A Hybrid Approach to Context Modelling in Large-Scale Pervasive Computing Environments

Deirdre Lee
Trinity College Dublin, Ireland
Distributed Systems Group
Department of Computer Science
Deirdre.Lee@deri.org

René Meier
Trinity College Dublin, Ireland
Distributed Systems Group
Department of Computer Science
Rene.Meier@cs.tcd.ie

ABSTRACT

Pervasive computing aims to unobtrusively embed computer systems into everyday life environments to enrich the user experience without demanding the user's explicit attention. For pervasive computing to be minimally invasive, computer systems must be conscious of and ultimately, be able to act according to the context of the user and her intelligent environment. However, such context is often heterogeneous as it is derived from myriads of independent systems and sensors, can be incomplete, and may even be erroneous. For independent systems to manage, share, correlate, and reason over context, contextual information must be modelled in a homogenous fashion. This paper proposes a hybrid approach to modelling contextual information that incorporates the management and communication benefits of traditional object-oriented context models, while also taking advantage of the semantic and inference benefits of ontology-based context models. Unlike other approaches, our hybrid model has been designed to support a specific large-scale pervasive domain, namely the transportation domain, and promotes exploiting primary context as the key to accessing and correlating distributed knowledge. The Primary-Context Model and the Primary-Context Ontology are part of a pervasive middleware architecture for integrating independent Intelligent Transport Systems and pervasive transportation services and has been applied to a prototypical realisation of such system and service integration.

Categories and Subject Descriptors

D.1.5 [**Object-oriented Programming**]; D.2.2 [**Design Tools and Techniques**]: *Object-oriented design methods*; D.2.10 [**Design**]: *Methodologies*; D.2.11 [**Software Architectures**]: *Data abstraction, Domain-specific architectures*; D.2.12 [**Interoperability**]: *Distributed objects*; E.1 [**Data Structures**]: *Distributed data structures*; H.3.3 [**Information Search and Retrieval**]: *Retrieval models*; H.3.4 [**Systems and Software**]: *Distributed systems*

General Terms

Design, Standardization.

Keywords

Context modelling, pervasive computing, hybrid context model, ontology, transportation.

1. INTRODUCTION

Since the dawn of computing, advances have been made to increase the automation of technology, while minimising human-computer interaction. Pervasive computing is an area of computing in which computer functions are integrated into everyday life, often in invisible ways. In order to realise the vision of true pervasiveness [1], computer systems must be able to seamlessly interact with each other and share context. However, achieving such interoperation has proven difficult, especially, across a multitude of legacy and emerging systems. Considering that a large-scale computing environment, for example, the transportation infrastructure of a city, often contains numerous independent systems, ubiquitous interaction between these systems is fundamental to enabling efficient and effective provision of pervasive services to the users in such an environment. A common approach to pervasive computing is for systems to model their contextual information in a homogenous fashion, where context is any information that can be used to characterise the situation of an entity [2]. If context is modelled homogenously, it can be accessed consistently, shared between systems, and correlated with context from other systems. There are several approaches to modelling context, which differ in the way the context is structured. The most popular of these approaches are object-oriented models and ontology-based models. Object-oriented models offer the benefits of inheritance, modularity, polymorphism, and encapsulation, but are lacking in formal semantics. Ontology-based models provide a formal specification of all entities in a domain and the relationships between those entities, ensuring a shared understanding of the context. Conversely, the complexity of ontology-based models detracts from their manageability.

This paper presents a hybrid approach to modelling context: the Primary-Context Model and the Primary-Context Ontology (PCM and PCOnt). The PCM and PCOnt combine the management and communication benefits of traditional object-oriented modelling techniques with the semantic and inference benefits of ontology-based models. This approach has previously been applied to the global domain resulting in all-encompassing, generic models that lack support for domain specific context. Focusing on a particular domain, namely the transportation domain, enables us to tailor our hybrid-model to explicitly support real-world transportation situations, while still facilitating context modelling for the wide range of systems and services associated with this domain. Moreover, the PCM and PCOnt also promotes a novel approach for correlating diverse sets of independent and possibly distributed data based on an integral role of primary context. Context data varies in its level of importance with regards to the interaction between systems or between systems and users. Primary context, however, is the most significant context data, as it is used as the basis for querying other context, and it can therefore be used to correlate context from different systems. Primary Context (PC) is identified as identity PC, location PC, time PC, and quality of service PC. The PCM provides a standard way for systems to store, manage, and share context in a scalable manner, based on primary context. The PCOnt formally specifies the concepts that may be referred to by the PCM as well as the relationships that hold between these concepts. Such a thorough representation of all knowledge in a domain can be used to correlate data from different sources and, ultimately, to reason about the context and to infer new context.

The transportation domain serves as an archetype of a large-scale pervasive computing environment, containing context-rich systems that have been deployed on a metropolitan scale. The context derived from such systems can serve as a basis for the provision of pervasive transportation information and services to users travelling by car, on foot, or using public transport. The PCM and PCOnt have been specifically designed for the transportation domain and are part of the iTransIT framework [3, 4], a middleware architecture for integrating the very many, distributed Intelligent Transport Systems (ITS) in an urban environment. A Smart Traveller Information Service (STIS) [5] for identifying travel routes across multiple modes of transport has been extended in order to evaluate the benefits of our hybrid approach to context modelling. The STIS incorporates context from various distributed ITS, all of which are modelled according to the PCM, in terms of the concepts from the PCOnt and can be accessed using our spatial programming interface, demonstrating the management, querying, and correlation benefits of the PCM. The most appropriate route, based on current weather conditions, traffic-congestion conditions, and user requirements, is inferred by the PCOnt and presented to the user, demonstrating the semantic and reasoning functionality of the PCOnt.

This paper has been structured as follows: section 2 discusses work that is related to our hybrid context model. Section 3 presents the proposed Primary-Context Model and the Primary-Context Ontology. Section 4 introduces the iTransIT framework for which our hybrid context model has been designed. Section 5 evaluates our approach demonstrating its applicability to pervasive transportation services. Finally, section 6 concludes this paper and outlines research that remains for future work.

2. RELATED WORK

Pervasive middleware architectures facilitate the use, reuse, and correlation of context by separating the context collected by distributed and independent sensors or systems, from how that context is accessed and used by pervasive applications or services. In order to enable context integration, pervasive middleware architectures adopt a specific context-modelling technique. There are several approaches to modelling context. These approaches differ in the way that context is structured, i.e. how contextual information is represented and exchanged. Strang and Linnhoff-Popien [6] survey the most relevant current context modelling approaches identifying key-value models, mark-up scheme models, graphical models, object-oriented models, logic-based models, and ontology-based models. They conclude that ontology-based models show the most promise for modelling context in pervasive computing domains with respect to handling context in a distributed fashion, validating context, providing quality of context indicators, supporting incompleteness and ambiguity of context, providing a formal definition of the domain, and applying context to existing domains.

While middleware for pervasive environments can be based on broker architectures, service-oriented architectures, or agent architectures, the main distinction between such middleware is the approach to modelling context in order to support context reuse and sharing across many pervasive services. Whereas context-aware projects such as Aura [7], Context Weaver [8], Nexus [9], and iTransIT take a traditional object-oriented approach, Gaia [10], Context Management Framework [11], ACAI [12], CoBrA [13], SOCAM [14], DAIDALOS [15], and Kamer [16] introduce ontologies into their architectures. Object-oriented modelling techniques provide support for storing, managing, sharing, and accessing context in a scalable manner. But they lack semantics and the ability to reason about context. Ontologies, on the other hand, enhance context with semantics for knowledge representation, autonomic processing, and reasoning. While ontologies allow for enhanced semantic and reasoning functionalities, research into efficient ontology storage, access, and management is still at an early stage. Becker and Nicklas [17] highlight that "the scalable management of context information is not a core feature of ontology-based context management". Object-oriented and data modelling techniques are still more advanced at these tasks. Therefore, an ontology-based approach alone is not sufficient to model context effectively in a pervasive environment. A hybrid approach, as suggested by Becker and Nicklas and implemented by DAIDALOS and Kamer, can incorporate the benefits of both approaches. DAIDALOS uses an object-oriented context model for sharing context information among the interested parties, while its COntext MAnagement oNTOlogy, called COMANTO, is employed to describe the properties and structure of context model objects. Kamer features a context model, called Context Distribution and Reasoning, consisting of two sub-models, an object-based Context Acquisition and Delivery Data Model (CADDM), which defines the format for data exchange among distributed components (including sensors and applications), and an ontology-based Context Reasoning Data Model, which is a semantically rich model, enabling reasoning on context data.

