New Generation Computing, 27(2009)177-196
Ohmsha, Ltd. and Springer

Invited Paper

Modeling and Processing Information for Context-Aware Computing: A Survey

Christian HOAREAU
The Graduate University for Advanced Studies
Shonan Village, Hayama, Kanagawa 240-0193, JAPAN
Ichiro SATOH
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, JAPAN
{hoareau,ichiro}@nii.ac.jp

Received 07 May 2009

Abstract Context-awareness is emerging as a central issue in ubiquitous computing research. Context-aware computing refers to the idea that computing devices can sense and react to the physical environment where they are deployed. A great deal of research on context-awareness has been conducted to explore and address the various challenges related to context acquisition, representation, distribution, and abstraction. This paper surveys the most relevant approaches to modeling context for ubiquitous computing. It also evaluates how the existing works utilize contextual information, with respect to the query processing approaches used to access and manage that information. We also discuss typical problems, shortcomings, and challenges posed by context modeling at large, and highlight some proposals to address some of them.

Keywords: Context Awareness, Context Modeling, Ubiquitous Computing, Query Processing.

§1 Introduction

Contextual Adaptation. Context-awareness is embedded into the fabric of our human nature. Consciously or unconsciously, we often derive our actions and behaviors from a particular set of circumstances, the *context*. This context is rich, subjective and ever-changing. The actual meaning of one's context, and thus its relevance, depends on what activity one might be engaged in. Context is multi-dimensional; it can encompass perceptual information – environmental (e.g., the level of pollution), physical (e.g., one's current location), social (e.g., one's family and colleagues), or temporal (e.g., the time of the day), just to name a few. Non-perceptual information, like memories of one's past experiences or one's emotional state, is part of one's context as well.

For one thing, our context-awareness stems from a huge number of stimuli coming from our surrounding world. These stimuli are first captured by an extremely sophisticated sensory system, encoded by our brains into electrical signals, then processed in one way or another, stored for later retrieval, and/or eventually lost. This process gives us the ability to sense, and adapt our behaviour to, the world around us. We are context-aware beings, and we would like to see our ever-growing ubiquitous computing ecosystem – homes, cities, personal and wearable devices – become context-aware as well. Ultimately, the technology will adapt to us.

Context-Aware Computing. The idea of context-aware computing was introduced in some of the pioneering work on ubiquitous computing research and has been subject to extensive research since. Context-aware computing stems from the vision articulated by Marc Weiser(64) in his seminal 1991 paper: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it". The goal of context-aware computing is to acquire and utilize information pertaining to the physical world, and then select, configure, and provide a variety of services accordingly.

Context-aware systems are concerned with the acquisition of context (e.g., using sensors to perceive a situation), the abstraction and understanding of context (e.g., matching a perceived sensory stimulus to a context), and application behaviour based on the recognized context (e.g., triggering actions based on context). Context-awareness is regarded as an enabling technology for ubiquitous computing systems. While there seems to be a consensus on the importance and usefulness of capturing and processing contextual information, its very nature, scale, and complexity poses major challenges:(35)

- Acquisition of context: Contextual information is acquired from non-conventional and heterogenous sensors, which may be connected through a network.
- Abstraction of context: It should be abstracted, because it may depend on the sensors that acquire it.(55)
- Understanding of context: It needs to be represented, managed, and used in relevant data structures and algorithms to be processed by context-aware services.

Of these technical issues, most researchers have addressed the third. In fact, several international workshops have been dedicated to these very issues in

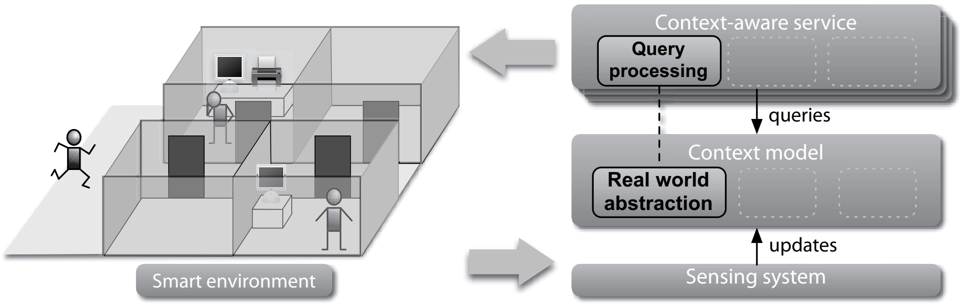

Fig. 1 Context-aware services can sense and react to the physical environment where they are deployed.

recent years.[*1]

Scope, Contributions and Organization of this Survey. This paper reviews related work that has focused on contextual information processing and management in addition to context modeling. We focus on the conceptual tools that have been proposed to address pertaining to query processing. We overview the most relevant approaches to modeling context for ubiquitous computing (Sect. 2). We provide an analysis of the existing surveys that have focused on context modeling, and discuss the relevance of their underlying evaluation frameworks (Sect. 3). We propose an original perspective of context modeling from the stance of information management (Sect. 4). We place particular emphasis on the query-processing approaches used to manage contextual information (Sect. 5). We discuss various problems, shortcomings, and challenges posed by context modeling at large, and highlight some proposals to address some of them (Sect. 6).

