, the aim was to infer the appropriate action a user should take given their stated goal and the current state of the environment. Finite state machines were used to perform event data generation for each scenario given a typical flow of events over a bounded time period. The results section details the inference process for the second scenario, described in the previous section and in Table I.

TABLE I. DESCRIPTION OF THE USER REQUESTS EVALUATED BY THE SYSTEM

Scenario	User	Goal	Constraint	Situation Progression	Action Recommended

3.	Emma	Has_Meeting_Begun	Spatial(OfficeMichael)
			Temporal(timeseg0-5)
				InOffice (time_seg0),
				InOffice (time_seg1),
				InOffice (time_seg2),
				InOffice (time_seg3),
				OutOffice(time_seg4)
					Procede_Office(time_seg0),
					Procede_Office(time_seg1),
					Procede_Office(time_seg2),
					Procede_Office(time_seg3),
					Reschedule_Meeting(time_seg4)

In this scenario the user Jane is searching for the status of her lecturer Michael. The system must assess Michael's situation to determine if he is both in the office and available for a meeting and relay this information back to her in form that it is intuitive to act upon. Table I depicts the situations that were inferred at specific time segments and the related action inferred in relation to the established situation. The constraints column represents that Michael must be in his office for the next 5 time segments in order for Jane to be able to meet him immediately.

Table II below describes the events that are occurring concurrently in each time segment that are related by the system to Michael's current activities. These events are recognized and parsed by the event model described earlier.

TABLE II. MEETING'S EVENT SEQUENCE

Time Segment	Concurrent Events
0.	OfficeDoorOpen ∧ LightOn ∧ Sit_Chair ∧ UsingTelephone ∧ MeetingTS20
1.	OfficeDoorOpen ∧ LightOn ∧ Sit_Chair ∧ UsingTelephone ∧ MeetingTS20
2.	OfficeDoorOpen ∧ LightOn ∧ Sit_Chair ∧ ¬UsingTelephone ∧ MeetingTS20
3.	OfficeDoorOpen ∧ LightOn ∧ Sit_Chair ∧ UsePC ∧ ¬UsingTelephone ∧ MeetingTS20
4.	OfficeDoorOpen ∧ LightOn ∧ Sit_Chair ∧ UsePC ∧ ¬UsingTelephone ∧ MeetingTS20

The detected events are made available to the situation model through ontology mappings between it and the event model consisting of the OWL class <owl:class ID="Event">. Each situation is a combination of events occurring either concurrently, sequential or interleaved between time segments. Time segment one from Table II demonstrates that the conjunction of concurrent events OfficeDoorOpen, LightOn, SitChair and UsingTelephone as well as the interleaving of a planned event from time segment 20 MeetingTS20 within a single situation specification. Time segment two demonstrates the negation of an event occurring during a situation, UseTelephone, which is an important indicator that Michael is not currently occupied and available for a meeting during that time slot.

The situation inference process is rule based as described previously. The rules make use of both time segments and location based parameters to provide context as to the relevance of events to situations. Figure 4 below provide an excerpt of the rules used in the situation model. This rule depicts the logic required to infer the situation InOffice in time segments two and three.

CompSituation(?comsit), TimeSeg(?timsegx), TimeSeg(?timsegy), Person(?person), Event(?door), Event(?light), Event(?sit), Event(?pc), EventNeg(?usephone), Location(?l), hasLocation(?door, ?l), hasLocation(?light, ?l), hasLocation(?sit, ?l), hasLocation(?pc, ?l), hasEvent(?comsit, ?door), hasEvent(?comsit, ?light), hasEvent(?comsit, ?sit), hasEvent(?comsit, ?pc), hasNegEvent(?comsit, ?phone), hasTime(?door, ?timsegx), hasTime(?light, ?timsegx), hasTime(?sit, ?timsegx), hasTime(?pc, ?timsegx), hasTime(?usephone, ?timsegx),

-> hasSituation(?person, ?comsit), hasLocation(?person, ?l), hasBeginning(?comsit, ?timsegx), hasEnd(?comsit, ?timsegy)

Fig. 4. Situation Inference Rule Example

A link between situations inferred within the situation model and the application model that uses them is performed through an ontology mapping based upon the OWL class <owl:class ID="Situation">. Section III and Figure 4 provide a description of the action inference process given knowledge of the users goal and situations that have occurred. As Table I depicts given knowledge of the situation the application model can infer the appropriate action to be recommended to the user in order to fulfill their goal. When Michael is in his office and not busy (in this scenario not on the phone as Table II events describe), he is inferred to be in office and available for a meeting and hence the action is recommended to the user Proceed_Office.

The results demonstrate the inference of a user specific action related to a high level situation recognized to be relevant to their goal using ontological reasoning and SPARQL rules, and inferred from low level events using ontological reasoning.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents an approach to application led situation awareness using a flexible framework of knowledge based reasoning components. In order to achieve this we have proposed a method of decoupling context recognition, situation inference and application oriented situation interpretation among three individual ontological models. As demonstrated, our approach can be used to identify situations of varying complexity, incorporating both temporal and spatial knowledge. Furthermore, the system is capable of understanding the meaning of a situation with respect to user goals. It is anticipated that the approach described herein can be integrated and adapted into a range of pervasive domains and application scenarios.

