A Travel Situation Management Ontology

A Travel Situation Management Ontology

Dr. Paul O'Brien

Griffith University Business School

226 Grey Street, Qld, 4101, Australia

Ph: +61 7 3735 3250 Fax: +61 7 3735 3272

Email: p.obrien@griffith.edu.au

Abstract

Highly mobile people (HMPs), such as international executives, airline crew, international sportspersons and independent travellers require flexible, reactive service delivery due to their regularly changing location and activities and the lack of a wired network connection. A mobile service delivery system should be able to detect relevant travel-related events such as change of location, availability of new last-minute specials, sales opportunities and safety issues and then reactively take action in response to those events. This paper describes a generic travel situation management ontology that was developed in the Ontology Language for the World Wide Web (OWL) using the ontology development tool, Protégé. This ontology can be used as the basis for mobile travel service applications.

Keywords: Tourism technology Situation Management, Ontologies, Context Awareness, Mobility

1. Introduction

Today, personal computers are available that are powerful, portable, use little energy, do not need special environments and are cheap. The original computers were good for large scale information processing. They were not suited to the tasks that highly mobile people (HMPs) such as international executives, airline crew, international sportspersons and independent travellers typically carry out as they go about their work or play. “They have not been aware of our needs or even of whether we were in the room with them”(MIT, 2004). The next generation of computing devices will be faster, cheaper, smaller, more powerful, and more portable than their predecessors. Wireless data technologies are maturing and improving in bandwidth, performance and reliability (WiMAX, 2006) so the next generation of portable computing devices will be networked via high speed, ubiquitous wireless technologies.

1.1 Problems with Current Automated Service Delivery Techniques

Current automated service delivery systems tend to be location-aware, context-aware within a particular room or single location, or temporally aware at a single location or within a single system. Their designs also tend to be restricted by consideration of narrow mobile bandwidth and limited handset capability and are generally designed for a specific task or a specific domain. Context dimensions of time alone or location alone or even a combination of time and location do not allow independent decisions to be made about situations that occur. Location, type of location, time, type of time, local bindings and user preferences are required as a minimum for effective decision-making for HMPs. Domain specific and application specific dimensions are not generalisable to other domains or applications.

Application and domain knowledge is essential for any working system, however, this information can be provided by creating a domain and application specific instance of a generic situation ontology that includes, of necessity, ontology elements that relate to context. By designing the architecture and system logic to be driven by a situation ontology, new domains and applications can be accommodated by creating a new *instance* of the situation ontology, thus eliminating the need to hard-code what is essentially object data into the application itself.

To automate the delivery of services the system must also be able to identify situations that require action (reactive situations) from the set of existing situations, and apply the appropriate action rules.

The focus of this research is therefore the design of a generic travel situation management ontology that can be used as the basis for an architecture for a ubiquitous reactive travel service delivery system for HMPs.

2. Philosophy Of Approach

The key to the provision of an effective, high added value mobile service is the ability to detect *relevant* events and to independently take the action that is most appropriate for the resulting *situation*. This *situation triggered action* approach relies on active context filtering to minimise the processing of situations (O'Brien, 2006). A review of previous work suggests that an automated ubiquitous travel service delivery systems requires:

- An agreed ontology of situations that allows the specification of situations, contexts and situation action rules, and
- An agreed travel domain ontology.

The use of ontologies, particularly ontologies encoded in a language accepted by industry ensures that the data can be easily accessed (agreed syntax) and can be understood (agreed semantics). While both are important , using agreed semantics is particularly valuable when dealing with travel products that may be sold, delivered and used by people of different cultures, languages and backgrounds.

The travel situation management ontology described in this paper forms the basis for the implementation of a system that addresses these requirements.

3. Methodology

The goal of this work is to find a satisfactory solution, not necessarily the best or optimal solution from amongst all the possible solutions. Generalisability of the artefacts produced by this work is not a key consideration as they will be the starting point for a line of future research in situation aware mobile service delivery systems and it will be refined in future research. For these reasons, a *design research* approach, as proposed by Hevner et al (2004) has been used. Design research aims to "create innovative artefacts that extend human or organisational understanding or capabilities" (Hevner et al., 2004) .

Considering this approach, the following tasks were undertaken:

1. To develop an awareness of the problem, interviews were undertaken with a small number of HMPs;
2. One service was selected from each of Angehrn's (1997) four virtual market spaces.;
3. A generic situation management ontology was designed for use in mobile service delivery systems;
4. The generic situation management ontology was extended by adding travel domain classes using a typical HMP scenario.

4. Travel Situation Management

4.1 Situations

There are a number of different definitions of the concept of a *situation*. While they are similar, they differ according to the viewpoint of the researcher. As this work is from an information systems viewpoint, it has drawn upon the work of Barwise (1987), Akman and Surav (1996) , Adi et al (2002) and Fischer et al (2002). They provide definitions that initially appear to be quite different, however, they all take an information systems perspective, taking context into account, either explicitly or implicitly, and they all consider the state of things in contexts.

From a theoretical perspective, situations do not necessarily require any action. Situations that do require some action to be taken, that is, *reactive situations,* require action rules to be defined. In this work the following, slightly modified version of Barwise's definition (Barwise, 1981): "a situation is *a state of affairs in a particular context"* was adopted but it draws upon Adi et al's work (Adi et al., 2002) to define reactive situations.

Rules need to be defined to specify the appropriate actions that should be taken in response to occurrences of each reactive situation (Adi et al., 2002). Where the same response is required for a number of reactive situations, these situations form a *class* or *type* of situation (Cherry, 2001) for which reaction rules must be defined.

Barwise (1987) argues that situations are *first class citizens (or objects)*, that can have properties and relationships with other objects including other situations. The relationships can be expressed by constraints or background conditions. Barwise & Perry (1983) found that *states of affairs* required spatio-temporal parameters to address seemingly incoherent situations such as *Paul is talking* and *Paul is not talking* which can both be true in different contexts that are not always persistent. This supports Lenat's position that the truth of any state of affairs can only be determined if we know its context (Lenat, 1998).

4.2 Situation Theory

Situation theory was introduced by Barwise (1981). Barwise's ideas were then applied to logic by Barwise and Perry (1983). Perry (1993) then built on his early work with Barwise. He clearly distinguishes *situations* from *worlds*. "A world determines the answer to every issue, the truth-value of every proposition. A situation corresponds to the limited parts of reality we in fact perceive, reason about, and live in (O'Brien, 2006)". Consequently, situations only allow limited decision-making about events and states of affairs within particular contexts and domains.

