AN ONTOLOGY-BASED MODEL FOR CONTEXT-AWARE INDOOR NAVIGATION

Umi Laili Yuhana[1,2], Wan-rong Jih[2], Han-Wen Chang[2], Jane Yung-jen Hsu[2], Wenn-Chieh Tsai[2]
[1] Department of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember (ITS), Kampus ITS Sukolilo, Surabaya, 60111, Indonesia
[2] Department of Computer Science and Information Engineering, National Taiwan University, 1, Roosevelt Rd. Sec. 4, Taipei, Taiwan
email: yuhana@if.its.ac.id[1,2], jih@agents.csie.ntu.edu.tw[2], r96922005@ntu.edu.tw[2], yjhsu@csie.ntu.edu.tw[2], d96944012@ntu.edu.tw[2]

ABSTRACT

In traditional indoor navigation, the starting and ending node of a route must be precisely assigned. We adapt context into tradition indoor navigation problem, and provide services that allow the users assign the destination of their requests imprecisely. Technologies of ontology reasoning are used to infer the users' requests. Finally, a scenario describes the navigation processes and shows the concept of knowledge, relationship between concepts, and the inference rules.

Keywords: ontology, context aware, indoor navigation

1 INTRODUCTION

In the real world, there are some problems need to be solved to give better navigation. For example, in campus and office buildings, some doors are opened in the daytime of weekdays but locked in the night and weekend. It means that at different time, there are different routes to go among different locations. Another situation in academic environment is that a professor will give lectures or have meetings at scheduled time and at specific places. Persons who want to meet her must know her location based on related event schedules. Hence, a navigation system that can deal with different context overtime by utilize calendar of events is necessary to support the demand.

There has been several navigation systems are build for navigation and guidance in indoor environment. Many research[1, 2] focus on the human-computer interaction to provide users with different conditions of a better navigation paradigm. Another research[3, 4] use different models and structures to capture the location information and other context. However, there are two assumptions in these work that the precise location destination is known or given, and the environment doesn't change.

A context-aware indoor navigation should be able to deal with the changes of the environment and the imprecise queries. With the time and event information, the real destination of a person or event-oriented query can be inferred. This work uses ontology and reasoning to model the context, find the routes, and provide the indoor navigation, even though the user may not know the exact location of her destination.

In this paper, researches on the indoor navigation problems will be discussed in the next section. Technologies that apply to our research are shown in Section 3. We define the context-aware indoor navigation in Section 4. Our knowledge representation model and system architecture are introduced in Section 5 and 6. Section 7 shows a scenario to describe the processes of our system. Finally, a discussion and conclusion are given in Section 8.

2 RELATED WORK

People have been working on intelligent routing, navigation, and guidance systems for a long time. Cyberguide[5] develops a location-aware guidance application supports indoor and outdoor tour including positioning, map, and information components in campus. NAVIO[6] focuses on the integration and fusion of pedestrian user tracking between indoor and outdoor location. These researches can only deals with the static environment, which cannot infer the destination of a path while the precise destination is unknown.

Recently, researchers adapt the idea of context-awareness to create better navigation services. OntoNav[1] provides a user-centric navigation paradigm for indoor environments based on the user's physical and perceptual capabilities and limitations by aware of the user capabilities such as normal, disable person or person bring many thing in her hand. Indoor wayfinding for people with impairment[2] has done in user profile, time and location context, and develops a prototype of user interface. These researches are focus on the human computer interface face design and do not consider the model of navigation problems.

To model the context information and provide navigation service, researchers use different representations and methods. CoINS[3] uses convex hull and quadtree to represent the indoor locations, and uses the idea of adjacent path to develop the guidance service. Hsieh *et al*.[7] proposed hybrid path calculation model to solve indoor navigation problem. These researchers try to improve and solve the pathfinding algorithms by using their navigation models, which are not deal with the context handling.

Almaida *et al*.[4] treats spatial and time continuously and provides a service oriented architecture to sell and buy products, which uses ontologies for context modeling, including user profiles, dynamic behaviors, physiological and emotional status, products, locations, and time. However, their models did not support the capability of context reasoning.

3 TECHNOLOGY OVERVIEW

An overview of the context-aware systems, context models, and service-oriented architecture are introduced in this section

3.1 Context-Aware Systems

Context-aware systems take the contextual information into account and provide services to fulfill the needs of users. During the past years, a number of context-aware systems have been developed to support pervasive computing and ambient intelligent environments such as Active Badge location system[8], ParcTab[9], and Context Toolkit[10]. These systems utilize various sensors and devices to provide location-aware services but do not consider the issues of context reasoning.