The hybrid context model presented in this paper differs from both the DAIDALOS context model and the Kamer context model with respect to the complexity of the object-oriented context model, the separation of the ontology-based model into upper and lower ontologies, and the emphasis on primary context. DAIDALOS proposes a basic object-oriented context model, consisting of only entities, attributes and associations. Kamer's object-based CADDM is considerably detailed, specifying many kinds of entities that may exist in a pervasive environment and their associated context. In contrast, our PCM presents a refined number of generic entities and primary context, resulting in a flexible, yet effective model. Furthermore, the PCM and PCOnt is distinct from DAIDALOS and Kamer in its use of primary context. Primary context is used in our context model to index all auxiliary context, meaning that it is used to query secondary context and other data pertaining to a domain entity. Entities may also be correlated based on primary context, for example to determine all entities (irrespectively of their kind) at a particular location. DAIDALOS and Kamer aim to be all embracing which has led to complex context models supporting a multitude of concepts. Our model centers on four primary context types to specifically provide for context sharing across the independent systems in large-scale pervasive environments without the need for global consensus.

3. A HYBRID CONTEXT MODEL

Pervasive computing domains typically comprise myriads of sensors, actuators, and independent context-aware systems. Pervasive services that support users in such domains may be based on context derived from several of these components. Context models aim to overcome incompatibility issues when using context from independent components by facilitating access to distributed context and by enabling sharing and correlation of context between systems and services. Advanced models also facilitate reasoning over context, by allowing for the deduction of further information and appropriate actions based on existing context.

The PCM and PCOnt ensures interoperability between data sets captured across individual distributed systems through its hybrid nature and by proposing a small set of predefined abstractions for representing context. Whereas the PCM incorporates traditional object-oriented modelling techniques, the PCOnt offers an ontology-based approach. Using both the PCM and PCOnt in conjunction with one another to model context enables access and efficient sharing and correlation of context, as well as allowing for semantic interpretation and inference of context. These abstractions are used to model data sets and their context according to the different roles data sets can assume in a pervasive environment. They are central to providing services with a common view on the wide range of information and the associated context that might be available. The abstractions hide the complexity and diversity of the independent systems and data sources, and represent the hooks for information integration through overlapping context. Developing such abstractions for any programming model is non-trivial; however, it is especially complex for large-scale, pervasive environments due to the scale and multitude of inter-relationships that exist between sensors, systems, services, users, and their data sets. Lehman et al. [9] suggest an exhaustive ontology for defining how context information can be shared between applications in augmented areas. However, based on our experience with a real pervasive environment in the transportation domain, we have found that a relatively small number of abstractions suffice to decompose a pervasive space domain model. Using a small set of (coarse-grain) abstractions rather than attempting to model the entire world in detail simplifies management and maintenance in light of continuously evolving environments. For example sharing context between a bus system and a train system does not require every aspect of each system to be modelled in detail: modelling general timetable and route context may suffice. Novel systems or services are expected to be modelled using combinations of existing abstractions whereas an exhaustive model might have to be expanded to capture the specific characteristics of novel systems.

3.1 Primary-Context Model (PCM)

The object-oriented PCM defines our abstractions for modelling information and associated primary context as spatial objects. Spatial objects have been designed as a series of abstract objects to enable management, communication, and access to information, and most importantly to support information correlation through primary context. As shown in Figure 1, there are three kinds of spatial objects for modelling global information: system object, real world object and data object. Spatial objects are associated with the corresponding primary context objects: identity, time, quality and location. Spatial objects represent information and context as a series of parameters structured as name-type-value triples. Name and type of context parameters are predefined in the PCOnt to enable semantic correlation based on context.

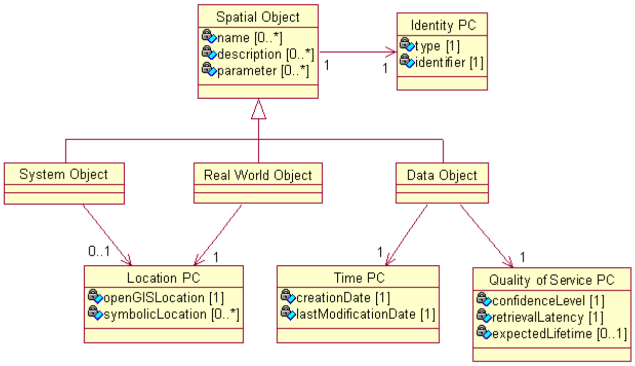


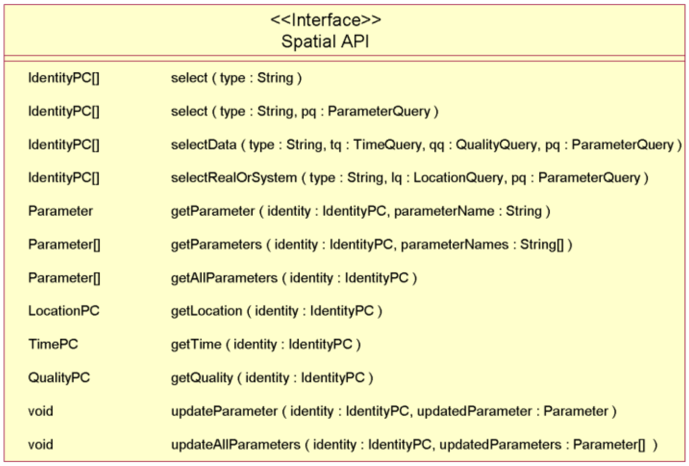
Figure 1. Primary-Context Model and spatial objects

The three kinds of spatial object reflect the different roles that objects can assume in a pervasive environment. System objects represent general information describing the operational status of software components, including systems and services. Real world objects represent physical entities. For example, in a transportation environment, system objects might capture the operational status of a car parking system or a journey time estimation service whereas real world objects might model vehicles, roads or traffic signals. Data objects represent non-physical entities and model any static or dynamic information regarding, or generated by, real world object or systems objects, such as public transport timetables or the number of available car parking spaces.

Primary context represents the essential context for indexing and correlating data. Identity PC uniquely describes spatial objects and consists of the spatial object type and an identifier. The type of an identity PC refers to a concept defined in the PCOnt. The identifier is expressed as a Uniform Resource Identifier (URI), meaning that it can uniquely identify any spatial object across the pervasive computing environment. Time PC models the temporal context of data objects enabling pervasive services to access and correlate information based on time relevance. Quality PC defines a set of quality requirements associated with a data object. Although quality is not typically considered as primary context, we argue that quality is essential for accessing and correlating information. Quality PC allows pervasive services to identify information that may provide certain service guarantees and enables management of deficient or erroneous context. Location PC is arguably the most important primary context of the PCM and uses geometric shapes to model the relevant special context of sensors, systems, services and even users. Location PC supports a topographical approach [18] to modelling space implying that shapes explicitly represent spatial context derived from the real world. Shapes may reflect the physical appearances of spatial objects modelling occupied space or may describe areas of interest that specify the regions covered by services. For example, a city-wide car parking system might use the spatial model to define the physical locations occupied by its car parks whereas a road weather service might use the spatial model to outline the locations occupied by weather stations as well as the areas to which reports from individual stations apply. Using a topographical approach to modelling space enables systems and services to independently define and use potentially overlapping spatial context in a consistent manner. Unlike topological approaches [18], in which geographical relationships between spatial objects are described explicitly, topographical models define relationships between spatial objects implicitly and without explicit interactions between objects. The relations between spatial objects (and ultimately systems, services and users) are defined by the position of their respective shape within a common coordinate system. This is particularly significant in large-scale pervasive environments where multitudes of independent systems are distributed over large geographical areas and direct communication across systems may be limited or expensive. Services using the PCM can exploit these implicit relations to link diverse information together for a user specific purpose. They may access spatially related information for example, by means of exploiting the distance between shapes or by exploiting containment and intersection relations. This might, for example, enable a vehicle-based information system to retrieve the exact locations of car parking facilities within a certain distance from its current location.