Situation semantics is normally applied to the theory of natural languages, but it can be useful to define real world, context dependent situations. Situation semantics is based on the concept that we classify parts or views of reality by considering the things, their properties and the relationships between them in that part of reality. The events that occur in a situation alter the values of the properties of things in that part of reality, that is, they change the state of affairs, in that context. A *state of affairs* may exist within a context because of events that occur outside that context, but determination of whether a situation *exists* depends only on the *state of affairs* of the things within that context.

In situation theory, a key concept is that of a *type of situation*. For example, the type of situation in which your car has a flat tyre (S_0) and the type of situation in which a car can't move (S_1). One type of situation may be related to or depend on another. For example, every time one situation occurs, another situation will also occur. Barwise and Perry (1983) say that S_0 *involves* S_1; cars with flat tyres can't move. They call these situations *constraints*, that is, a car that has a flat tyre is *constrained* from moving.

Devlin (1991) built on the work of Barwise and Perry and introduced a modern notation and terminology for describing situations such as these. In Situation Theory, *infons* are basic discrete pieces of information. Using Devlin's (1991) notation, an infon is represented as:

$«P, a_1, \ldots, a_n, i»$

where P is an n-place relation, $a_1, \ldots, a_n$ are objects of the relation P and i is the polarity (0 or 1) indicating whether the relation holds.

Situations are related to infons by the *supports* operator ($\models$). s $\models$ α indicates that the infon α is supported in situation s, or in other words, s makes it the case that α is true. For example, s $\models$ «bites, Fido, Mary, l, t, 1 » represents the situation where Fido bites Mary at some location (l) and time (t).

Devlin's notation and terminology are used to represent situations in this paper with the exception that the words *True* or *False* are used to represent polarity rather than the numbers 1 and 0. This has been done simply for clarity and to avoid confusion related to the representation of Boolean values in different programming languages.

Devlin (1991) also specified a number of basic types that are necessary for situation specification. They include TIM to specify time, LOC to specify location, IND to specify parameters, REL to specify relationships, SIT to specify situations and INF to specify infons. For example,

Ss = [s' | s' $\models$ << slaps, a', b', l', t', 1 >>]

St = [s' | s' $\models$ << touches, a', b', l', t', 1 >>]

means that if a’ slaps b’ at l’ (a LOC type) & t’ (a TIM type) then a’ also touches b’ in the same l’ & t’.

Factual infons and constraints are also required because some facts are only true under certain background conditions. A constraint C may only allow situation S_2 to be true when S_1 is true if the background condition B is true, that is $C = S_1 \rightarrow S_2 \mid B$. For example, birds can only fly if there is an atmosphere (a background condition.).

Contexts provide a useful way to apply specific “rules and pre-suppositions related to a particular point of view” (Akman & Surav, 1996) to simplify the specification and detection of situations. When binding parameters in situations, “only objects that are in one of the current contexts need to be considered.” (Akman & Surav, 1996) Furthermore, contexts can be considered as objects, that is, as ‘first class citizens’. This means that they can be used in the same way as any other object so ‘they can be denoted by constants in the logical language” and “variables can range over them (Akman & Surav, 1996)”. Combining the work of Devlin and Akman & Surav, TIM and LOC were combined, together with other dimensions where necessary, into a single, complex type, context (CON). This simplifies the specification, filtering and comparison of situations.

4.3 Situation Management

Reactive systems are systems that react automatically to changes in the environment. They have been used in command and control, active databases, system management tools, customer relationship management systems and electronic commerce systems. This paper extends previous work that tended to provide mechanisms for reacting to single events only to one which reacts to situations.

To be able to detect situations some key information must be included in the situation definition or in related objects:

- The events that can participate in situation detection;

- Context during which situation detection is relevant;
- The semantic conditions that must be satisfied in order to detect a situation;
- Whether and under what conditions an event is consumed (Adi et al., 2002).

Adi et al's (2002) event class does not include any location attributes. For mobile users it must be extended to include at least location attributes and most likely, other attributes of context. This is particularly important when inferring events from the world's state and concrete events such as user movement. Adi et al's (2002) lifespan class defines a set of events that can initiate a lifespan, a set of events that can terminate a lifespan, conditions for initiation and termination and maximum lifespan length. To take account of location a locationspan class was added that has similar attributes to lifespan but which are related to location, namely,

- A set of events that can initiate a locationspan;
- A set of events that can terminate a locationspan;
- The conditions for initiation and termination of a locationspan;
- maximum locationspan radius;
- granularity of location.

A granularity of time attribute was also added to the lifespan class.

Adi et al's (2002) group collections of semantically associated event instances into event groups. They are used to match different event instances that refer to the same entity or concept. An event collection is the collection of event instances that must be considered for situation detection. They are only evaluated if they occur during the time that a context that is associated with situation detection is active. Event collections were used to group together sets of events that result in a context and/or situation transition. A single event collection may include temporal events, location-based events and other service and/or user events that each cause the same transition.

Although this work is based on Adi et al's (2002) concepts, there are many differences, namely,

- Context is used to filter out irrelevant situations rather than just lifespan;
- An object reference attribute is used to relate a situation to event collection objects rather than simply an event collection name;
- Background conditions associated with a situation are specified using references to situation objects rather than using a simple situation expression.
- Adi et al's (2002) Situation Manager uses lifespan to filter out irrelevant situations. If there is no open lifespan for a particular situation, the situation is ignored. This is clearly insufficient because a situation may still be irrelevant, even if there is no open lifespan, if there has been a change in some other dimension of context such as location, or even a change in user preferences. Therefore, for HMPs, situations need to be filtered by context, not just lifespan.

Cherry (2001) proposed the use of pre-specified rules, *situation-action rules*, to specify the actions that are required to be taken in response to the detection of a reactive situation. He shows that rules can be defined for classes, or types, of situations rather than individual situations, thereby reducing the number of rules required. This approach was taken, using Semantic Web Rule Language (SWRL) to specify the rules (SWRL, 2004).

5. Ontologies

In Artificial Intelligence (AI), an ontology is "the specification of a conceptualisation. That is, defined terms and the relationships between them, usually in some formal and preferably machine readable format" (Hendler, 2001). The AI concept of an ontology generally refers to "an engineering artefact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words (Guarino, 1998).” These formal ontologies provide a means of representing, or at least approximating, a conceptualisation of the world.