Other important topics that are not directly related to computer-system issues fall outside the scope of this survey. Some of these are context-awareness applied to the fields of human-computer interactions, expert systems, and software agents. As a side note, we have not specifically overviewed context-aware systems that have been developed since their introduction in the early 90s, nor have we looked at context middleware, infrastructures and toolkits. We refer to Strimpakou et al.'s review63) of the main projects and to Baldauf et al.3) for their comprehensive overviews.

§2 Context Modeling

Much of the context information involved in pervasive systems is derived from sensing devices. There is usually a significant gap between sensor output and the level of information that is useful to applications. Therefore, this gap

---

[*1] Workshop on Modeling and Reasoning in Context (MRC 2008 - http://events.idi.ntnu.no/mrc2008/). Workshop on Context Modeling and Reasoning (CoMoRea 2009 - http://nexus.informatik.uni-stuttgart.de/COMOREA/). Workshop on Context Modeling and Management for Smart Environments (CMMSE'08 - http://flash.lakeheadu.ca/ rbenlamr/cmmse08/)

must be bridged by various kinds of processing of context information before the information is passed to context-aware services.

2.1 Location Models

Location is one the most typical contextual information. Context-aware applications primarily use location – of people, objects, and computational devices – as their main source of contextual information. Indeed as significant progress is achieved in location-sensing and -tracking technologies – e.g., the Global Positioning System (GPS), Radio Frequency IDentification (RFID) tags, and ultrasonic-based tracking systems – location has become de-facto the preeminent contextual information. The functions of a location sensor can be classified into two typical approaches: positioning and tracking. The former measures its own location with the help of the infrastructure, and the latter measures the location of other located objects. Their output may be in raw coordinates, whereas an application might be interested in identifying the building or room users are in. Moreover, requirements can vary between applications. Therefore, a context model must support multiple representations of the same context in different forms and at different levels of abstraction, and must also be able to capture the relationships that exist between alternative representations.

There are many types of location information, e.g., rooms in a house and latitude-longitude on the earth. These different types need to be expressed with different coordinates and data structures. Most existing context-aware systems maintain contextual information in the output format of the underlying sensing system. Therefore, they tend to depend on their current sensing systems. They cannot be used when their sensing systems are changed. Therefore, several researchers have proposed and defined data structures for abstracting and modeling location-based information to be as independent of sensing systems as possible. Becker and Dürr reviewed the location models.6)

Geometric Models. They represent the positions of people and objects as geometric coordinates, captured by positioning sensors. The most prominent reference-coordinate system for outdoor environments is GPS, which is widely used for navigation-based applications. However such fine-grained information is often meaningless in human interactions and lacks any semantics for describing the relations between locations. Geometric coordinates are therefore contextualized with secondary human-readable information. In navigation-based applications, GPS coordinates are represented by moving points on city maps.

Symbolic Models. They are applied in areas where information about data interconnectivity or topology is more important, or as important, as the data themselves. In these applications, the data and relations between them, are usually at the same level. Introducing graphs as a modeling tool has several advantages for these types of data. It allows for data to be more naturally modeled. Graph structures are visible to the user and they allow a natural way of handling application data, e.g., hypertext or geographic data. Graphs have the advantage of being able to keep all the information about an entity in a single node and show related information by arcs connected to it. Existing symbolic-based models can be viewed as subsets of this category.

Hybrid Models. We need to choose location models carefully with respect to the requirements for spatial reasoning and modeling effort involved. Both humans and computers can easily understand the symbolic model, but there is no precision with geometric models. Therefore, several researchers have proposed eclectic models, called *hybrid location models*.33) A hybrid model has both the advantages and disadvantages of the previous tow models. A few researchers have proposed semantic models, which have rather been like focusing purely on position by using ontologies.