3.1.1 Accessing PCM Spatial Objects

The PCM is realised through an interface, called the Spatial Application Programming Interface (Spatial API). The Spatial API is an object-oriented interface, which provides (conceptually) common access to domain context through primary context queries. Individual systems use spatial objects to model their respective contextual information in accordance with the PCM and implement the Spatial API, to provide access to these objects. As outlined in Figure 1, each spatial object is associated with Identity PC that contains a spatial object type, which corresponds to a concept defined in the PCOnt. This augments spatial objects with semantics that can be used for context correlation, consistency checking and for inference purposes.



```
                         <<Interface>>
                          Spatial API

IdentityPC[]     select ( type : String )
IdentityPC[]     select ( type : String, pq : ParameterQuery )
IdentityPC[]     selectData ( type : String, tq : TimeQuery, qq : QualityQuery, pq : ParameterQuery )
IdentityPC[]     selectRealOrSystem ( type : String, lq : LocationQuery, pq : ParameterQuery )
Parameter        getParameter ( identity : IdentityPC, parameterName : String )
Parameter[]      getParameters ( identity : IdentityPC, parameterNames : String[] )
Parameter[]      getAllParameters ( identity : IdentityPC )
LocationPC       getLocation ( identity : IdentityPC )
TimePC           getTime ( identity : IdentityPC )
QualityPC        getQuality ( identity : IdentityPC )
void             updateParameter ( identity : IdentityPC, updatedParameter : Parameter )
void             updateAllParameters ( identity : IdentityPC, updatedParameters : Parameter[]  )
```

Figure 2. The spatial application programming interface

As illustrated in Figure 2, the operations of the Spatial API provide a means for systems and services to locate, access and manage spatial objects. A set of `select()` operations is available for locating spatial objects using primary context queries (including temporal, quality of service, and geometric queries) and queries based on the typed parameters of objects. For example, according to the PCM, real world objects are associated with Location PC, therefore real world spatial objects can be retrieved using Location PC, spatial object type and/or parameter values. For example, if both 'pcont:bus_stop' and 'pcont:route' are defined as real world objects, identifying whether a 'pcont:bus_stop' is on a particular 'pcont:route' can be determined by retrieving the location context of both the bus stop and the route, and correlating the results. Spatial objects representing data can be retrieved using Time PC, Quality PC spatial object type and/or parameter values. For example, the data associated with a weather station might be selected using the value of a measurement in combination with the temporal occurrence and quality of the measurement. Such primary context queries may identify zero, one or more spatial objects. The Identity PC of the returned spatial objects may then be used to either retrieve more information about the object or to update the parameters of spatial objects.

Significantly, the PCM allows services to exploit primary context along a combination of the spatial, temporal, quality and type dimensions. For example, a road-user information system might use the location and time primary context of an accident to retrieve the prevailing weather conditions at the accident site and may subsequently advice drivers of dangerous road conditions. Furthermore, the PCM enables a federation of independent systems to model their respective information and context locally as spatial objects. Each of these systems implements the Spatial API to provide access to its respective set of spatial objects. This enables systems and services to use, share, locate and correlate these distributed objects using a common set of context operations irrespective of the complexities of the systems accommodating the objects and without the need for an overall close integration of the systems. This mapping of the PCM and its programming interface onto individual systems therefore provides for truly pervasive services in large-scale and heterogeneous intelligent environments.

3.2 Primary-Context Ontology (PCOnt)

Object-oriented approaches to context modelling facilitate context management, access, and correlation based on primary context, but lack support for seamless semantic integration across independent systems and services. Independent systems introduce inherent differences in their respective contextual information. They introduce differences in the terminology used to describe entities, in the properties associated with individual entities, and in the relationships that hold between entities. Ontologies facilitate such semantic integration by providing a formal and machine interpretable model of an environment. This essentially means that all systems and services have a shared understanding of the meaning of their information. For example, if a system provides contextual information (in the form of a spatial object) describing a bus, another system or pervasive service can recognise the semantic meaning of the term ‘bus’, as a public transport vehicle that drives along a route and follows a timetable, and which in turn is a real world spatial object, and therefore must have associated Location PC and Identity PC. Our hybrid approach to modelling context introduces, in addition to the PCM, the PCOnt. The PCOnt is an ontology-based context model, built using the OWL Web Ontology Language [19], that facilitates machine interpretability of content through the provision of additional vocabulary along with a formal semantics. The PCOnt is divided into the Upper PCOnt and Extended PCOnts. The Upper PCOnt formally defines the spatial objects of the PCM and the primary context they must be associated with. Extended PCOnts are sub-ontologies of the Upper PCOnt and provide a formal representation of specific sub-domains within a pervasive environment, for example, in a transportation environment, Extended PCOnts might describe the weather sub-domain or the public transport sub-domain.

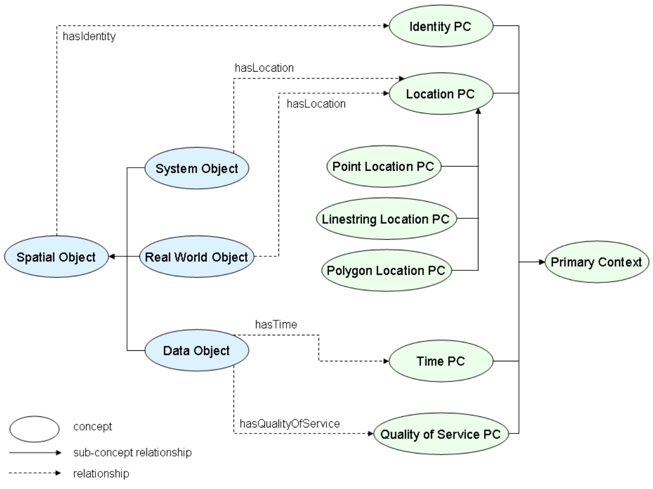


Figure 3. The Upper PCOnt

Recent research [20, 21] argues that it is unfeasible to reach consensus on a single global context-ontology that formally defines all possible contextual concepts and their relationships. Even in a contained environment, such as a single pervasive computing domain, it is difficult to create a single ontology for encompassing the semantics of every system. Moreover, maintaining such an ontology, especially, in terms of storage, distributed access and consistency in light of likely extensions, raises further non-trivial issues. We argue that the division of the PCOnt into the Upper PCOnt and the Extended PCOnts facilitates extensibility as it allows for the accommodation of new systems and service into a pervasive environment without a need for domain wide consensus. Also, Extended PCOnts enable distributed management of (sub-)ontologies and alleviate the burden of systems and services having to maintain a single, possibly very large ontology.

Table 1. Upper PCOnt relationships

Relationship	Domain	Range	Restriction
hasIdentity	spatial object	identity PC	exactly 1
hasLocation	system object	location PC	no restriction
hasLocation	real world object	location PC	exactly 1
hasTime	data object	time PC	exactly 1
hasQuality	data object	quality PC	exactly 1

As depicted in Figure 3, the Upper PCOnt is our top-level ontology and captures the basic context concepts of a designated large-scale pervasive computing domain. It uses the logic-based ontology language OWL, to semantically define the objects of the PCM and the relations between these objects. The structure of the Upper PCOnt fully supports the spatial object and primary context concepts. System object, real world object, and data object are defined as sub-concepts of the spatial object concept, whereas identity PC, location PC, time PC, and quality PC are defined as sub-concepts of the primary context concept. Location PC is subsequently divided into three sub-concepts to support basic geometric shapes including point, linestring and polygon. As summarised in Table 1, in addition to supporting these sub-concept relationships, the Upper PCOnt also defines (and imposes restrictions on) the relationships between spatial object concepts and primary context concepts.