5.1 An Ontology of Situations

Matheus et al (2003) proposed a core ontology for situation awareness that defines a situation as a collection of goals, situation objects and relations between situation objects. Situation objects are “entities in a situation – both physical and abstract – that can have characteristics (i.e. Attributes) and can participate in relationships” (Matheus et al., 2003). This ontology was designed to be the core of a situation awareness system that assumes that the recognition of the existence of a particular situation will occur in the mind of a human operator who interfaces to the system through a human-computer interface. As some reactive situation management systems need to operate without human intervention, Matheus’s ontology was extended to include classes for user preferences.

Devlin (1991) defined a basic ontology of situation theory, based on Barwise and Perry’s work (Barwise, 1981; Barwise & Perry, 1983) and documented it using a new notation that he developed. Devlin’s situation theory ontology was later used to develop a number of new situation-oriented languages (Tin & Ersan, 1995) including Baby-Sit (KEE, 1993).

Barwise & Perry’s situation theory and Devlin’s notation are now well accepted within the situation theory research community. The adaptation of this ontology for use on the worldwide web provides a solid theoretical basis for the development of a situation management ontology for use in semantic web applications. As OWL has become the language of choice for most of the recent work that involves the development of ontologies for the worldwide web (Fukazawa et al., 2006; Jakobson, Lewis, Matheus, Kokar, & Buford, 2005; Matheus, 2005) OWL was used to implement the core situation management ontology in this work.

5.2 Relationship between Events, Context and Situations

Barwise's TIM and LOC parameters, and a number of other dimensions of context were combined into a complex parameter, context (or CON). For example the following situation type, specified using Devlin's notation,

S = [ŝ | ŝ ⊨ «*cancelled*, Flight, ĺ, ť, True»]

could be rewritten as,

S = [ŝ | ŝ ⊨ «*cancelled*, Flight, c', True»]

Where S = situation type, ŝ = situation, ⊨ = supports, c' = context (composed of ĺ = location and ť = time) and « **….** » = an infon.

A service or HMP action causes an event. An event causes a change in HMP and/or a service state. The change in state may cause a change in context, which in turn may cause a change in situation. The change in state may also directly cause a change in situation *without* a change in context. A change in context may cause a change in situation. A *reactive* situation may occur subject to existence of a background situation. Similarly a reactive situation may occur due to a change in situation subject to a existence of a background situation. In response to the occurrence of a reactive situation one or more actions are taken.

An event that can cause a transition to a new context is included in an event collection for that transition. Background conditions cause normal and/or reactive situations to occur.

Table 1 summarises the relationship between events, states, contexts, situations and actions.

INSERT TABLE 1 HERE

5.3 Situation Management Example

A simple example of a typical mobile user scenario and how the situations that occur would be handled by the system are given below.

The date is 26 March 2007. Fred is traveling around the islands of the South Pacific on business. He is currently in Brisbane and is scheduled to travel to Vanuatu tomorrow and then on to the Solomon Islands on 28 march 2007. Fred is unaware that a serious conflict has broken out in Solomon Islands. He is not contactable because he is in a high level meeting. The US State Department issues a travel warning for the Solomon Islands. The system detects the warning, selects relevant contexts and checks to see if there are any HMPs registered in these contexts and sends the required alert through to them. The system then checks Fred's itinerary for planned Solomon Islands trips in the near future. The system identifies that a reactive travel safety situation exists for Fred's planned Solomon Islands trip and takes appropriate actions to cancel this leg of his trip. The next time Fred switches on his handheld device, the alert is displayed advising him of the situation and the changes to his itinerary.

The scenario above assumes that Fred has set his preferences to allow the system to automatically change his itinerary without consulting him. Each HMP will have different preferences regarding this and other automatic responses. HMP's must be able to set their own preferences regarding what actions the system is allowed to take before consulting them, including level of automation of itinerary changes, granularity of location and time for alerts, type of alerting preferred etcetera. The generic situation ontology described in this paper includes a class SERVICE_PREFERENCE that stores the HMPs preferences for a particular type of service, such as the Transport Service that would manage the situation described in the scenario above. The mobile situation management ontology must include information regarding these preferences so that system designers can use them to determine what, if any, independent action should be taken.

The class (or type) of reactive situation, itineraryChangeRequired (S_R1) can be specified by the following situation type definitions:

$S_1 = [\hat{s} \mid \hat{s} \models$ « *travelSafetyWarning*, c', True»]

$S_2 = [\hat{s} \mid \hat{s} \models$ «*hasFlightReservation*, p', c', True »]

$SR_1 = [\hat{s} \mid \hat{s} \models$ «*itineraryChangeRequired*, p', c', True »]

$C_1 = [S_1 \Rightarrow SR_1 \mid S_2]$

The *specific instances* of the situations that relate to the example scenario above are:

$s_1 \models$ « *travelSafetyWarning*, SolomonIslands070328, True»

$s_2 \models$ «*hasFlightReservation*, Fred, SolomonIslands070328, True »

$sr_1 \models$ «*itineraryChangeRequired*, Fred, SolomonIslands070328, True »

Although this is a very simple example it does demonstrate how the system would handle a typical HMP scenario.

6. A Travel Situation Management Ontology

A working travel situation management system requires the definition of a task ontology for situation management and a domain ontology for the travel domain A high level, or generic, situation management ontology can be specified independent of the domain in which it will be used. The primary concern of this high level situation management ontology is to specify the concepts that are important to the generic task of managing situations. Similarly, the travel domain ontology can be initially specified at a high level.

The high level situation management and travel domain ontologies must then be combined and specialised to create a travel situation management ontology. While the generic situation management ontology provides the underlying mechanism for reacting to any situation, the specific situations related to travel and the rules defining how to react to them must be specified within the travel domain ontology. Situation constraints can be specified within the ontology itself or they can be specified using RuleML (2003) or SWRL (2004).

The travel situation Management ontology is detailed below. It combines the generic situation management ontology and a travel domain ontology. Details of a concrete ontology containing instantiations related to the travel domain, detailed OWL specifications of these three ontologies, and sample rules defined in Semantic Web Rule Language (SWRL) are provided in (O'Brien, 2006).

6.1 Travel Situation Management Ontology

Brief descriptions of each of the classes in the travelsituation management ontology can be found in Table 2 below. More detailed descriptions of each of the classes and how they can be used, graphical representations of the ontology and the full OWL specification of the ontology can be found in (O'Brien, 2006).

Insert Table 2 Here

A graphical representation of the ontology is shown in Figure 1 below.

****Insert Figure 1 here***

6.2 Travel Management Situations

The generic travel situation ontology described above includes a number of infons and HMP, service and reactive situations. These are more formally specified below using a modified version of Devlin's situation notation (Devlin 1991).