2.2 Context Models

While the computer-science community has initially perceived context as a matter of user location, as Dey stated,19) in the last few years this notion has not simply been considered as a state, but part of a process in which users are involved; thus, sophisticated and general context models have been proposed to support context-aware applications that use them to (a) adapt interfaces, (b) tailor the set of application-relevant data, (c) increase the precision of information retrieval, (d) discover services, (e) make user interactions implicit, or (f) build smart environments.8) For example, a context-aware mobile phone may know that it is currently in a meeting room, and that its carrier has just sat down. The phone may conclude that the user is currently in a meeting and reject any unimportant calls.59)

Computing systems may need to understand the real world to provide services according to context within it. But unlike humans, they cannot maintain all the information about the real world. Consequently, they are required to extract and maintain their context of interest as models of the real world inside them. Most existing context models have been designed and implemented in an ad-hoc manner, in the sense that their context models are premature or dependent on the underlying sensing systems. A few researchers have begun to look at the frameworks for context-aware systems more generally, independently of specific applications, including context middleware, infrastructures, and toolkits.4)7)31)54) Such work facilitates the building of context-aware systems. Tools for end-users to program context-aware systems are also being built on top of these infrastructures.

Chen and Kotz12) thoroughly reviewed the research on context-aware mobile computing. They provided a complete overview of the major context-aware applications that had been built, and gave a snapshot of the underlying context models. They highlighted six data structures:

Key-value Model. A key (or identifier) in this simple model corresponds to an attribute of the environment that has a value. This value is usually measured by sensors embedded in the environment. For example, `<Room 31 temperature,`

`24>` is such a `<key, value>` pair. Key-value models were used by Schilit et al.58) to manage location information. For obvious reasons, this simple model has been replaced by more expressive and flexible alternatives. The model has problems with its expressiveness but it can be directly maintained on large-scale distributed systems like cloud computing, where data are maintained on a huge number of computers that are loosely coupled.

Markup Model. Markup-based models use a hierarchical data structure consisting of markup tags with attributes and content. Profiles represent typical markup-scheme models. Typical examples are the Composite Capabilities/Preference Profile (CC/PP) and User Agent Profile (UAProf) which are encoded in RDF/S with XML notation.32)

Object-oriented Model. This is based on the concept of objects and relationships between them as in the object-oriented programming paradigm. Rather than using implementation-based concepts like records, object-oriented models provide flexible structuring capabilities by fully-fledged object-orientation mechanisms such as encapsulation, reusability, and inheritance. Objects are used to represent various types of contextual information (e.g., temperature and location); they encapsulate the operations used for context processing.

Logic-based Model. Facts, expressions, and rules are used to define a context model. Contextual information comprise facts described by some rule-based mechanisms. Ranganathan et al.51) and Katsiri and Mycroft36) applied first-order logic to reason with contextual information, and Henricksen26) applied the same techniques to describe and reason with situations. Others like Sohn61) provided end-users with a toolkit that allowed them to develop context-aware applications by specifying rules. Overall, rule-based programming has proved to be intuitive and well-suited to prototyping context-aware applications.

Ontology. Numerous projects use ontologies and the tools from the “Semantic Web” to represent and reason with context.14) Ontology-based techniques support a vocabulary for situation predicates42), so that the data representations in these projects might be shared across different systems, or even retrieved from a store over the Web. Also, ontologies can also provide vocabularies and additional semantics to sensor predicates, i.e., such predicates can represent context attributes specified by an ontology. This approach not only has the advantages but also the disadvantages of ontology-based data representation.

Situation Logic. Loke42) took rule-based programming further by representing situations and not only programming rule-based triggers for context-aware actions. He explored the integration between context infrastructures and LogicCAP, with the infrastructures providing sensed contextual information and LogicCAP as the programming layer.

§3 Evaluating Context Models

This section reviews some existing survey papers and discusses their methods to classify and evaluate context models. These papers are presented in chronological order of publication.

3.1 Requirement-based Evaluation Framework

In a 2004 paper, Strang and Linnhoff-Popien[62)] evaluated six modeling approaches that are deemed to be the most relevant to context-aware computing. Their evaluation framework was based on a set of six requirements that ubiquitous computing systems should satisfy.

Distributed Composition. Context-aware systems are distributed, and thus do not have any central instance responsible for the creation, deployment, or maintenance of data and services, particularly context descriptions. Instead, the composition and administration of a context model and its data varies with notably high dynamics in terms of time, network topology, and source.

Partial Validation. On the structural as well as on instance level, even if there is no single place or point in time where the contextual knowledge is available on one node as a result of distributed composition. This is particularly important because of the complexity of contextual interrelationships, which make any modeling intention error-prone.