EasyMeeting[11] is a prototype of an intelligent meeting room that built on a Context Broker Architecture (CoBrA), an agent-based broker that maintains all the context knowledge represented in RDF-triple and utilizes Jena and Jess to support context reasoning. The MyCampus[12] project has been developed to provide context-aware mobile services in the university and the e-Wallet is its key element that use rule-based reasoning to support access control of user's privacy. Both EasyMeeting and MyCampus deployed context reasoning on their systems but lack of a well-defined knowledge model to represent the contextual information.

3.2 Context Models

Many context-aware systems concentrate on location aware services. MINDSWAP Group at University of Maryland Institute for Advanced Computer Studies develops Semantic geoStu[1] to express basic geographic features such as countries, cities, and relationships between these spatial descriptors. The Open Geospatial Consortium, Inc. prescribes OpenGIS[2] specifications for GIS data exchange and process, and OpenCyc Spatial Relations[3] specify the vocabularies of spatial objects and relations.

Temporal reasoning plays an essential role in context-aware systems. OWL-Time[4] and ISO 8601 date and time formats are the popular structure and standard. Traditional time structure is based on a set of points, Bry *et al*.[13] introduce CaTTs and treat the cultural calendars as interval-based time. Ma and Hayes[14] analyze the temporal interval-based models in a recent literature report.

The RFC 2445[5] defines iCalendar format for calendaring and scheduling applications, which provides users to create personal activities. Google Calendar[6] is a popular web-based calendar supports iCalendar standard and users can share their own personal activities with others. These human activities are related to people, time, and location. Consequently, the contents of persons' schedules can help us to derive their location at a given time.

3.3 Service-oriented Architecture

A context-aware system must react immediately to adapt the changes of environmental context. For dynamically adapting the context, a modular design approach is paramount needed. The concept of service-oriented architecture satisfies

[1] http://www.mindswap.org/2004/geo/geoStuff.shtm
[2] http://opengeospatioal.org/standards/
[3] http://www.cyc.com/cycdoc/vocab/spatial-vocab.html
[4] http://www.w3.org/TR/owl-time/
[5] http://tools.ietf.org/html/rfc2445
[6] http://calendar.google.com

such requirement. Systems are built in independent, loosely coupled pieces of services that achieve a specific, coarse-grained functionality. Each service is developed independently, and can be invoked by other services or clients using a declarative description of access points and accepted messages. Multi-agent systems can be investigated as a complement of service-oriented architecture. Autonomous agents are distributed in the system and interact with each other in order to achieve goal.

4 CONTEXT AWARE INDOOR NAVIGATION

An indoor environment can be represented as $G = (V, E)$, where the node set $V = \{v_0, v_1, v_2, \dots, v_n\}$ is a set of indoor places and the edge set $E = \{(v_i, v_j) \mid (v_i \neq v_j), v_i, v_j \in V\}$ is a set of route segments that connect two indoor places.

Given the source node *s* and destination node *d*, an indoor navigation service provides a route from place *s* to *d* and every route segment is associated with a direction. We define a route is a sequence of $r_1 \xrightarrow{o_1} r_2 \xrightarrow{o_2} \dots \xrightarrow{o_{d-1}} r_d$, where $r_1, r_2, \dots, r_d \in V$, $s = r_1$, $d = r_d$, $(r_1, r_2), (r_2, r_3), \dots, (r_{d-1}, r_d) \in E$, and the directions $o_1, o_2, \dots, o_{d-1}$.

We add context-aware ability to the original indoor navigation service. When the user is located in the building, the source node *s* can be assigned as the current location of this user. The indoor navigation is to help the new visitors in a building to go to their target destination, which can be either a person or a location. If the target is a person, his location may change due to his schedule. For example, a professor may move around to his office, laboratories, classrooms, or seminar rooms. Therefore, the destination node *d* for a context-aware indoor navigation, is not a fixed location but is inferred from the context of the target. Clearly, our goal is to provide a navigation service that starting from the user's current location to the location of his target. At different time, the target location is different based on context.