6.2.1 Service Situations

SS = [ŝ | ŝ ⊨ « *serviceSituation*, c', True»]

SS_T = [ŝ | ŝ ╞ « *transportSituation*, c’, True»] ⇨ SS

SS_H = [ŝ | ŝ ╞ « *healthSituation*, c’, True»] ⇨ SS

SS_S = [ŝ | ŝ ╞ « *safetySituation*, c’, True»] ⇨ SS

SS_F = [ŝ | ŝ ╞ « *financeSituation*, c’, True»] ⇨ SS

SS_A = [ŝ | ŝ ╞ « *accommodationSituation*, c’, True»] ⇨ SS

SS_C = [ŝ | ŝ ╞ « *communicationSituation*, c’, True»] ⇨ SS

where SS = Service Situation Type, SS_T = Transport Situation Type, SS_H = Health Situation Type, SS_S = Safety Situation Type, SS_F = Finance Situation Type, SS_A = Accommodation Situation Type, SS_C = Communication Situation Type, ŝ = a situation, ⇨ = *implies,* and c’ = a context

.

This reads as SS is the set of situations where a service situation exists in context c’. Similarly SS_C is the set of situations where a communication situation exists in context c’ and that imply a Service Situation exists because a Communication Situation *ISA* Service Situation.All the following situation definitions use the same notation.

SS_SW = [ŝ | ŝ ╞ « *severeWeatherSituation*, c’, True»] ⇨ SS_S

SS_ST = [ŝ | ŝ ╞ « *highTerroristThreatSituation*, c’, True»] ⇨ SS_S

SS_HD = [ŝ | ŝ ╞ « *severeDiseaseOutbreakSituation*, c’, True»] ⇨ SS_H

SS_HW = [ŝ | ŝ ╞ « *highWaterRiskSituation*, c’, True»] ⇨ SS_H

SS_HF = [ŝ | ŝ ╞ « *highFoodRiskSituation*, c’, True»] ⇨ SS_H

SS_TD = [ŝ | ŝ ╞ « *tripDelayedSituation*, c’, True»] ⇨ SS_T

SS_TC = [ŝ | ŝ ╞ « *tripCancelledSituation*, c’, True»] ⇨ SS_T

SS_TR= [ŝ | ŝ ╞ « *roadTrafficSituation*, c’, True»] ⇨ SS_T

SS_TN = [ŝ | ŝ ╞ « newTr*ipAvailableSituation*, c’, True»] ⇨ SS_T

SS_TP = [ŝ | ŝ ╞ « *tripPriceChangedSituation*, c’, True»] ⇨ SS_T

SS_TS = [ŝ | ŝ ╞ « *tripSeasonChangedSituation*, c’, True»] ⇨ SS_T

SS_AN = [ŝ | ŝ ╞ « *newAccommodationAvailableSituation*, c’, True»] ⇨ SS_A

SS_AP = [ŝ | ŝ ╞ « *accommodationPriceChangedSituation*, c’, True»] ⇨ SS_A

SS_AU = [ŝ | ŝ ╞ « *accommodationUnavailableSituation*, c’, True»] ⇨ SS_A

SS_FC = [ŝ | ŝ ╞ « *majorCurrencyChangeSituation*, c’, True»] ⇨ SS_F

SS_CW = [ŝ | ŝ ╞ « *workAvailableSituation*, c’, True»] ⇨ SS_C

SS_CF = [ŝ | ŝ ╞ « *friendNearbySituation*, c’, True»] ⇨ SS_C

6.2.2 Highly Mobile Person (HMP) Situations

HS_hasFltRes = [ŝ | ŝ ╞ «*hasFlightReservation*, p’, c’, True »] ⇨ HS

HS_hasAccRes = [ŝ | ŝ ╞ «*hasAccommodationReservation*, p’, c’, True »] ⇨ HS

HS_inLoc = [ŝ | ŝ ╞ «*inLocationn*, p’, c’, True »] ⇨ HS

HS_pastLoc = [ŝ | ŝ ╞ «*pastLocation*, p’, c’, True »] ⇨ HS

HS_futureLoc = [ŝ | ŝ ╞ «*futureLocation*, p’, c’, True »] ⇨ HS

HS_hasMobile = [ŝ | ŝ ╞ «*hasMobile*, p’, c’, True »] ⇨ HS

HS_mobilePref = [ŝ | ŝ ╞ «*mobilePreference*, p’, c’, True »] ⇨ HS

HS_futureLoc = [ŝ | ŝ ╞ «*futureLocation*, p’, c’, True »] ⇨ HS

HS_hasFoodPref = [ŝ | ŝ ╞ «*hasFoodPreference*, p’, c’, True »] ⇨ HS

HS_friendAlert = [ŝ | ŝ ╞ «*enabledFriendAlert*, p’, c’, True »] ⇨ HS

HS_allowLoc = [ŝ | ŝ ╞ «*allowLocationDetection*, p’, c’, True »] ⇨ HS

HS_jobAlert = [ŝ | ŝ ⊨ «*enabledJobAlert*, p', c', True »] ⇨ HS

6.2.3 Reactive Situations, Constraints and Background Conditions

S_R1 = [ŝ | ŝ ⊨ «*itineraryChangeRequired*, p', c', True »]

C1 = [SS_H ⇨ S_R1 | HS_hasFltRes ∨ HS_inLoc ∨ HS_futureLoc]

C2 = [SS_S ⇨ S_R1 | HS_hasFltRes ∨ HS_inLoc ∨ HS_futureLoc]

C3 = [SS_T ⇨ S_R1 | HS_hasFltRes ∨ HS_inLoc ∨ HS_futureLoc]

where S_R1 = reactive situation (1), p' = a person, ∨ = logical OR and | = subject to the condition on the right hand side.

The first definition reads as the set of reactive situations S_R1 is the set of situations ŝ where an itinerary change is required for p' in context c'. The following three lines define constraints that apply to certain service situations implying that S_R1 is true. The second definition,

C1 = [SS_H ⇨ S_R1 | HS_hasFltRes ∨ HS_inLoc ∨ HS_futureLoc] reads as, a health service situation (SS_H) in a particular context implies that the reactive situation S_R1 is true in that context subject to the background condition that a HMP has a flight reservation for that context OR a HMP is in that context (in the location at the same time) OR a HMP will be in that context in the future. Similarly for the other constraint and reactive situation definitions above and below.