Richness and Quality of Information. The quality of information delivered by sensors varies over time, as well as the richness of information provided by different kinds of sensors characterizing an entity in an ubiquitous computing environment, may differ. Thus, a context model appropriate for use in ubiquitous computing should inherently support both quality and richness.

Incompleteness and Ambiguity. The set of contextual information available at any point in time characterizing relevant entities in ubiquitous computing environments is usually incomplete and/or ambiguous, particularly if this information is gathered from sensor networks. This should be covered by the model, for instance by interpolating incomplete data on the instance level.

Level of Formalism. It is always a challenge to describe contextual facts and interrelationships in a precise and traceable way. For instance, carrying out the task “print document on the nearest printer” requires a precise definition of terms used in the task, for instance what “nearest” means. It is extremely important that all the entities share the same interpretation of the data that are exchanged, as well as their underlying meaning.

Application to Existing Environments. From the implementation perspective, it is important for a context model to be applied within an existing infrastructure of ubiquitous computing components.

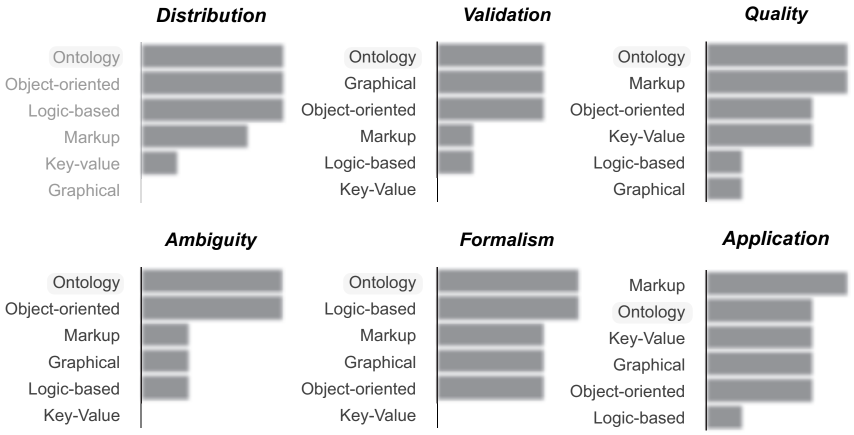

Fig. 2 Evaluation of the six major context-modeling approaches based on six requirements defined by Strang et Linnhoff-Popien.[62)]

In addition to the models previously described, Strang and Linnhoff-Popien added the graphical models. Their conclusions are summarized in Fig. 2. The results reveal that ontologies have the most expressive models and fulfil most of the requirements.

3.2 Data-based Evaluation Framework

Bolchini et al.[8)] provided a comprehensive evaluation framework for comparing context-aware services and their underlying context models. The results of their analysis were intended to help designers build context-aware systems and choose the context model that was the most appropriate to their systems' specifications and requirements. They classified the features of context models into three groups:

- Modeled aspects: The set of context dimensions managed by the model. This includes the notion of space and time – which can be represented as absolute or relative –, context history, subject, and user profile.
- Representational features: The general characteristics of the model itself. This includes the type and level of formalism, flexibility of the model, variable context granularity, and valid context constraints.
- Context management and usage: The way the context is built, managed, and exploited.

Bolchini et al. placed great emphasis on context-aware data-tailoring, and designed their evaluation framework according to this. Contextual data acquired from the sensing environment was used to filter the data normally processed by the applications. Context-data tailoring has three goals: (i) to provide more relevant data for users (e.g., time- and location-based), (ii) to match the physical constraints of the devices, and (iii) to improve the efficiency of query processing.

3.3 Discussion

Any context model is designed according to a specific set of requirements. These requirements are based on the nature of the contextual information to be handled by the context-aware services. In this section, we look at the characteristics of context information, and then discuss the requirements used to build the evaluation frameworks that were previously presented.

[1] Target-dependent context information

Contextual information can have different natures. Dey(19) distinguished between three kinds of contextual information or entities: places, people and things. Each entity is characterized by attributes that fall into one the following categories: identity, location status, or time. Context information can be characterized as static or dynamic. Static context information describes those aspects of a pervasive system that are invariant, such as a persons birthday.

The persistence of dynamic context information can be highly variable. E.g., the relationships between colleagues typically can endure for months or years, whereas a persons location and activity often change from one minute to the next. Persistence characteristics influence the means by which context information must be gathered. While it is reasonable to obtain largely static context directly from users, frequently changing contexts must be obtained by indirect means, such as through sensors.