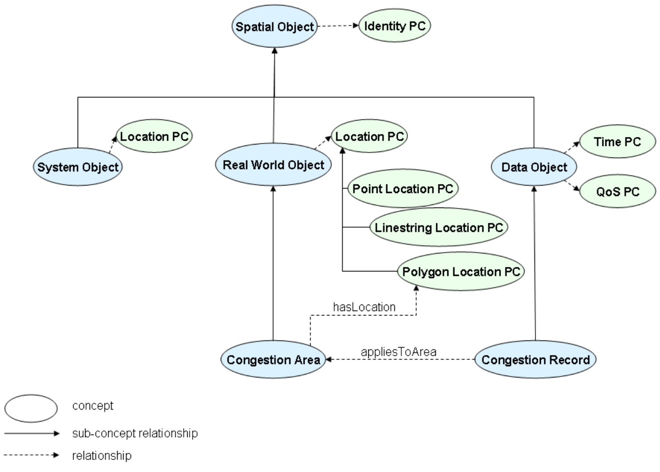


Figure 4. The Extended PCOnt of a traffic congestion system

Extended PCOnts are sub-ontologies of the Upper PCOnt and provide a formal, semantic representation of specific sub-domains. For example, our Extended PCOnts represent sub-domains of the transportation domain, including a traffic congestion sub-domain and a road network sub-domain. As previously outlined, this approach offers extensibility and alleviates the burden of having to conform to a single, possibly very large ontology. Moreover, it supports efficient context processing and reasoning as a small number of Extended PCOnts is required by a system at any one time. Similarly, for pervasive services to benefit from ontological features, they only require the Extended PCOnts of the systems that provide the contextual information used by a particular service rather than of all systems in the domain. For example, a pervasive service advising drivers of possible routes might use context from a road network system and from a traffic congestion system. Such a service depends on the PCOnts of these two sub-domain systems only to infer the best route a driver might take, considering prevailing congestion levels. Figure 4 illustrates an Extended PCOnt of a traffic congestion system that serves as a canonical example of how to construct an Extended PCOnt based on the Upper PCOnt. Two new concepts are defined in this Extended PCOnt for the congestion area and for the congestion record. The congestion record represents the current traffic congestion data and the congestion area represents the geographical area to which the record applies. Congestion area is a sub-concept of the real world object concept, meaning that it must be associated with location PC. The hasLocation relationship to a location PC is overwritten with a hasLocation relationship to a polygon location PC. This still satisfies the Upper PCOnt as polygon location PC is a sub-concept of location PC. The congestion record is a sub-concept of the data object concept, meaning that it must be associated with both time PC and quality PC. The relationship appliesToArea is also defined between a congestion record and a congestion area, stating that a congestion record must be associated with a congestion area.

Such a formal specification of data enables the correlation and merging of context from multiple sources. The following use-case highlights the benefits of a shared understanding of context: In Newtown, traffic congestion on the west side of town is recorded using a road-sensor system, while traffic congestion on the east side of town is recorded using a modern camera system. However, both systems model their context in terms of the PCOnt, i.e. using the 'pcont-traffic:congestion' concept. Therefore, a web application that provides a graphical interface of all traffic congestion in the town can retrieve congestion context from both systems, using the Spatial API, and merge the context based on the PCOnt concepts used. Context from both systems are displayed homogenously to the web user, regardless that the information was sourced in different, heterogeneous congestion systems.

In summary, our PCOnt, composed of an Upper PCOnt and one or more Extended PCOnts, supports semantic compatibility between systems that wish to interact with one another. The PCOnt provides a shared understanding of the domain to all systems and services. All entities within that domain, the properties associated with the entities, and the relationships between the entities are semantically defined. The PCOnt is machine interpretable as it is expressed using an ontology language that is based on a formal logic and, as such, also facilitates reasoning and consistency checking. Reasoning enables inferring of high-level, implicit context from low-level, explicit context, thereby allowing deduction of further context from the context that is directly available from sensors or systems. Consistency checking can be used to verify the logical consistency of context, which may help to identify erroneous context and ultimately, to prevent incorrect interpretation of context.

4. ITRANSIT FRAMEWORK

The transportation domain is a canonical example of an area that can benefit greatly from pervasive middleware architectures. An abundance of context information is continuously collected, processed, and stored by ITS. However, the context produced by ITS is mainly system specific and therefore cannot be easily shared with or used by other systems. The iTransIT framework is a pervasive middleware architecture for integrating ITS in large scale, pervasive environments. Using a distributed, tiered approach, it facilitates the sharing and collating of data, while allowing the scalable integration of new and legacy ITS.

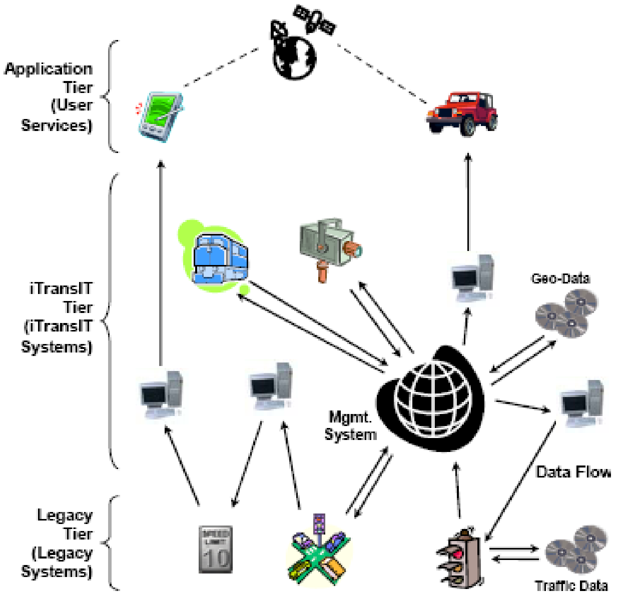

Application Tier (User Services)
iTransIT Tier (iTransIT Systems)
Legacy Tier (Legacy Systems)
Geo-Data
Mgmt. System
Data Flow
Traffic Data

Figure 5. iTransIT framework overview

As illustrated in Figure 5, the iTransIT framework structures legacy systems, ITS, and context-aware, end-user applications and services into three tiers. The legacy tier provides for the integration of existing systems that were not developed to conform to the iTransIT system architecture, but which still provide valuable information that can be used by pervasive transportation services. The iTransIT tier comprises a federation of ITS that have adopted the iTransIT system architecture shown in Figure 6, implying that they model their context according to the PCM and in terms of the PCOnt. In our framework, these ITS are known as iTransIT systems. The common data model is a key component of the architecture, as it represents all context available in the domain. It facilitates the integration of context from existing iTransIT systems that were designed specifically as part of the iTransIT framework, as well as those iTransIT systems that act as wrappers for legacy systems. iTransIT systems use spatial objects to model their context and implement the Spatial API, to provide access to these objects. Each system models the subset of the spatial objects that is relevant to its respective purpose and context-aware user services exploit the Spatial API to integrate and share information in a common way regardless of the specifics of the system implementing a particular part of the data model. As a result of the object-oriented nature of the PCM and of the extensibility of the PCOnt, the common data model is scalable and facilitates the incorporation of context from future and as yet unknown systems. Object-oriented queries are submitted to the interface based on the primary context associated with spatial objects thereby enabling queries such as ‘select all data objects created after 10th October 2008’.

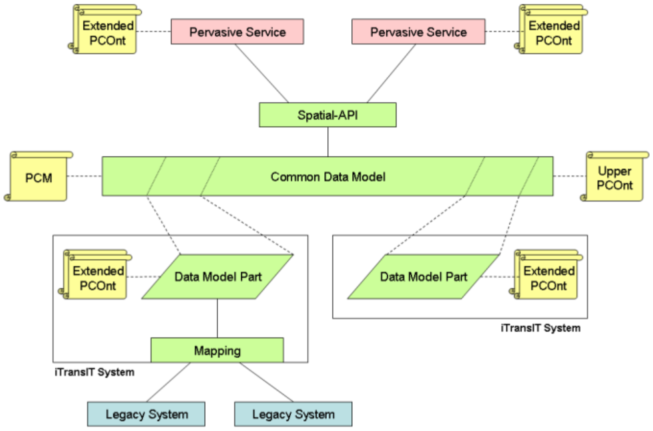


Figure 6. iTransIT framework architecture including PCM and PCOnt

As the Upper PCOnt is the ontological representation of the PCM, all context structured according to the PCM may also be formally expressed in terms of the Upper PCOnt, introducing semantic capabilities to the context. Figure 6 also illustrates that iTransIT systems may also model their context in terms of the concepts and relationships defined in one or more Extended PCOnts. An Extended PCOnt may be specific to a single iTransIT system, or it may cover several iTransIT systems. A pervasive service wishing to access context can do so by submitting primary-context queries to the Spatial API. If the query includes identity PC, the identity PC type may be a general concept, defined in the Upper PCOnt, or the system-specific concept, defined in an Extended PCOnt. Hence, using concepts from an Extended PCOnt allows for more expressive queries, such as ‘select all bus-timetable objects created after 10th October 2008’.