S_R2 = [ŝ | ŝ ⊨ «*accommodationChangeRequired*, p', c', True »]

C4 = [SS_A ⇨ S_R2 | HS_hasAccRes]

C5 = [S_R1⇨ S_R2 | HS_hasAccRes]

S_R3 = [ŝ | ŝ ⊨ «*alertRequired*, p', c', True »]

$C6 = [\text{SS\_H} \Rightarrow S_R3 \mid \text{HS\_hasRes} \vee \text{HS\_inLoc} \vee \text{HS\_pastLoc} \vee \text{HS\_futureLoc}]$

$C7 = [\text{SS\_S} \Rightarrow S_R3 \mid \text{HS\_hasRes} \vee \text{HS\_inLoc} \vee \text{HS\_futureLoc}]$

$C8 = [\text{SS\_F} \Rightarrow S_R3 \mid \text{HS\_hasRes} \vee \text{HS\_inLoc} \vee \text{HS\_futureLoc}]$

6.2.4 Travel Domain Events

Situations occur as a result of a change in the state of affairs (infons) in a context. This can be caused by a change in context or a change in the state of affairs within the same context, or both. A change in context results from certain temporal, HMP or service events. Those of interest in the travel domain include, but are not limited to:

- Fare Paid
- ReservationDepositPaid
- SMSAlertsEnabled
- WAPAlertsEnabled
- FriendReservationMade
- FlightArrived
- FlightDeparted
- FlightDelayed
- FlightCancelled
- RestaurantReservationMade
- AccommodationReservationMade
- LocationChanged

- ExchangeRateServiceEnabled
- TrafficHazard
- WeatherChange
- SecurityAlertsEnabled
- WeatherAlertsEnabled
- ClockTickSameDay
- ClockTickNewDay

6.3 Generic Situation Action Rules

The situation-action rules are stored on a server and are accessed via a unique URI that is encoded into the ACTION_RULE class. SWRL is recommended for all future rule development as it is now a World Wide Web Consortium Recommendation and is therefore likely to become the defacto standard in the future. The logic of reactive situation detection and action can be described as:

```
if ServiceSituation(c) then
  for all HMPSituations(c’) or ServiceSituation (c’) where (c = c’ or  c intersects c’)
     if HMPSituations(c’) is a background condition on ReactiveSituation (c’) or
               ServiceSituations(c’) is a background condition on ReactiveSituation (c’)
          then ReactiveSituation(c) is true
     endif
endif
```

where ServiceSituation(c) is a situation that exists because of a HMP's state of affairs of a service in context c, HMPSituation(c') is a situation that exists because of a HMP's state of affairs in context c', background condition is a required condition of a defined situation constraint and ReactiveSituation(c) is a consequential situation that exists that requires some action to be taken.

Similarly for HMP situations.

7. Problems And Issues Encountered

As can be expected in an "inherently iterative and incremental activity" such as design science research (Hevner et al., 2004), some problems and issues were encountered when developing the ontology. These are discussed below together with the adaptations of the original design that were made or are recommended to address the problems and issues.

7.1 Ontology Development

In the process of developing the travel situation management ontology some unexpected, but resolvable, issues and problems were encountered. They were generally related to the ability to represent things and relationships in different ways. Nothing was encountered that could NOT be represented in the ontology.

7.1.1 Specification of location

As previously discussed, a key dimension of context is spatial location. Latitude, longitude and altitude were chosen to specify absolute location as it is universally understood and accepted.

Specification of an area on the earth's surface can be achieved in many ways. Natural Area Coding (NAC) was chosen as it can be used to specify an area on the earth of virtually any size, and if necessary, a volume of any size anywhere in the universe (NAC, 2006). While it is a proprietary encoding scheme, it has been gaining wide acceptance and is flexible enough to enable specification of everything from a continent to an absolute location. As conversion between NAC format and latitude/longitude can be achieved with simple algorithms, NAC was chosen for encoding of areas in conjunction with latitude and longitude. This dimension of context was called *locationspan*.

7.1.2 Specification of Time

Another key dimension of context is time, or temporal location. Similar to spatial location, a temporal location can be absolute, that is a specific point in time, or it can span many points in time, that is, it can have a duration. As local time is relative to location, a universal absolute time is necessary to allow comparison of time in different locations. The most common absolute time for civil usage is Co-ordinated Universal Time (UTC) which was used in this ontology.

However, the local timezone must also be specified to allow appropriate reaction to situations. For example, opening and closing times of hotels and restaurants, arrival and departure times of flights and flight curfews are generally determined by local time, not by UTC. The Timezone entity in the generic ontology allows the number of hours ahead of UTC to be specified for the local timezone.

Duration, which is called *timespan* in this work, is specified by an absolute start time specified in UTC and an absolute end time, also specified in UTC. For ease of reading and automation an additional property, *granularity*, is included, which can take values such as second, minute, hour, half-day, day etcetera.

7.1.3 Specification of Contexts

As the reaction to situations can depend on the temporal and spatial subsumption relationship between the user context and the service context, the context class includes object properties for specifying which contexts it subsumes (*SpatiallySubsumes* and *TemporallySubsumes*) and contexts by which it is subsumed (*IsSpatiallySubsumedBy* and *IsTemporallySubsumedBy*).

For the situation type S where S = [s | |= <<isPrimeMinister, c', John Howard, True>>], an example of a situation that is a member of S would be one in which the context was 31 March 1996 and Australia because the timespan "2 March 1996 to today" subsumes 31 March 1996. A context of 31 March 1996 and Brisbane would also create a situation that supports the infon because Australia subsumes Brisbane.

To specify the above, each context with a locationspan of Australia would have Brisbane included in its set of SpatiallySubsumes properties and each context with a locationspan of Brisbane would have Australia included in its set of IsSpatiallySubsumedBy properties. Similarly the timespan of John Howard's term of office as Prime Minister would temporally subsume all days, months and years within his term and this would be specified in the TemporallySubsumes property of each context that has a timespan of Howard's term in office.

7.1.4 Context Property of Infons and Situations

The difference between an infon and a situation is that an infon can exist without being situated. When it becomes situated it becomes a fact. For example, the infon <<isPrimeMinister, John Howard, True>> and <<isPrimeMinister, John Howard, False>> are both valid infons although they appear to be contradictory. This is because they are not situated. When we put either of these infons into a particular context, it becomes either a truth (or fact) or a falsehood depending on the context. If the context is the United Kingdom in 2006 then it is clearly false but it is true, or factual, in Australia in 2006. A situation is a state of affairs (infon) in a particular context, so the infon

<<isPrimeMinister, John Howard, True>> is supported in all contexts where country equals Australia, and the date is between 2 March 1996 and today. That is,

S = [s | |= <<isPrimeMinister, c', John Howard, True>>]

An infon in a specific context is a situation. For this reason the infon class of the situation ontology does not have a context property. The context property is specified in the situation class, together with the associated infons and background conditions. This simplifies the specification of the generic situation ontology and the action rules associated with it.