5 ONTOLOGY MODELING

Ontology is explicit and formal specification of a conceptualization that consists of finite list of terms and the relationships between these terms[15]. Ontology can be used as data model that represents a domain and it provides knowledge for reasoning about the objects and relations in the domain. Figure 1 depicts our ontology as representation of common concepts about indoor location navigation and suitable for route searching.

Context information are collected from real-word classes such as Event, Sensor, Time, Location, and Person. We adopt these contexts and provide context-aware service navigation that is represented in Navigation concept.

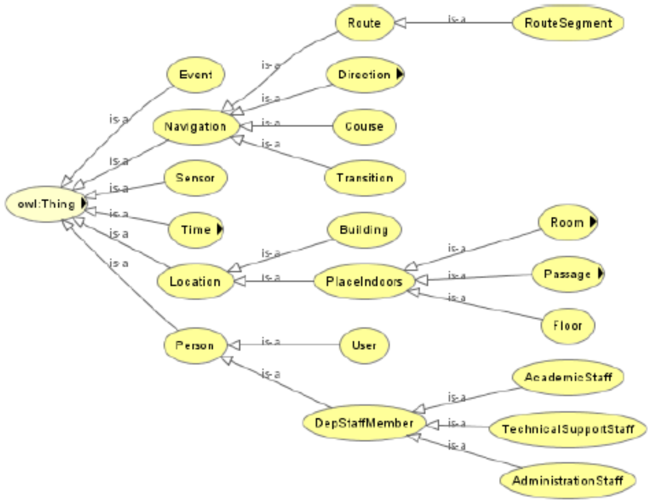

Figure 1. An Indoor Navigation Ontology

The class hierarchy represents a subclass relationship; an arrow points from a subclass to another superclass. As an example we can see in Figure 1, the indoor place class is a concept of the location, that *is - a* is represent the subclass relationship.

In Figure 1, concept *Route* is starting from a source node *s* to a destination node *d*, where the node *s* and *d* are the instances of *PlaceIndoors*. Suppose that a route $r_1 \xrightarrow{o_1} r_2 \xrightarrow{o_2} \dots \xrightarrow{o_{d-1}} r_d$ consists of several connected route segments that represented in *RouteSegment* class. Therefore, the edges (r_1,r_2), (r_2,r_3), ...,(r_{d-1},r_d) and direction $o_1, o_2, \dots, o_{d-1}$ are the instances of *RouteSegment* and *Direction*, respectively. As a route is a sequence of connected edge, a relationship *connectWith* describes the connectedness between two nodes is required; Figure 2 shows the *connectWith* is the relation between two instance of *Location*.

The source node s of a navigation route is getting from the user's position, which can be obtained from a location sensors[16]. For example, a *Person* John can be identified by a *Sensor* sensor1, is described by the relationship *representedBy*. Property *isMovingTo* reflects the movement of John; when John is moving to a location, the sensor that he carried will detect the movement. The property *isLocatedIn* represents that a person is located in a place.

If the target of a navigation route is a person, a context-aware service has to check the schedule of the target person. In Figure 2, property *participatedIn* describes that an instance of Person is participated in an *Event*.

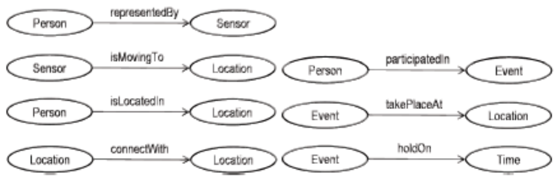

Figure 2. Relationships between Ontology Classes

The Event instance will take place at a given *Location* on a specific *Time*. That is, the property *takePlaceAt* and *holdOn* represents the "where" and "when" relationship of an event, respectively.

6 SYSTEM ARCHITECTURE

Figure 3 is our system architecture that consists of the following building blocks. GUI will send user request to Context Aware Service Platform. Context Collection Agents send the contexts from Context Resources to Context Aware Services Platform. Context Aware Services Platform uses these contexts to find the route and navigate the user through the GUI. All these agents use message to communicate with each others.