Pervasive computing applications are often interested in more than the current state of the context. They can also rely on activities planned for the future. As a result, the contextual description might comprise context histories, both past and future.

[2] Requirements of services

Arguably, distributed composition is orthogonal to any context-modeling issues, because it is concerned with the architecture of the system and its underlying implementation. Also, the richness and quality of the information is both technology- and application-dependent. Data models have their own sets of requirements. For example, Korpip and Mntyjrvi(39) presented four requirements and goals having designed a context ontology:

- Simplicity: The expressions and relations should be as simple as possible to simplify the work done by applications developers.
- Flexibility and extensibility: The ontology should support the simple addition of new context elements and relations.
- Genericity: The ontology should not be limited to special kinds of context atoms but rather support different types of contexts.
- Expressiveness: The ontology should allow as many context states to be described as possible, and with arbitrary levels of details.

[3] Knowledge- vs. data-modeling

While summarizing Strang and Linnhoff-Popien's review(62) of the major context-modeling approaches, Moore et al.[44)] raised somes concerns about knowledge-based modeling, i.e. ontology-based modeling. In particular, they pointed out that the modeling function of a given ontology fails to address the issue of context processing. Acknowledging that ontologies provide a powerful support for knowledge reasoning, Roussaki et al.[53)] nevertheless mentioned the same shortcomings in regard to the acquisition, management, and processing of contextual data. Serious concerns about ontologies have been raised[46)] in the Semantic Web community as well. Pils et al.[50)] summarized these problems in terms of:

- Readability: Because of the strict standardisation of semantics, it is difficult for us to comprehend specifications defined by an ontology, and find relevant pieces of information.
- Costs: It can be expensive to adopt ontologies especially for small businesses like shops or restaurants.
- Disorder: The more the ontology model is structured, the more effort is required to increase its structure or even maintain the status quo. Indeed, the system $\mathcal{S}$ of semantic specifications exhibits the entropy $\mathcal{H}$: $\mathcal{H}(\mathcal{S}) = -\sum_{i=0}^{n} \pi(l_i) \cdot \log(\pi(l_i))$ where $\pi(l_i)$ is the probability that humans will misinterpret information, and it is proportional to the frequency of information use.

3.4 Context Modeling as Active Database Systems

Conventional databased management systems are passive. Data are created, retrieved, modified, and deleted only in response to operations issued by users. By contrast, context-aware computing is required to automatically carry out some services in response to certain changes in the real world, once conditions are being satisfied. Therefore, it needs some facets that active database systems have.[43)] Unfortunately, there have been few practical implementations of general-purpose active database systems,[65)] and as a result, several research projects on context-aware computing have attempted to implement active-database systems optimized for context-aware computing or to use various techniques found in the literature on active-database systems.

Satoh[55, 56)] introduces a world model for location-aware and user-aware services in ubiquitous computing environments, that can be dynamically organized like a tree based on geographical containment such as that in a user-room-floor-building hierarchy. Each node in the tree is constructed as an executable software component, that enables location-aware services to be managed without databases and by multiple computers. The proposed world model also provides a unified view of the locations, as they not only refer to physical entities and spaces, including users and objects, but also computing devices and services. The model is unique among other existing context-aware computing systems, because it consists of not only data elements but also programmable entities to define services. Therefore, the model itself automatically provides context-

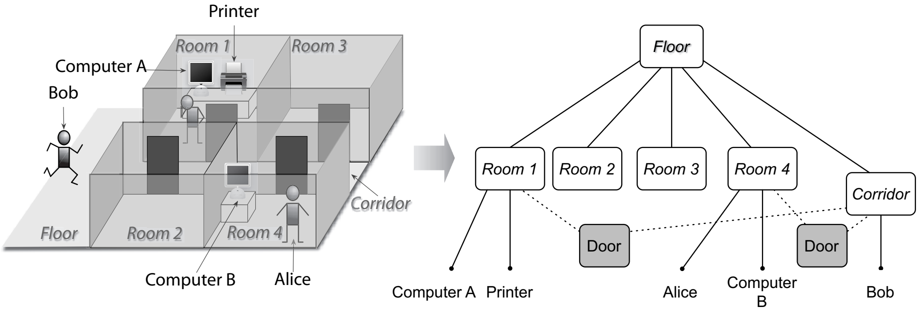

Fig. 3 Example of a graph-based data structure representing a real-world space

aware services in response to structural changes in the model corresponding the contextual changes in the real world.