7.2 Specification of Situation and Context Transition Rules

Event collections cause a transition from one context to another but do they also cause a transition from one situation to another? Events can cause a change in context and they can directly cause a new situation to occur with or without a change in context. An event will create a new state of affairs, that is a new infon, in an existing context. The combination of this new infon in an existing context will result in a new situation if that state of affairs did not previously exist. Therefore, both context transition rules and situation transition rules should include any relevant event collections. Situation constraints are still required to specify the transitions.

7.2.1 Rule Specification

As production of a prototype was outside the scope of this work, detailed action and context subsumption rules were not developed. However, consideration was given to the rule specification methods that were available. The contemporary choices for web based situation management systems include RuleML, Semantic Web Rule Language (SWRL), rule language for OWL (OWL-S) and the use of situation constraints. It was clear from the development of the situation ontology that the use of situation constraints to specify complex rules was not an appropriate choice. As an SWRL add-on is now available for the Ontology Editor, Protégé (2006), and Protégé can export its data in a format that is compatible with agent and expert system development tools such as the Java Expert System Shell (JESS) (2006), it is recommended that SWRL be used to specify the rules for the prototype.

7.3 Scalability

Rules and actions are defined for classes or types of reactive situations, not individuals. The complexity of the rules versus the granularity of the situation classes and contexts is a trade-off. The choice of granularity depends on the power of the end-user devices, the context granularity of the service to which the user subscribes and the users desired level of automation. If the context granularity is kept large, the complexity of the situation detection process is minimised, however, the complexity of the action rules will increase as more logic will be required to determine the most appropriate actions. This is appropriate for today's mobile infrastructure where the system components would execute on large internet based servers and handheld devices are of limited capability. In the future, as handheld devices become more powerful, the context granularities could be reduced to provide more specific situation detection and management.

Also, many of the current service providers only provide information at quite coarse spatio-temporal granularity, making it pointless to have a situation detection and management system that uses a finer granularity. For example, the Australian Department of Foreign Affairs and Trade (DFAT) issues travel warnings by country, not by city, state or region. However, transport situations need to be at the granularity of city and day to be able to manage flight cancellation and delays, train trips etcetera.

8. Summary

This work has introduced the concept of a reactive situation management system that utilises contexts to eliminate irrelevant situations from consideration for action. The aim is to improve system efficiency and accuracy. It has brought together concepts from ontology theory, context theory and situation theory in the design of an extensible travel situation management ontology. The system can be adapted to a new domain by replacing the travel related ontology classes with classes related to the new domain.

9. Contribution

This work makes several useful contributions to the field of mobile travel service delivery, particularly in relation to automated, reactive services. It also provides a sound basis for future research work in ubiquitous travel service delivery and for the development of prototype situation aware travel systems. Hevner et al (Hevner et al., 2004) state that “Effective design science research must provide clear contributions in the areas of the design artefact, design construction knowledge (i.e. foundations), and/or design evaluation knowledge (i.e. methodologies)…. One or more of these contributions must be found in a given research project“. This research contributes a novel design artefact, namely a travel situation management ontology, thus satisfying the first type of contribution. It also contributes to the foundation knowledge base of mobile service delivery systems by providing a generic situation management ontology, based on situation theory, that can be used in future research and systems design. This demonstrates the applicability, and feasibility of using situation theory in the design of reactive information systems.

Furthermore, the support within the ontology for context based filtering for situation detection contributes to the efficiency of implementation and operation of situation driven reactive travel information systems.

Hevner et al (2004) also state that any artefacts produced by design science research “must be implementable”. Representations of contexts, situations and rules must be available in an agreed digital form before an automated reactive system can be developed. Recently XML has become the standard for encoding data that will need to be shared across heterogeneous systems.

However, XML by itself does not carry any semantic information related to the data that is included in the document. The semantic information comes from the agreed meaning of the tags surrounding the data. As the recent research in contexts, situations, ontologies and rule specification has been carried out in a number of different projects, in a number of institutions by researchers with quite different viewpoints on the subject areas, there is neither a single ontology nor a single markup language that covers all these areas. Bearing this in mind, the travel situation management ontology developed in this work, which covers contexts, situations, situation action rules and travel domain concepts were encoded in the de facto standard ontology and rule markup languages, OWL and SWRL. This ensures that they are relevant to current and future work being undertaken in web-based travel information systems. Ontology development and management tools such as Protégé (2006) that have been used in this research have built-in support for OWL and SWRL, simplifying the development of OWL based ontologies. Furthermore, the ontologies developed in these tools can be exported in a form suitable for use in agent development systems such as JESS. Thus, the requirement of implementability is satisfied.

Bio

Paul has a PhD in Information Systems. He teaches online business and information systems. His research focuses on mobile information systems in travel and tourism. He has over 20 years experience in technical, consulting and executive management in information technology. He is a Vice-President of the Australian Computer Society and a member of a number of national Boards. He is a member of the School's Teaching and Learning Committee.

10. References

Adi, A., Botzer, D., & Etzion, O. (2002). *The Situation Manager Component of Amit - Active Middleware Technology*. Paper presented at the 5th international Workshop on Next Generation Information Technologies and Systems.

Akman, V., & Surav, A. (1996). Steps Towards Formalizing Context. *Artificial Intelligence Magazine, 17*(3), 55-72.

Angehrn, A. (1997). Designing Mature Internet Business Strategies: The ICDT Model. *European Management Journal, Vol, 15*(4), 361-369.

Barwise, J. (1981). Scenes and other situations. *Journal of Philosophy, 78*(7), 369-397.

Barwise, J. (1987). The Situation in Logic. *Centre for the Study of Language and Information, CSLI Lecture Notes No, 17*, 59-77.

Barwise, J., & Perry, J. (1983). Situations and Attitudes. *MIT Press, Cambridge*.

Cherry, G. (2001). Extending and Formalizing The Object-Oriented Paradigm with Situation-Driven Modeling. *Thought**Tools Corp, Cambridge, Massachusetts, Available at: www.sdml.com*.

Devlin, K. (1991). Logic and Information. *Cambridge University Press, Cambridge*.

Fischer, K., Hoyer, J., Laube, M., Müller, J., Schirmer, J., Steinmann, R., et al. (2002). *Multimedia Workplace of the Future MAP – Multimedia Arbeitsplatz der Zukunft*, 2, from informatiksysteme.pt it.de/mti/cd-rom/projects/map/beitrag_MAP.pdf

Fukazawa, Y., Luther, M., Wagner, M., Tomioka, A., Naganuma, T., Fujii, K., et al. (2006). *Situation-Aware Task-Based Service Recommendation*. Paper presented at the Fourth International Conference on Mobile Systems, Applications and Services, MobiSys.