Validation of the framework has been made using a smart office based scenario. A case study featuring a realistic usage scenario has been presented, demonstrating application led situation recognition and interpretation. The use of rule-based reasoning allowing more complex inference under highly dynamic conditions has been emphasized, facilitating more accurate and powerful reasoning. This process can be automated to allow rule generation. Initial validation of this approach to situation awareness indicates that the current application, using ontology models and a personalized situation interpretation mechanism can deliver accurate and reliable guidance on how inhabitants of an environment should pursue their goals. The works aims to highlight the utility of situation awareness within pervasive environments and the research challenges associated with application-oriented use of this technology. Future work will continue to explore situation awareness from an application-oriented perspective. The knowledge engineering effort involved in general domain knowledge abstraction, situation specification and situation relation is considerable. Ontological rule based specification is built heavily upon expert knowledge and requires considerable effort during initial development and maintenance of functional models. In future, we will investigate automated model construction, adaption and rule generation to alleviate these issues. We also intend to improve the situation context relation and modeling through the use of semi-formal models such as the Petri-Net that can specify the relationship between dynamically changing context with respect to context processing for situation awareness. Finally, work will continue on an approach that aims to provide a systematic method of handling uncertainty under a knowledge based reasoning process.

REFERENCES

[1] D. J. Cook and S. K. Das, “How smart are our environments? An updated look at the state of the art,” *Pervasive and Mobile Computing*, vol. 3, no. 2, pp. 53–73, 2007.

[2] M. R. Endsley, “Theoretical underpinnings of situation awareness: a critical review,” in *Situation awareness analysis and measurement*, 2000, pp. 3–32.

[3] L. Chen and C. Nugent, “Situation aware cognitive assistance in smart homes,” *Journal of Mobile Multimedia*, vol. 6, no. 3, pp. 263–279, 2010.

[4] J. Ye, G. Stevenson, and S. Dobson, “KCAR: A knowledge-driven approach for concurrent activity recognition,” *Pervasive and Mobile Computing*, vol. 19, pp. 47–70, 2014.

[5] J. Ye, S. Dobson, and S. McKeever, “Situation identification techniques in pervasive computing: A review,” *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 36–66, Feb. 2012.

[6] L. Chen, C. D. Nugent, and H. Wang, “A knowledge-driven approach to activity recognition in smart homes,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961–974, Jun. 2012.

[7] N. Baumgartner, S. Mitsch, A. Müller, W. Retschitzegger, A. Salfinger, and W. Schwinger, “A tour of BeAware – A situation awareness framework for control centers,” *Information Fusion*, vol. 20, pp. 155–173, Nov. 2014.

[8] M. G. C. a. Cimino, B. Lazzerini, F. Marcelloni, and A. Ciaramella, “An adaptive rule-based approach for managing situation-awareness,” *Expert Systems with Applications*, vol. 39, no. 12, pp. 10796–10811, Sep. 2012.

[9] G. D’Aniello, V. Loia, and F. Orciuoli, “A multi-agent fuzzy consensus model in a Situation Awareness framework,” *Applied Soft Computing*, vol. 30, pp. 430–440, 2015.

[10] G. Tamea and M. Cusmai, “Situation awareness in airport environment based on Semantic Web technologies,” in *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2014, pp. 174–180.

[11] M. M. Kokar, C. J. Matheus, and K. Baclawski, “Ontology-based situation awareness,” *Information Fusion*, vol. 10, no. 1, pp. 83–98, Jan. 2007.

[12] M. M. Kokar and M. R. Endsley, “Situation awareness and cognitive modeling,” *IEEE Intelligent Systems*, vol. 27, no. 3, pp. 91–96, May 2012.

[13] J. F. Allen and G. Ferguson, “Actions and Events in Interval Temporal Logic,” *Journal of logic and computation*, vol. 4, no. 5, pp. 531–579, 1994.

[14] G. Okeyo, L. Chen, and H. Wang, “Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes,” *Future Generation Computer Systems*, vol. 38, no. 1, pp. 1–15, Mar. 2014.

[15] K.-L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, W. Burns, and I. Solheim, “Ontological user modelling and semantic rule-based reasoning for personalisation of Help-On-Demand services in pervasive environments,” *Future Generation Computer Systems*, vol. 34, pp. 97–109, May 2014.

[16] I. Horrocks, P. F. Patel-schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL : A Semantic Web Rule Language Combining OWL and RuleML,” *W3C Member submission 21*, no. May, 2004.

[17] W3C OWL Working Group, “OWL 2 Web Ontology Language Document Overview,” *OWL 2 Web Ontology Language*, no. December, pp. 1–7, 2012.

[18] J. Euzenat and P. Shvaiko, *Ontology Matching*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.