§4 Context Modeling from the Perspective of Data Engineering

Current research on ubiquitous computing has paid attention to the design and implementation of application-specific location-aware services, such as smart rooms and navigation systems. These have often been designed for particular sensing systems (e.g., GPS and RFID tags) that capture context information about users and objects in the environment. In this section, we review work pertaining to the management of contextual data, which is often overlooked. In fact, most existing context-aware services maintain and process contextual information in an ad-hoc manner and/or rely on centralized and inadequate data infrastructures. Hereafter, we present the underlying connection between context models and query processing.

From a data-management viewpoint, contextual information is data that have to be captured, processed, and managed by the system. In Sect. 2, we discussed typical context models such as key-value and markup models. These models may be suitable for expressing contextual information, but not so for maintaining the information in computing systems. Context modeling shares very similars characteristics with data modeling. There has been a broad consensus in the database and data-engineering communities about the gap between the structure of data modeling and structure of data maintenance.

4.1 Mapping Contextual Information on Relational Database Model

A variety of data structures has been proposed and these have then faded away. Of these, the relation database (RDB) model, which was proposed by Edgar Codd in 1970,[16)] is still the reference model in database systems. Indeed, there have been many commercial and open-source implementations of RDB. Many researchers have attempted to maintain contextual information in existing RDB systems. However, the RDB model is different from context models. Nevertheless, several researchers have extended RDB systems to have the ability of supporting contextual information. Spatial databases represent the most typical extensions of RDB systems.[23)] They can store and query data related to objects in space, including points, lines and polygons, whereas typical RDB systems only support numerical and character types of data. They may be useful for maintaining geometric models for location-awareness, but they cannot support other models.

4.2 Resurgence of Graph-database Models

Graph database models are applied in areas where information about data interconnectivity or topology is more important, or as important, as the data itself.[2)] In these applications, the data and relations between the data, are usually at the same level. Graph database models are often used in knowledge representations such as semantic nets and frames. Introducing graphs as a modeling tool has several advantages for this type of data:

- It allows for a more natural modeling of data. Graph structures are visible to the user and they allow a natural way of handling applications data, e.g., hypertext or geographic data. Graphs have the advantage of being able to keep all the information about an entity in a single node and showing related information by arcs connected to it. Graph objects (like paths and neighborhoods) may have first order citizenship; a user can define some part of the database explicitly as a graph structure, allowing encapsulation and context definitions.[41)]
- Queries can refer directly to this graph structure. Associated with graphs are specific graph operations in the query language algebra, such as finding shortest paths and determining certain subgraphs. Explicit graphs and graph operations allow users to express a query at a high level of abstraction. To some extent, this is the opposite of graph manipulation in deductive databases, where fairly complex rules often need to be written. It is not important to require full knowledge of the structure to express meaningful queries. Finally, for purposes of browsing it may be convenient to forget the schema.
- For implementation, graph databases may provide special graph storage structures, and efficient graph algorithms for accomplishing specific operations.

4.3 Semi-structural Data Models for Contextual Information

There has been an avalanche of work on *semi-structured* data in the last decade in database engineering research, because XML, which is a data representation based on a semi-structured model has been widely used. Several researchers have attempted to represent contextual information using XML-based notation, in particular the RDF (Resource Description Framework) notation. RDF is a standard put forward by the the World Wide Web Consortium (W3C) to mainly represent metadata on Web data. RDF represents data resources with object-attribute-value groups. It provides a means of adding semantics to a document without making any assumptions about its structure. Therefore, RDF can support the complicated data structures of contextual information. In fact, there have been many projects that have attempted to represent contextual information in XML or RDF. However, XML and RDF notations are not suitable for maintaining contextual information. Contextual information dynamically evolves according to changes in the real world, whereas the notations are static.

§5 Query for Contextual Information

Query mechanisms are indispensable for most database systems.60) There has been a broad consensus about what these conceptual tools should offer: data structuring, description, maintenance and some mechanisms to retrieve or query the data.2) However, research on contextual information has focused on the representation of information. Instead, few researchers have paid attention to query mechanisms for contextual information. This is a serious obstacle in context-aware services.