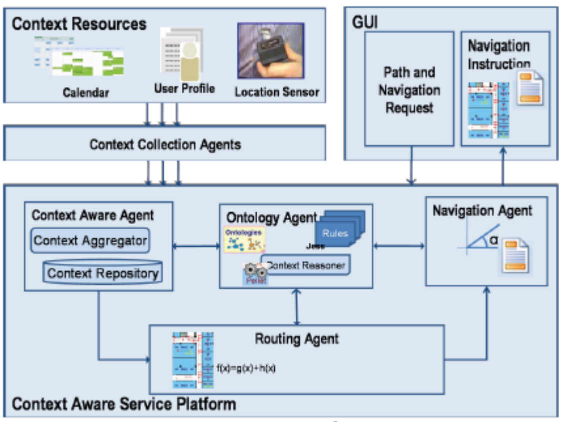

Figure 3. System Architecture

GUI (Graphical User Interface) : GUI is the main interface between user and the system, it receives user request and responds with navigation instruction based on user, location, events, and time context.

Context Resources : Context resources can be obtained from either software or hardware sensors. For context-aware indoor navigation, the contexts can be achieved from calendar, user profile, and location tracking sensors. While these sensors percept a new contextual information, they will send this context to Context Collection Agents.

Context Collection Agents : Context collection agents obtain raw contexts from Context Resources. Raw context refers to data obtained directly from context sensors, such as user location from the location tracking sensor, the user profile, and a meeting schedule from the calendar. Every raw context is converted into a semantic representation and sent to Context Aware Service Platform.

Context Aware Service Platform : This is the core of our system architecture, contains the following agent: context aware agent, ontology agent, route agent, and navigation agent.

- **Context Aware Agent** receives contextual information and maintains the consistency of contexts. This agent consists Context Aggregator and Context repository. The Context Aggregator collects contexts from Context Collection Agents and Ontology Agent. All the contexts will be stored in Context Repository.
- **Ontology agent** uses the ontology of section 5 to convert the sensor location into an indoor place. In addition, a context reasoner combines a rule-based engine Jess[7] and a DL reasoner Pellet[8] to perform the query matching and infer high-level context. For example, to query a person's location, a rule will be triggered for matching the time and schedule to derive the person is participated in what kind of activity and where is this person.
- **Routing Agent** generates a route from start node to end node. A route contains route segments with direction instructions will be delivered to Navigation Agent.
- **Navigation Agent** will send the navigation instructions to GUI. When the user is moving to a place, the Navigation Agent will calculate the corresponding orientation and direction based on the given sequence of route segments.

[7] http://herzberg.ca.sandia.gov/
[8] http://pellet.owldl.com

7 A DEMONSTRATION SCENARIO

We use an example to illustrate detailed processes of how to use ontology to deliver context-aware indoor navigation service.

Professor Jane is the advisor of iAgent group in iSpace Lab at Computer Science of National Taiwan University. She is always busy in teaching, meeting, and discussing research with her students. When a visitor John wants to meet with Professor Jane, he needs a context-aware navigation service to guide him to reach the location of Professor Jane.

Figure 4 shows the process of how to help John to find Professor Jane.

Figure 4. Sequence of Tasks in Navigation System

7.1 Determine Starting Node

Suppose that a location tracking sensors[16] can detect the movement of John. Figure 5 shows the sequence follow to detect the location of a user. This task involves Context Collection Agent, Context Aware Agent, and Ontology Agent. Figure 6 describes context and knowledge are represented in RDF-triple, which can be represented as subject, predicate, and object format. Subject is resources named by an URI with an optional anchor identity. The predicate is a property of the resource, and the object is the value of the property for the resource.

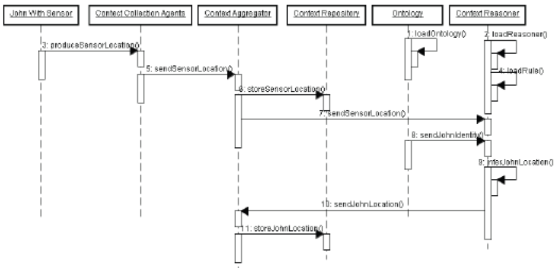

Figure 5. A Sequence Diagram to Determine User's Location

Sequence 3 of Figure 5 describes that when John moves to a location, the sensor sends its location to a Context Collection Agent. The second row in Figure 6 represents *sensor1* is moving to *loc1*. From Figure 1 and Figure 2, *sensor1* and *loc1* are instances of class *Sensor* and *Location*, respectively.

Context Name	Subject	Predicate	Object
Sensor's Location	<http://....#sensor1>	<http://.....#isMovingTo>	<http://......#loc1>
John's Identity	<http://....#John>	<http://.....#representedBy>	<http://......#sensor1>
John's Location	<http://....#John>	<http://.....#isLocatedIn>	<http://......#loc1>

Figure 6. Context in RDF-triple

In the Context Aware Agent, Context Aggregator will receive and store this new context to Context Repository. Sequence 5-7 of Figure 5 shows Context Collection Agent send the sensor data to Context Aware agent and Ontology Agent.