The application tier includes the pervasive services that provide context-aware user access to and interaction with traffic information. These pervasive services use the distributed data model and the associated context to access information potentially provided by multiple systems and might include a wide range of interactive (Internet-based) and embedded control services ranging from monitoring of live and historical traffic information to the display of road network maps.

5. EVALUATION

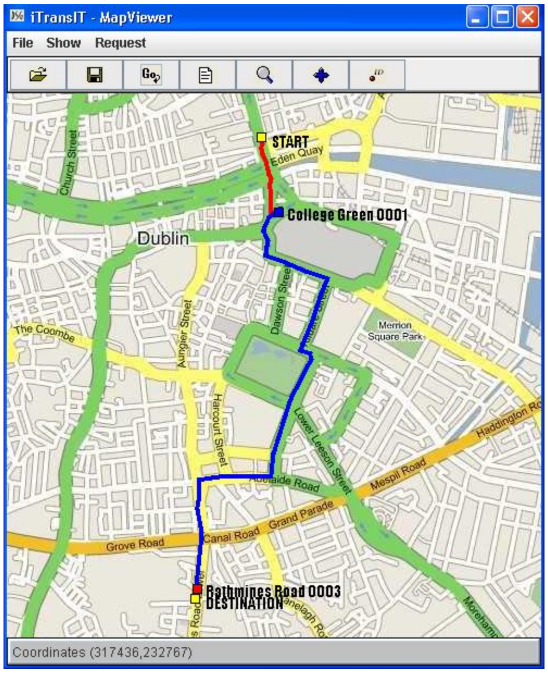
In order to evaluate the hybrid approach to context modelling in large-scale pervasive computing environments presented in this paper, we have applied the approach to an existing pervasive service, namely the Smart Traveller Information Service (STIS). The STIS has been developed using the Java programming language as part of a prototypical implementation of the iTransIT framework that captures real context derived from several of the transportation systems currently deployed in Dublin City in the Republic of Ireland [3, 4]. This demonstrates that our PCM and PCOnt can support context modelling in a real-world scenario and using real transportation context. The STIS has also been used to validate the benefits of our approach and to measure the costs of these benefits. We then conclude our assessment of the PCM and PCOnt by analysing the evaluation results with respect to the motivation criteria of this work demonstrating how these criteria have been met.

5.1 Hybrid Context Modelling for Pervasive Services

The STIS is an existing iTransIT pervasive service that has been implemented previously using only an object-oriented context model. The service provides users with a means of planning a journey through the Dublin metropolitan area, using multiple modes of transport. The STIS responds to real-time travel requests from urban and sub-urban commuters in Dublin with up-to-date, multimodal transport routes. Through a presentation layer, presented as a web-browser or a stand-alone application, the user specifies her desired journey’s start and destination locations, along with desired waypoints or travel-mode preferences. The STIS processes the route request, accesses the relevant information from a number of disparate, transport data sources, calculates the appropriate route, and returns the resultant route as an XML file, which may then be graphically displayed to the user. The travel modes that the STIS takes into account are walking, cycling, driving a vehicle, public bus, and tram. Figure 7 shows a multimodal travel-route, incorporating both walking and bus sections, displayed via the STIS MapViewer application.

The STIS (and the underlying iTransIT systems) has been re-implemented in order to assess the feasibility of applying the PCM and PCOnt hybrid context-modelling approach to a real-world application. By re-modelling the STIS according to the PCM and PCOnt the benefits, functionality, and overhead of a hybrid approach to context modelling, based on primary context, can be compared to the traditional object-oriented modelling techniques previously adopted by the iTransIT framework. Similar to the original service, the re-implementation of the STIS, hereafter known as STIS2, enables a user to find a route through Dublin by specifying her desired journey’s start and destination locations. However, while the STIS2 provides the same service to users as the original STIS implementation, it incorporates additional context sources. The STIS2 incorporates systems that provide highly dynamic context so that route selection may also account for and reason over the prevailing environmental conditions. This allows users to predefine their travel preferences depending on current conditions, instead of having to explicitly select the currently preferred mode of transport for each route request.

To support the STIS2, context generated by a variety of iTransIT systems is modelled according to the PCM spatial objects. This context is made available through the Spatial API, which offers access to context based on identity PC, location PC, time PC, and quality of service PC, as opposed to simple geometry-based queries, which were offered in the original interface. The incorporation of an ontology-based approach to context modelling has resulted in features that were not supported in the original implementation of this service. This functionality includes the universal interpretation of context, the correlation of context, and reasoning over context, which are discussed in more detail below.



Coordinates (317436,232767)

Figure 7. A recommended STIS route consisting of walking and public bus-based modes of transportation

As illustrated in Figure 8, five independent iTransIT systems provide the context for the STIS2 including the road-network system, the bus-network system, the tram-network system, the traffic-congestion system, and the weather system. The road-network system provides context on the road system in the Dublin metropolitan area, including streets, junctions, and one-way systems. The bus-network system provides context on bus routes and bus stops in Dublin. The tram-network system provides context concerning tram lines and tram stops in Dublin. The weather system provides context on the current weather conditions across Dublin. The traffic-congestion system provides context on the current congestion levels throughout Dublin, as recorded by Dublin's traffic management system. The weather and traffic-congestion systems are introduced in the STIS2 realisation, as they provide highly dynamic context describing environmental conditions. Weather context may change over the course of a day while traffic congestion also changes dramatically depending on the time of day, day of the week, traffic accidents, and road maintenance work. Both weather and traffic congestion are considered very influential when suggesting a possible travel route to a user, as they are likely to affect journey times and may influence a user's preferred mode of transport. Hence, the addition of and reasoning over the weather and traffic-congestion context is expected to improve user experience by providing a more realistic and beneficial route-finding service.

In summary, context is accessed via the Spatial API, based on the PCM and identifiers defined in the PCOnt. Correlation may be performed based on primary context: location, time, and quality of service. However, data encapsulated within this objects are defined in terms of Extended PCOnts, ensuring the reusability and universal interpretability of the context. This will be shown in the next section.

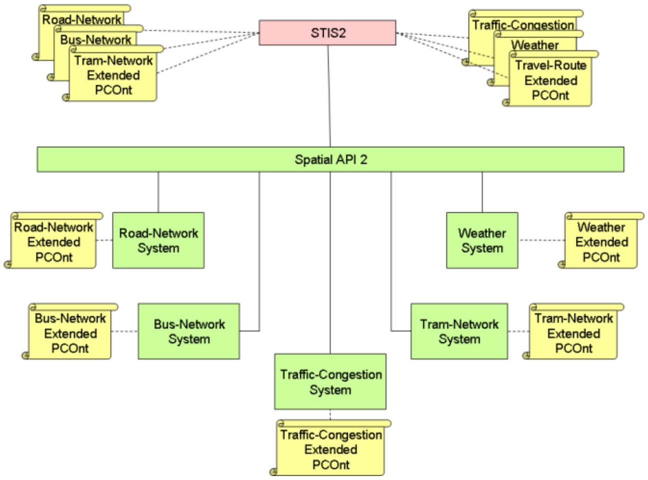

Road-Network
Bus-Network
Tram-Network Extended PCOnt
STIS2
Traffic-Congestion
Weather
Travel-Route Extended PCOnt
Spatial API 2
Road-Network Extended PCOnt
Road-Network System
Weather System
Weather Extended PCOnt
Bus-Network Extended PCOnt
Bus-Network System
Tram-Network System
Tram-Network Extended PCOnt
Traffic-Congestion System
Traffic-Congestion Extended PCOnt