5.1 Hybrid Approaches

Aiming at providing contextual-information retrieval that is both scalable and robust, Roussaki et al.53) enhanced the database with some location-based retrieval mechanisms. They implemented a hybrid context model that maintained (i) symbolic entities like streets and buildings, and (ii) objects located in a coordinate system. The context database maintained a hierarchy of entities. By combining the semantical expressiveness of ontologies with the hierarchical structure of tree-based location models, they could dispatch and route the queries along the context database hierarchy and obtain a significant increase in information retrieval. The same approach has been developed further to improve context filtering.50)

Ilarri et al.30) investigated query processing for location-based services. They targeted outdoor applications that have to handle a fair number of moving objects in real-time. Responsiveness and performance are therefore critical. The authors focused on geometric location models that provide both quantitative and high-resolution data on moving objects. Acknowledging that fine-grain GPS coordinates may be inappropriate for many location-based applications (high-resolution requires more overhead), they introduced the notion of location granules. Granules are geographical areas that can be defined at different scales, e.g., freeways and buildings. However, the semantics of the granules are rather poor, since relationships between granules are not explicitly defined. Therefore, the query language used to retrieve the location cannot benefit from the spatial semantics of the granules.

5.2 Application-independent Query Processing

Grossniklaus and Norrie22) proposed an object-oriented version control to address the challenges of data management in context-aware services. The query-processing framework was application-independent. Indeed, as many frameworks were specific to a single application domain, their notion of context (in particular the context dimensions e.g., users, devices, and environmental factors) was only relevant to a subset of applications. Moreover, representation and storage are often implied by the context model. They proposed a general approach, not limited to any particular context model. It was similar to Bolchini et al.'s8) that emphasized the importance of context-based data tailoring.

Perich et al.48) proposed a solution to managing data in pervasive computing applications. It broadly consisted of two parts: treating the devices as semi-autonomous entities guided in their interactions by profiles and context, and designing a contract-based transaction model. The profile was grounded in a semantically rich language for representing information about users, devices, and data objects each described in terms of "beliefs", "desires", and "intentions" - a model that has been explored in multi-agent interactions. They introduced data-based routing algorithms, semantic-based data caching, and replication algorithms enabling mobile devices to utilize their data-intensive vicinities.

5.3 Application of Formal Methods

Hoareau and Satoh28) introduced a query-processing framework for location-based services that is based on model checking. As most context-aware systems tend to query the underlying location models in an ad-hoc manner, it is difficult to guarantee the quality of the results and the reliability of context-aware services. Hoareau and Satoh proposed (i) a hybrid logic-based language that can express location queries over symbolic representation of space, and (ii) a model-checking-based query-processing engine that processed queries over these symbolic representations. They subsequently implemented the language in to a query-processing framework. The latter ensures that the results of any query (i) would not miss any information that satisfied its necessary and sufficient conditions and (ii) would not contain any information that did not satisfy the conditions. Model checking has proven valuable to query graph-based context models. Indeed, query language for context-aware services need some theoretical underpinnings, just like SQL is built on relational algebra.

§6 Discussion

6.1 Holistic Approach to Context Awareness

Davies and Gellersen18) discussed the technological and sociological challenges in creating the widely deployed, ubiquitous computing systems that Weiser envisioned more than fifteen years ago. They considered Weiser's scenario64) and its "foreview mirror" application, which Sal uses on her way to work, and describe how it could be implemented with *existing* technologies. The application would require a satellite navigation and information system, a video camera-based system for detecting available parking spaces, and a location-aware system for recommending nearby shops. But since these three components are currently designed, built, and deployed *independently*, they lack the actual mechanisms for seamlessly working together. In particular, they cannot integrate and process different views of the world, or contexts, in which they run.

Arguably, one needs to have a holistic approach to context-awareness.

Such an approach, however, raises several issues that go beyond mere engineering. We will not elaborate on these issues, but it is worth mentioning that besides people's concerns about their privacy, the designers of context-aware systems may have to cope with various legislations on data protection, which could eventually inhibit the deployment of their systems. Moreover, the question of cost repartition is critical for the success of context-aware systems, in cases where several companies would be providing components for *individual* services. As Davies and Gellersen[18)] point out, no effective business models have been successfully implemented thus far.

6.2 Semantic Computing

The technical issues of integrating different components for the sake of seamless and more enjoyable user experiences have been initially approached within the Web community. The *Semantic Web*, which Noy defines as "a form a Web content that will be processed by machines with ontologies as its backbone",[46)] has had a considerable influence on the context-aware research community. For example, in their review on context modeling, Strang and Linnhoff-Popien[62)] concluded that ontologies were the most flexible and effective approach. The Semantic Web stems from our sheer inability, as humans, to process the ever-growing amount of online information. We need some help from machines. The idea behind the Semantic Web is to create *independent* software agents that would share the notions they operate with. This mutual "semantic foundation" composed of ontologies would make the agents more amenable to work together. Reasoning on ontological models has proved valuable in supporting many context-aware applications. Yet, it is far from being the panacea for achieving context-awareness. Moore et al.[44)] pointed out that ontologies fails to address the issues of context processing. Roussaki et al.[53)] further elaborated on these issues.