While Ontology Agent receives the raw sensor data, a query will match that John is identified by *sensor1* shows in third row of Figure 6. In addition, Ontology Agent contains a context reasoner, it uses rules to infer new contexts. Rules of a rule-based system serve as IF-THEN statement. Context reasoner load rule and use it to infer new context. Figure 7 is a rule to locate a person. Patterns before → are the conditions, matched by a specific rule. On the other hand, patterns after → are the statements that may be fired, called right hand side (RHS) of rule. If all the conditions are matched, then the actions of RHS will be executed. The RHS statement can be asserted as new high-level contexts. The second and third rows of Figure 6 are match with the rule in Figure 7, therefore, the context reasoner will infer that "John is located in *loc1*" shows in fourth row of Figure 6. Sequence 1, 2, 4, 8, and 9 of Figure 5 depicts the processes of Ontology Agent.

representedBy(?user, ?sensor) ^ isMovingTo(?sensor, ?location)
→ isLocatedIn(?user, ?location)

Figure 7. Rule to infer User Location

Sequence 10-11 of Figure 5 illustrates when Ontology Agent derives John's location, it will be send to Context Aware Agent. When John sends his request for locating Professor Jane, the current location of John in Context Aware Agent will be the starting node of a route.

7.2 Determine Destination Node

Given the destination is Professor Jane's location, Ontology Agent must to deduce her location from her calendar. Figure 8 shows if Professor Jane has a meeting she will be in the place where the calendar recorded. This event and location relationship has been defined in Figure 2, that is, the *takePlaceAt* relationship.

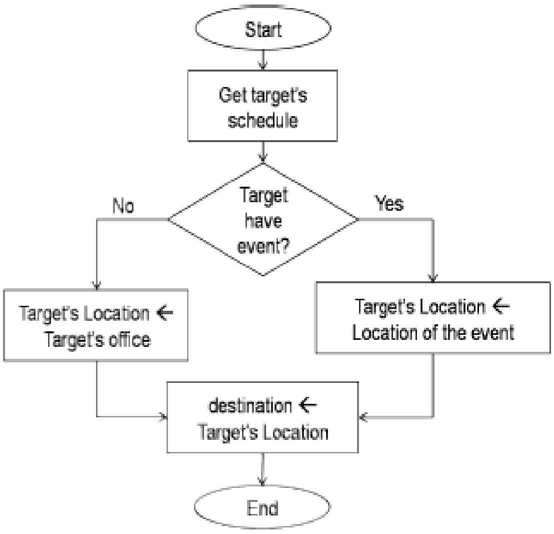

Figure. 8. Flowchart to determine ending node *d*

When the location of Professor Jane has been inferred, this location is taken as the destination node a route.

7.3 Generate Route

Routing Agent receives the starting node *s* and destination *d* from Context Aware Agent and Ontology Agent, respectively. Use the LRTA* algorithm[17] as the routing algorithm where the distance from starting node to current node is the real cost and the straight line distance from current node to destination node is the heuristic function.

Like the breadth-first search, Route Agent will find the available edges to connect a route. As the security reason, after office hours, campus buildings will lock some entrance doors and left one door for free access. In addition, the elevators or rooms may need access card to enter during the security control hours. These security control information can be defined as the attributes of buildings, rooms, and passages. By applying the similar approach of Figure 8, system uses the knowledge of available time to decide whether the edge is available or not.

We define the *connectWith* as the connectedness relationship between locations in Figure 2. An attribute of route segment class *RouteSegment* in Figure 1 describe the available time of the edges. Therefore, Routing Agent will produce a route from place *s* to place *d* and every edge of this route is connected.

7.4 Perform navigation

Navigation Agent uses the definition of direction in Figure 1 to calculate the navigation directions and delivers the instructions to GUI based on the sequence of route segments, which is send from Route Agent. The orientation instruction is depended on the location of user and his movement.

8 CONCLUSION AND FUTURE WORK

This research investigates an ontology model to provide context-aware indoor navigation service, which allows the target of navigation is imprecise. Traditional indoor navigation systems must precisely define the starting and ending node of a route, while context-aware indoor navigation allow the users assign their destination as a person or location target.