Figure 8. The PCM and PCOnt for STIS2

5.1.1 *Universal Interpretation of Context*

In our hybrid context modelling approach, the PCOnt provides pervasive services and iTransIT systems with a common understanding of the environment. For the STIS2 to access context from a system through the Spatial API of that system, both the STIS2 and the system must have a shared understanding of the domain and the entities it contains. This is achieved through the use of Extended PCOnts. As is shown in Figure 8, the STIS2 uses the respective Extended PCOnts of the iTransIT systems containing relevant context. Context is modelled by the iTransIT systems according to PCM spatial objects. Associated with each spatial object is an identity PC containing a spatial object type that corresponds to a concept defined in the Upper PCOnt or in an Extended PCOnt. Thus, semantic information is encapsulated within each spatial object, i.e. associated with context. In addition, with a formal specification of all entities and their relationships in a domain, the primary context and the meaning of the parameters associated with each concept can be deduced. The STIS2 requires context from each of the iTransIT systems in order to return the optimal travel-route based on current conditions. Primary context queries are submitted to the Spatial API based on spatial object type, concepts from the relevant Extended PCOnt, and other primary context values. Based on the context retrieved from the road-network system, the bus system, and the tram system, a travel route for each of the five modes of transport can be determined. The walking, cycling, and driving routes are calculated using context from the road-network system. The bus route and tram route are calculated using context from the bus-network system and the tram-network system respectively.

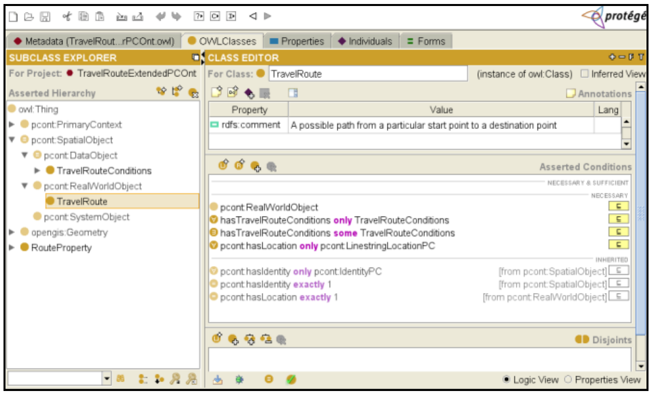
5.1.2 *Correlation of Context*

For the STIS2 to determine which of the travel routes are optimal with regards to a user's preferences, context from independent iTransIT systems needs to be correlated. When calculating a travel route based on a particular travel mode, straightforward primary context queries and/or parameters queries are submitted to the Spatial API. However, when the current weather and traffic-conditions of each route are being calculated, context from multiple independent systems must be compared and merged. The determination of weather and traffic-congestion properties of a particular travel route demonstrates how context from multiple systems can be correlated using the PCM and PCOnt. As shown in Figure 9, a travel route is defined in the Travel-Route Extended PCOnt as a real world object and is associated with a linestring location PC, a sequence of points that constitute a path. The WeatherAccessManager class is responsible for determining the current weather along a travel route. The pseudo code below illustrates the algorithm used by the WeatherAccessManager to identify the prevailing weather conditions along a particular route.

```
For each point on the travel route
  Find weather area containing the point
  For each weather area
    Find weather record
    Find the average weather record
```

A weather area denotes a geographical region specified by the weather system. A weather record describes the weather properties of a particular weather area. A travel route may pass through one or more weather areas. The location PC of each point along the travel route is submitted to the Spatial API in a location query to find the weather area containing that point.

Similarly, context can be correlated based on identity PC, time PC, or quality of service PC. The Spatial API enables primary context queries to be formed based on the primary context values of other context, essentially merging the information. In the original iTransIT model, much of the correlation of context was left to the pervasive service developer. Modelling context according to the PCM with associated primary context facilitates the correlation of context from independent, distributed pervasive-computing systems and assists developers of pervasive services when merging context.



For Project: TravelRouteExtendedPCOnt

owl:Thing
pcont:PrimaryContext
pcont:SpatialObject
pcont:DataObject
TravelRouteConditions
pcont:RealWorldObject
TravelRoute
pcont:SystemObject
opengis:Geometry
RouteProperty

For Class: TravelRoute (instance of owl:Class) □ Inferred View

Property	Value	Lang
rdfs:comment	A possible path from a particular start point to a destination point	

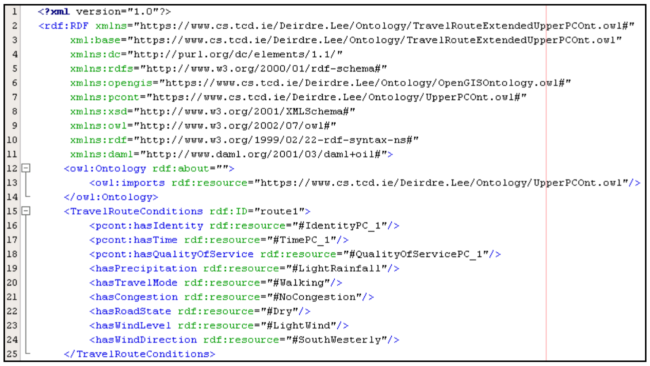
pcont:RealWorldObject
hasTravelRouteConditions **only** TravelRouteConditions
hasTravelRouteConditions **some** TravelRouteConditions
pcont:hasLocation **only** pcont:LinestringLocationPC

pcont:hasIdentity **only** pcont:IdentityPC [from pcont:SpatialObject]
pcont:hasIdentity **exactly** 1 [from pcont:SpatialObject]
pcont:hasLocation **exactly** 1 [from pcont:RealWorldObject]

Figure 9. The Travel-Route Extended PCOnt

5.1.3 *Reasoning over Context*

Our hybrid context-model introduces ontological modelling to the iTransIT framework. The PCOnt formally specifies all entities in the domain and the relationships that hold between these entities. One of the major benefits of ontologies, as opposed to other context modelling approaches, is the innate reasoning functionality provided by them. STIS2 takes advantage of this feature by reasoning over the five possible travel routes and inferring the optimal route based on restrictions set out in an Extended PCOnt. An optimal travel route is defined as the preferred route of a user, based on current weather and congestion conditions. For example, if the weather is dry, traffic congestion is high, and distance is less than 2km, a user may prefer to walk. However if the weather is icy and traffic congestion is high, a user may prefer to take a tram. The Travel-Route Extended PCOnt defines the conditions for a travel route to be deemed acceptable, such as good walking route conditions and good driving route conditions. The STIS2 retrieves current weather and traffic conditions for a particular route and, based on the shared understanding of the context, it can easily merge the context from unique systems and model it according to the Travel-Route Extended PCOnt. An example of such a route is shown in Figure 10.



```
<?xml version="1.0"?>
<rdf:RDF xmlns="https://www.cs.tcd.ie/Deirdre.Lee/Ontology/TravelRouteExtendedUpperPCOnt.owl#"
     xml:base="https://www.cs.tcd.ie/Deirdre.Lee/Ontology/TravelRouteExtendedUpperPCOnt.owl"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:opengis="https://www.cs.tcd.ie/Deirdre.Lee/Ontology/OpenGISOntology.owl#"
     xmlns:pcont="https://www.cs.tcd.ie/Deirdre.Lee/Ontology/UpperPCOnt.owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:daml="http://www.daml.org/2001/03/daml+oil#">
    <owl:Ontology rdf:about="">
        <owl:imports rdf:resource="https://www.cs.tcd.ie/Deirdre.Lee/Ontology/UpperPCOnt.owl"/>
    </owl:Ontology>
    <TravelRouteConditions rdf:ID="route1">
        <pcont:hasIdentity rdf:resource="#IdentityPC_1"/>
        <pcont:hasTime rdf:resource="#TimePC_1"/>
        <pcont:hasQualityOfService rdf:resource="#QualityOfServicePC_1"/>
        <hasPrecipitation rdf:resource="#LightRainfall"/>
        <hasTravelMode rdf:resource="#Walking"/>
        <hasCongestion rdf:resource="#NoCongestion"/>
        <hasRoadState rdf:resource="#Dry"/>
        <hasWindLevel rdf:resource="#LightWind"/>
        <hasWindDirection rdf:resource="#SouthWesterly"/>
    </TravelRouteConditions>
```

Figure 10: Travel Route Information expressed in PCOnt

The hasTravelMode, hasPrecipitation, hasRoadState, hasWindLevel, hasWindDirection, and hasCongestion properties are given the appropriate values according to the weather and traffic-congestion context associated with each travel route. Similarly, each route is modelled and reasoning is performed to infer if any of the defined travel routes match a user's preferences. Inferring the optimal travel-route in STIS2 demonstrates the powerful reasoning capabilities introduced by the PCOnt. The PCOnt complements PCM by introducing semantics to the context, providing a shared understanding of the context, and facilitating increased autonomy through the formal specification of the domain.