Guarino, N. (1998). Formal Ontology and Information Systems. In N. Guarino (Ed.), *Formal Ontology in Information Systems*. Amsterdam: IOS-Press.

Hendler, J. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems, Vol. 16, No. 2, pp.*

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, Vol, 28*(1), 75-105.

Jakobson, G., Lewis, L., Matheus, C., Kokar, M., & Buford, J. (2005). *Overview of Situation Management at SIMA 2005.* Paper presented at the SIMA Conference, Atlantic City.

Jess. (2006). *Jess, The Rule Engine for the Java Platform, The Java Expert System Shell*, from http://www.jessrules.com

KEE. (1993). *KEE tm (Knowledge Engineering Environment) Software Development System, Version 4.1*

Lenat, D. (1998). The Dimensions of Context-Space. *Cycorp, Austin, Available at: www.cyc.com.*

Matheus, C. (2005). Using Ontology-based Rules for Situation Awareness and Information Fusion, *W3C Workshop on Rule Languages for Interoperability*.

Matheus, C., Kokar, M., & Baclawski, K. (2003). A Core Ontology for Situation Awareness, *Sixth International Conference on Information Fusion* (pp. 545-552). Cairns.

MIT. (2004). MIT Oxygen Project.

NAC. (2006). *NAC Natural Area Coding*, from http://www.nacgeo.com/

O’Brien, P. (2006). *An Architecture for Ubiquitous Mobile Service Delivery.* University of Queensland, Brisbane.

Perry, J. (1993). The Problem of the Essential Indexical and Other Essays. *Oxford University Press, New York.*

Protége. (2006). *The Protégé Ontology Editor and Knowledge-Base Framework*, from http://protege.stanford.edu

RuleML. (2003). from http://www.dfki.unikl.de/ruleml

SWRL. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *Available at: http://www.w3.org/Submission, 20040521*, SWRL-20040521.

Tin, E. A. V., & Ersan, M. (1995). Towards Situation-Oriented Programming Languages. *ACM SIGPLAN Notices, Vol, 30*(1), 27-36.

WiMAX. (2006). *The Wimax Forum*, from http://www.wimaxforum.org

TABLES

$A \rightarrow e$

$e \rightarrow \delta s$

$\delta s \Rightarrow \delta C$ OR $\delta s \Rightarrow \delta S$

$\delta C \Rightarrow \delta S$

$\delta C \Rightarrow S_R \mid B$

$\delta S \Rightarrow S_R \mid B$

$S_R \rightarrow A1, A2 \ldots . An$

where e = event, δC = change in context, δs = change in state, S = situation, S_R = reactive situation, δS = change in situation, B = background situation (a situation as described in Section 4.1 above), A = action, $\rightarrow$ = "causes", $\Rightarrow$ = "*may* cause" and $\mid$ = "subject to".

Table 1 Relationship between events, context, situations and actions

Class Name	Class Description
ABSOLUTE_LOCATION	Geocode for absolute geographical location in latitude, longitude and altitude
ABSOLUTE_TIME	Absolute time in UTC
ACTION_RULE	URI of SWRL action rules
ADDRESS	Street address of place
AGENT	URI of HMPA or SMA software agent
COMPOUND_INFON	An infon that is a logical combination of other infons
CONTEXT	A full context definition of a unique context
CONTEXT_TO_SITUATION TRANSITION_RULE	A rule defining the conditions under which a situation can be created from within a context
CONTEXT_TRANSITION_RULE	A rule defining what events can cause a transition from one context to another
EVENT	Defines an event that causes a change of state of an entity
EVENT_COLLECTION	A collection of events that is used in context transition rules

HMP	A highly mobile person
HMP_EVENT	Defines an event that causes a change of state of a HMP
HMP_EVENT_CHANGED CONTEXT	Defines an event where a HMP’s current context changes
HMP_EVENT_PREF_CHANGED	Defines an event where a HMP’s service preferences change
INFON	A unit of information or state of affairs as defined by Barwise [5]
INFON_ARRIVED	An infon that asserts that something such as a HMP has arrived in a location
INFON_FUTURE_LOCATION	An infon that asserts that something such as a HMP will be in a location in the future
INFON_INLOCATION	An infon that asserts that something such as a HMP is currently in a location
INFON_INTRANSIT	An infon that asserts that something such as a HMP is travelling between two locations
INFON_PAST_LOCATION	An infon that asserts that something such as a HMP was in a location in the past
LOCATION_SPAN	A geographical area
PERSON	A human being who is currently living, will be alive in the future or was alive in the past
PLACE	A type of physical location
PLACE_BLOC	A group of countries that form a political, economic or social group
PLACE_BUILDING	A physical structure
PLACE_CITY_BLOCK	A city block typically bounded by a number of streets
PLACE_CONTINENT	One of the seven continents
PLACE_COUNTRY	A sovereign state
PLACE_LGA	A local government area. Typically a city or shire.
PLACE_LOCALITY	A small part of a city
PLACE_REGION	A part of a state that generally contains a number of LGAs
PLACE_STATE	A province of a country
PLACE_STREET	A named road within a city
PLACE_SUBURB	A named area of a city that generally contains a number of localities
SERVICE	An information, communication, distribution or transaction service provided by a service provider
SERVICE_EVENT	Defines an event that causes a change of state of a service provided by a service provider

SERVICE_MANAGER	A software agent or program that manages one or more service providers of a particular type
SERVICE_PREFERENCE	Preferences that a particular HMP has chosen in relation to the actions that he/she requires to be taken in reactive service situations for a particular type of service.
SERVICE_PROVIDER	A physical provider of particular service such as an airline
SITUATION	A state of affairs, or an infon, that is true in a particular context
SITUATION_HMP	A state of affairs, or an infon, of a HMP that is true in a particular context
SITUATION_HMP_FUTURE LOCATION	A state of affairs, or an infon, where it is true that a particular HMP will be in a location at a particular time in the future.
SITUATION_HMP_INTRANSIT	A state of affairs, or an infon, where it is true that a particular HMP is in transit between two locations
SITUATION_HMP_IN_LOCATION	A state of affairs, or an infon, where it is true that a particular HMP is in a location at a particular time
SITUATION_HMP_PAST LOCATION	A state of affairs, or an infon, where it is true that a particular HMP was in a location at a particular time in the past.
SITUATION_REACTIVE	A situation that requires some action to be taken
SITUATION_SERVICE	A state of affairs, or an infon, of a service that is true in a particular context
SITUATION_TRANSITION CONDITION	A rule defining what background conditions can cause a transition from the current situation to a new situation
TIMEZONE	A standard international timezone
TIME_SPAN	A period of time
HMP_EVENT_FARE_PAID	Event where HMP pays fare for a reservation
HMP_EVENT_RES_MADE	Event where HMP pays makes a reservation
INFON_HAS_RESERVATION	Infon describing state of affairs where a HMP has a reservation
PLACE_BUILDING_AIRPORT	A building categorised as an airport
PLACE_BUILDING_HOTEL	A building categorised as an hotel
RESERVATION	A reservation for accommodation or transport services
RESERVATION ACCOMMODATION	A reservation for accommodation
RESERVATION TRANSPORTATION	A reservation for transport services
RESERVATION TRANSPORTATION_AIR	A reservation for air transport services