We propose an ontology model that can support indoor navigation with context-awareness. This ontology model can derive user's location and deduce the activity that a person currently participated, whereas traditional indoor navigation models do not have such capability of reasoning.

In addition, context-aware path finding must deal with the edge availability problem, that is, two places is not always connected or disconnected and the edge status is depended on the context. Traditional models define node and edge relationships statically and cannot adopt context into their models. We introduce the connectedness relationship between two places and this relation can be changed by using context reasoning.

As the routing algorithm provides the path for the navigation direction, it is possible to explore pathfinding algorithms to improve the routing process. While some of indoor navigation systems are focus on helping people with cognitive impairment, we will try to address this issue to our problem. Finally, we propose a scenario to describe context-aware indoor navigation in campus building, if there are more scenarios that will help us to evaluate our research.

REFERENCES

[1] Anagnostopoulos, C., Tsetsos, V., Kikiras, P., Hadjiefthymiades, S.P.: OntoNav: (2005) A semantic indoor navigation system. In: Proceeding of the first International Workshop on Managing Context Information in Mobile and Pervasive Environments.

[2] Liu, A.L., Hile, H., Kautz, H., Borriello, G., Brown, P.A., Harniss, M., Johnson, K.: (2006) Indoor wayfinding:: developing a functional interface for individuals with cognitive impairments. In: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility (Assets '06), Portland, Oregon, USA, ACM Press: 95 - 102

[3] Lyardet, F., Grimmer, J., Muhlhauser, M.: (2006) CoINS: Context sensitive indoor navigation system. In: Proceeding of the Eight IEEE International Symposium on Multimedia (ISM'06).

[4] de Almeida, D.R., de Souza Baptista, C., de Andrade, F.G.: (September 2006) Using ontologies incontext-aware applications. In: Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA '06), IEEE Computer Society 349{353 Washington, DC, USA.

[5] Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., Pinkerton, M.: (October 1997) Cyberguide: a mobile context-aware tour guide. Wireless Networks 3(5) 421 - 433

[6] Retscher, G., Thienelt, M.: (November 2004) NAVIO - a navigation and guidance service for pedestrians. Journal of Global Positioning Systems (CPGPS) 3(1-2) 208-217

[7] Hsieh, W.T., Miller, N., Yang, Y.F., Chou, S.C., Steenkiste, P.: (2004) Calculating walking distance in a hybrid space model. In: First International Workshop on Advanced Context Modelling, Reasoning And Management.

[8] Want, R., Hopper, A., Falc~ao, V., Gibbons, J.: (January 1992) The active badge location system. ACM Transactions on Information Systems (TOIS) 10(1): 91-102

[9] Schilit, B., Adams, N., Want, R.: (1994) Context-aware computing applications. In: Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US: 85 - 90

[10] Dey, A.K., Abowd, G.D., Salber, D.: (2001) A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction 16: 97-166

[11] Chen, H., Finin, T., Joshi, A., Kagal, L., Perich, F., Chakraborty, D.: (November-December 2004) Intelligent agents meet the semantic web in smart spaces. IEEE Internet Computing 8(6) 69-79

[12] Gandon, F., Sadeh, N.M.: (October 20-23 2003) A semantic e-Wallet to reconcile privacy and context awareness. Lecture Notes in Computer Science: The SemanticWeb (ISWC 2003) 2870: 385-401

[13] Bry, F., Ries, F.A., Spranger, S.: (2005) CaTTS: calendar types and constraints for web applications. In: Proceedings of the 14th international conference on World Wide Web (WWW '05), Chiba, Japan, ACM Press : 702-711

[14] Ma, J., Hayes, P.: (2006) Primitive intervals versus point-based intervals: Rivals or allies? The Computer Journal **49**(1): 32-41

[15] Noy, N.F., McGuinnes, D.L.: (2001) Ontology development 101 : A guide to creating your first ontology. Technical report, Stanford University

[16] You, C.w., Chen, Y.C., Chiang, J.R., Huang, P., Chu, H.h., Lau, S.Y.: (2006) Sensorenhanced mobility prediction for energy-e_cient localization. In: Proceedings of Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2006). Volume 2., Reston, VA, USA : 565-574

[17] Russel, S.J., Norvig, P.: (2003) Informed Search and Exploration. In: Arti_cial Intelligence: A Modern Approach. Second edn. Prentice Hall : 94-136

[This page is intentionally left blank]