5.2 Performance Evaluation

The previous section presented the benefits of using a hybrid approach to context modelling in a real-world pervasive service. However, these benefits carry additional costs, such as the time it takes to retrieve live data and the time it takes to reason over this data dynamically. This section analyzes the applicability and usability of the PCM and PCOnt hybrid approach by comparing the performance of the STIS2 with that of the original STIS realisation, hereafter named STIS1.

Experiments were carried out to measure the latency for retrieving a route, using both STIS1 and STIS2. Figure 11 displays the retrieval latency for a route from a particular starting point to a particular destination in Dublin city. This measurement has been taken 100 times and the results shown in Figure 11 are the averages of these measurements. The mode of transport, such as walking, cycling, etc., was predetermined to preclude reasoning in STIS2 so that route retrieval latencies for STIS1 and for STIS2 can be compared. These results show that both implementations of the STIS (and the Spatial API) expose similar route-retrieval latencies for calculating a route from static information demonstrating that the advanced Spatial API of STIS2 has been introduced without adding significant computational overhead.

This experiment has been repeated for a number of distinct routes within Dublin. These measurements have all yielded similar results and were therefore omitted from this paper.

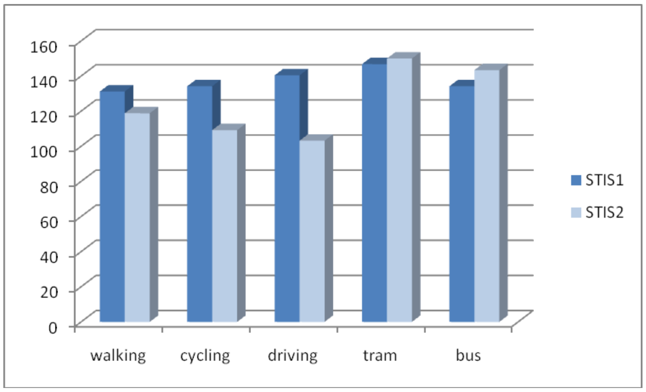


Figure 11. Average route retrieval latency [ms] for modes of transport

In STIS1, a user needs to specify his/her preferred transport mode explicitly, while in STIS2 a travel route is generated automatically based on predefined user preferences and on current context including weather conditions and traffic congestion levels. Such dynamic reasoning over context is made possible through the use of the PCOnt, and as such, was not supported by STIS1.

Further experiments were carried out to measure the overhead introduced by reasoning. Table 2 shows the latency for retrieving a reasoning-based route in STIS2 broken down according to its main parts. Calculating a travel route for a user entails first calculating a travel route for all five modes of transport, then determining the current weather conditions along each travel route, and finally, with the aid of the Travel-Route Extended PCOnt, inferring which of the travel routes will be most favourable to the user. This experiment has been repeated multiple time and the first 5 of these runs are shown in Table 2.

Table 2. The latency for generating a reasoning-based route in STIS2

Run	Route Retrieval Latency [ms]	Weather Retrieval Latency [ms]	Route Reasoning Latency [ms]	Total Latency [ms]
1	646.8	3344	1563	5553.8
2		469	1312	2427.8
3		375	1296	2317.8
4		344	1172	2162.8
5		359	1109	2114.8
Average	646.8	978.2	1290.4	2915.4

The weather retrieval latency represents the time for retrieving the prevailing weather conditions along a particular travel route. This involved correlating the travel-route context and the weather context, which is stored as OWL data. Correlation of context is made possible through the use of primary context in the PCM. As can be seen from Table 2, the very first instance of retrieving the weather conditions along a particular travel route takes significantly longer than subsequent retrievals. This is due to issues related to software initialisation. During the start-up period, the Java virtual machine may compile and optimise methods into native code, which results in performance differences between initial and subsequent code execution [22]. Once the current weather for a particular route has been determined, reasoning is performed to infer which of the routes is considered optimal. For example, if the weather along a route is dry, the distance is less than 2km, and the user has specified a preference for environmentally-friendly mode of transport, the optimal route may be the cycling route. On the other hand, a tram-based route may be considered optimal for icy weather conditions. The route reasoning latency captures reasoning over the context to infer the optimal route. Using the Travel-Route Extended PCOnt, the best travel route conditions are inferred from the asserted model.

The total latency for retrieving a route for all modes of transport, for retrieving the weather conditions along each route, and for reasoning over this data was found to converge towards an average of 2 seconds. These measurements were repeated for various start and destination points, with similar results. Although a service response latency of around 2 seconds (for non initial measurements) has been deemed acceptable from a user's perspective [23], this represents a significant increase from the route-retrieval latency for a route with a predefined travel-mode as shown in Figure 11. The increase is due to the time required to retrieve a travel route for each of the five travel modes, to calculate the weather context along each route, and to reason over the context and infer the optimal route. Although the autonomous determination of the optimal route based on current conditions carries an overhead, we argue that this is offset by the benefits introduced by modelling context using PCM and PCOnt. Benefits include common access to context provided by independent, distributed, heterogeneous sources, sharing and correlation of context between individual pervasive-computing system, providing a universal interpretation of context by all pervasive computing environment entities, and reasoning about explicit context, so that implicit knowledge can be inferred. These, ultimately, lead to a much improved user experience and quality of the recommended travel route.

5.3 Evaluation Analysis

We further assess the benefits of the PCM and PCOnt hybrid approach by summarising and analysing our previous findings in light of the criteria that motivate this work.

- **Accessibility:** The Spatial API, PCM's object-oriented interface, allows homogenous access to context data via primary context queries. The implementation details of the original context source and structure is hidden from the user because of object encapsulation.

- **Manageability:** Object-oriented modelling is an established and widely-used approach to system modelling, and especially, to context modelling. There are many mature commercial and open-source technologies available that support object-oriented data discovery, communication, access, and storage, such as the Java and .NET suite of tools, object-relational mapping techniques for converting data between relational schemas and objects, and object-relational databases, which are database management systems similar to a relational database, but with an object-oriented database model. Conversely, ontology modelling is a relatively new concept, and, as such, it does not yet have mature technologies support. For example, RDF stores such as Sesame, Jena, and YARS are not as efficient, scalable, or widely-adopted as relational database technology. Therefore, adopting a hybrid approach to context modelling, incorporating the established techniques of object-oriented modelling and the semantic benefits offered by ontology modelling, results in rich, extensible, and manageable context models.

- **Correlation:** Using the PCM and PCOnt, the sharing and correlating of context data may be performed at two levels. Firstly, the primary context objects associated with each PCM spatial object may be used to programmatically correlate data from disparate iTransIT systems. In this way, objects with corresponding identity, location, time, or quality of service primary context may be found to relate to each other. Additionally, semantics is intrinsic to context data defined in terms of the PCOnt, which, ultimately, enables autonomous context correlation. Using an ontology query language, such as SPARQL, data correlations across the common data model may be exposed.

- **Extensibility:** For a new system to integrate with the iTransIT framework, it must expose its context data as spatial objects with associated primary context through an implementation of the Spatial API. As the Upper PCOnt is based on PCM, all context modelled according to the PCM can be mapped directly to Upper PCOnt instances, thereby inherently adopting semantic features, such as autonomous correlation, consistency checking, and reasoning. If such a system needs to add richer semantics to its context, for example, to facilitate more complex semantic reasoning and data correlation, it simply needs to define a new Extended PCOnt describing its data. Such integration has explicitly been addressed in the design of the iTransIT framework.