RESERVATION TRANSPORTATION_Bus	A reservation for bus transport services
RESERVATION TRANSPORTATION_TRAIN	A reservation for train transport services
SERVICE_EVENT_ARRIVAL DELAY	Event where arrival of something has been delayed to time later than the scheduled time
SERVICE_EVENT_DEPARTURE DELAY	Event where departure of something has been delayed to time later than the scheduled time
SERVICE_EVENT_HEALTH WARNING	Event where a health warning has been issued by a health information service provider
SERVICE_EVENT_SAFETY WARNING	Event where a safety warning has been issued by a travel safety information service provider
SERVICE_EVENT_TRIP CANCELLED	Event where a scheduled trip has cancelled
SITUATION_HMP_RESERVATION REQUIRED	State of affairs in a particular context that requires action to be taken to make a reservation for a HMP
SITUATION_SERVICE ACCOMMODATION	A state of affairs of an accommodation service provider in a particular context
SITUATION_SERVICE ACCOMMODATION_AVAILABLE	A state of affairs of an accommodation service provider where accommodation is available in a particular context
SITUATION_SERVICE ACCOMMODATION_PRICE CHANGE	A state of affairs of an accommodation service provider where the price of accommodation has changed in a particular context

Table 2 Travel Situation Management Ontology Classes

FIGURES

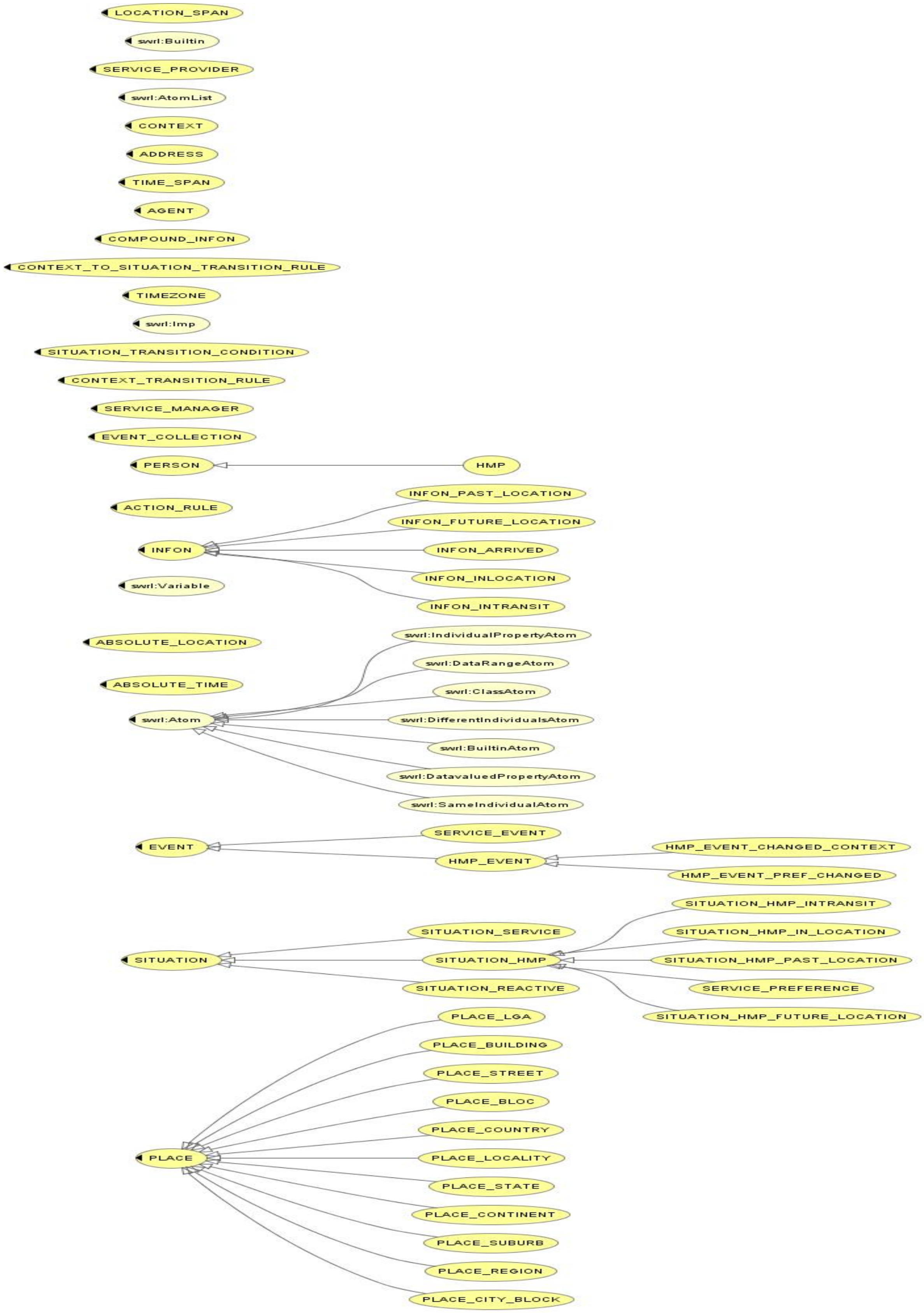

Figure 1 Travel Situation Management Ontology

Bio

Paul has a PhD in Computer Science & Information Systems. He is the MBA Director at the Griffith University Business School where he teaches information systems management and governance. His research focuses on mobile information systems and tourism technology. He has over 20 years experience in technical, consulting and executive management in information technology. He is a Board member of the Australian Computer Society and a member of the ACS National Professional Standards Board and Disciplinary Committee. He is also a member of Griffith’s Teaching and Learning Committee.