6. CONCLUSIONS AND FUTURE WORK

Components, devices, users, and system must be able to interact with one another and exchange context seamlessly in order to achieve true pervasiveness. This paper presented an approach to facilitate management, sharing, and correlation of context between components in a large-scale pervasive computing environment. The main contribution of this paper is the Primary-Context Model (PCM) and the Primary-Context Ontology (PCOnt), a hybrid approach to modelling context in the iTransIT framework.

Our hybrid approach to modelling context incorporates both object-oriented context modelling techniques and ontological context modelling techniques, so that the benefits of both approaches are exploited, while negating their shortcomings. The PCM is based on object-oriented principles and is built on the notion of a spatial object. A spatial object represents any entity in the pervasive computing environment and is sub-classed by the real world object, the system object, and the data object. Any system in the large-scale domain can be represented using a combination of these spatial objects. As the actual context is encapsulated within each object, this approach lends itself to scalability, extensibility, and distribution. Primary context is a key feature of the PCM and PCOnt, as it is used to index other context and to correlate context from individual systems. Each spatial object is associated with specific primary context, namely identity PC, location PC, time PC, and quality PC, and may be accessed through a common interface, the Spatial API. The Spatial API accepts object queries based on primary context associated with the object, enabling all context in the environment to be readily accessible to pervasive services. The PCOnt, which is divided into the Upper PCOnt and the Extended PCOnts, captures a formal semantic specification of the objects, concepts, and entities in a large-scale pervasive computing domain (and its sub-domains) and of the relationships that hold among them. This provides a common understanding of context in an intelligent environment and enables advanced reasoning/inference, consistency checking and, ultimately, allows systems and services to act with a certain degree of autonomy.

The PCM and PCOnt approach of distinguishing between the management of context objects and the management of context knowledge has been evaluated by modelling context in a prototypical realisation of the iTransIT framework. A pervasive user service for dynamic journey planning in the Dublin metropolitan area has been realised to support primary context-based queries, correlation of context across multiple systems and inference of new context from existing context. This has demonstrated that the PCM and the PCOnt can support context integration and reasoning across independent, distributed systems.

While this paper has made several contributions towards context modelling in large-scale pervasive computing domains some research remains for future work. The division of the PCOnt into the Upper PCOnt and the Extended PCOnts lends the design to extensibility, allowing for straightforward addition of further Extended PCOnts and alleviates the constraints on all iTransIT systems having to comply with a single ontology. As a result, a pervasive service (or system) requires only those Extended PCOnts that are relevant to its operation. Further work might address storage, management, and discovery of the Extended PCOnts distributed between iTransIT systems and pervasive services.

Acknowledgments. The work described in this paper was supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero - the Irish Software Engineering Research Centre (www.lero.ie). The authors would also like to thank Dublin City Council for providing information on Dublin's transport infrastructure, which made this work possible.

7. REFERENCES

[1] M. Weiser, "The Computer for the 21st Century," in *Scientific American*, vol. 265, 1991, pp. 94-104.

[2] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, vol. LNCS 1707. Karlsruhe, Germany Springer-Verlag, 1999, pp. 304-307.

[3] R. Meier, A. Harrington, K. Beckmann, and V. Cahill, "A Framework for Incremental Construction of Real Global Smart Space Applications," *Elsevier Pervasive and Mobile Computing*, 2009, to appear.

[4] R. Meier, A. Harrington, and V. Cahill, "A Framework for Integrating Existing and Novel Intelligent Transportation

Systems," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems (IEEE ITSC´05)*. Vienna, Austria: IEEE Computer Society, 2005, pp. 650-655.

[5] S. Brennan and R. Meier, "STIS: Smart Travel Planning Across Multiple Modes of Transportation," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (IEEE ITSC 2007)*. Seattle, Washington, USA: IEEE Computer Society Press, 2007, pp. 666-671.

[6] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," presented at presented at the First International Workshop on Advanced Context Modelling, Reasoning and Management in conjunction with UbiComp 2004, Nottingham, UK, 2004.

[7] G. Judd and P. Steenkiste, "Providing Contextual Information to Pervasive Computing Applications," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*. Fort Worth, TX, US: IEEE Computer Society, 2003, pp. 133-142.

[8] N. H. Cohen, J. Black, P. Castro, M. Ebling, B. Leiba, A. Misra, and W. Segmuller, "Building Context-Aware Applications with Context Weaver," IBM Research Division, T.J.Watson Research Center, NY, US 2004.

[9] O. Lehmann, M. Bauer, C. Becker, and D. Nicklas, "From Home to World - Supporting Context-aware Applications through World Models," in *Proceedings of Second IEEE International Conference on Pervasive Computing and Communications (Percom'04)*. Orlando, Florida: IEEE Computer Society, 2004, pp. 297-308.

[10] A. Ranganathan and R. H. Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments," in *Proceedings of the ACM/IFIP/USENIX International Middleware Conference*, vol. LNCS 2672. Rio de Janeiro, Brazil: Springer-Verlag, 2003, pp. 143-161.

[11] P. Korpipaa, J. Mantyjarvi, J. Kela, H. Keranen, and E.-J. Malm, "Managing Context Information in Mobile Devices," *IEEE Pervasive Computing*, vol. 2, pp. 42-51, 2003.

[12] M. Khedr and A. Karmouch, "ACAI: Agent-Based Context-Aware Infrastructure for Spontaneous Applications," *Journal of Network and Computer Applications*, vol. 28, pp. 19-44, 2005.

[13] H. Chen, F. Perich, T. Finin, and A. Joshi, "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications," in *Proceedings of The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '04)*. Boston, MA, US: IEEE Computer Society, 2004, pp. 258-267.

[14] T. Gu, H. K. Pung, and D. Q. Zhang, "A Service-Oriented Middleware for Building Context-Aware Services," *Journal of Network and Computer Applications*, vol. 28, pp. 1-18, 2005.

[15] M. Strimpakou, I. Roussaki, and M. E. Anagnostou, "A Context Ontology for Pervasive Service Provision," in *Proceedings of the Twentieth International Conference on Advanced Information Networking and Applications (AINA'06)*, vol. 2. Vienna, Austria: IEEE Press, 2006, pp. 5-9.

[16] F. Paganelli, G. Bianchi, and D. Giuli, "A Context Model for Context-Aware System Design Towards the Ambient Intelligence Vision: Experiences in the eTourism Domain," in *Proceedings of Universal Access in Ambient Intelligence Environments, 9th ERCIM Workshop on User Interfaces for All*, vol. LNCS 4397. Königswinter, Germany: Springer-Verlag, 2006, pp. 173-191.

[17] C. Becker and D. Nicklas, "Where do spatial context-models end and where do ontologies start? A proposal of a combined approach," in *presented at the First International Workshop on Advanced Context Modelling, Reasoning and Management in conjunction with UbiComp 2004*. Nottingham, UK, 2004.

[18] M. Bauer, C. Becker, and K. Rothermel, "Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks," *Personal and Ubiquitous Computing*, vol. 6, pp. 322-328, 2002.

[19] M. K. Smith, C. Welty, and D. L. McGuinness, "OWL Web Ontology Language Guide," in *W3C Recommendation*, 2004.

[20] M. Khedr and A. Karmouch, "ACAI: agent-based context-aware infrastructure for spontaneous applications," *Journal of Network and Computer Applications*, vol. 28, pp. 19-44, 2005.

[21] M. A. Strimpakou, I. G. Roussaki, and M. E. Anagnostou, "A context ontology for pervasive service provision," in *Proceedings of the Twentieth International Conference on Advanced Information Networking and Applications (AINA'06)*. Vienna, Austria: IEEE Computer Society, 2006, pp. 1-5.

[22] Sun Microsystems Inc., "Java Tuning White Paper," White Paper http://java.sun.com/performance/reference/whitepapers/tuning.html, 2005.

[23] R. B. Miller, "Response time in man-computer conversational transactions," in *Proceedings of the Fall AFIPS Joint Computer Conference (part 1)*, vol. 33. San Francisco, California, USA: ACM, 1968, pp. 267-277.