6.3 Shortcomings with "Modeling" Context

Arguments have recently been aired that the current notion of "context" and hence context-aware computing builds on a positivist philosophical stance, where "context" is stable, delineable, and sense-able information separated from human activity. The argument is that the notion of "context" as referring to the "usage context" for a specific person using some technology cannot be separated from the human activity. "Context" then becomes firmly tied to "meaning" i.e., that context cannot be seen (and much less sensed) as an objective entity in the world, but only exists in connection to subjective meaning in an activity.[21)]

§7 Conclusion

Research on context-awareness has been conducted over a decade, but modeling and processing contextual information continues to pose some major challenges. The intrinsic complexity of representing such heterogeneous and volatile information adds to a large range of applications. Although each of the various proposals have addressed reasonable requirements for ubiquitous computing systems, it is unclear whether they can be generalized to suit all context-aware applications. Indeed there is no well-defined standard metric to actually evaluate the advantages and disadvantages of contextual models.

The systems that tend to be completely general and support a wide range of context models and applications often fail to be effective. In fact, the practical applicability and usability are important parameters to determine the quality of context modeling, and they are often inversely proportional to the generality of the model: the more expressive and powerful, the less practical and usable. Different context subproblems and applications have almost incompatible requirements, and there is no common and standard solution. That is the reason why the context models are defined on a per-application basis.

References

1) Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R. and Pinkerton, M., "Cyberguide: a mobile context-aware tour guide," *Wireless Networks*, *3*, *5* pp. 421-433, 1997.

2) Angles, R. and Gutiérrez, C., "Survey of graph database models," *ACM Computing Survey*, *40*, *1*, pp. 1-39, 2008.

3) Baldauf, M., Dustdar, S. and Rosenberg, F., "A survey on context-aware systems," *The International Journal of Ad Hoc and Ubiquitous Computing*, *2*, *4*, pp. 263-277, 2007.

4) Bardram, J. E., "The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications," in *Proc. of the 3rd International Conference on Pervasive Computing (Pervasive 2005)*, Springer, pp. 98-115, 2005.

5) Bardram, J. E., Hansen, T. R., Mogensen, M. and Søgaard, M., "Experiences from Real-World Deployment of Context-Aware Technologies in a Hospital Environment," in *Proc. of the 8th International Conference on Ubiquitous Computing (UbiComp 2006)*, Springer, pp. 369-386, 2006.

6) Becker, C. and Dürr, F., "On location models for ubiquitous computing," *Personal and Ubiquitous Computing*, *9*, *1*, pp. 20-31, 2005.

7) Biegel, G. and Cahill, V., "A Framework for Developing Mobile, Context-aware Applications," in *Proc. of the 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom 2004)*, IEEE, pp. 361-365, 2004.

8) Bolchini, C., Curino, C., Quintarelli, E., Schreiber, F. A. and Tanca, L., "A data-oriented survey of context models," *SIGMOD Record*, *36*, *4*, pp. 19-26, 2007.

9) Brown, P. J., Bovey, J. D. and Chen, X., "Context-aware Applications: from the Laboratory to the Marketplace," *IEEE Personal Communications*, *4*, *5*, pp. 58-64, 1997.

10) Cao, Y., Klamma, R., Hou, M. and Jarke, M., "Follow Me, Follow You – Spatiotemporal Community Context Modeling and Adaptation for Mobile Information Systems," in *Proc. of the 9th International Conference on Mobile Data Management (MDM 2008)*, IEEE, pp. 108-115, 2008.

11) Chaari, T., Ejigu, D., Laforest, F. and Scuturici, V.-M., "A comprehensive

approach to model and use context for adapting applications in pervasive environments," *Journal of Systems and Software*, *80*, *12*, pp. 1973-1992, 2007.

12) Chen, G. and Kotz, D., "A Survey of Context-Aware Mobile Computing Research," *Technical report*, Dartmouth College, 2000.
13) Chen, G., Li, M. and Kotz, D., "Data-centric middleware for context-aware pervasive computing," *Pervasive and Mobile Computing*, *4*, *2*, pp. 216-253, 2008.
14) Chen, H., Finin, T. and Joshi, A., "An Ontology for Context-Aware Pervasive Computing Environments," *Knowledge Engineering Review*, *18*, *3*, pp. 197-207, 2003.
15) Cheverst, K., Davies, N., Mitchell, K. and Friday, A., "Experiences of developing and deploying a context-aware tourist guide: the GUIDE project," in *Proc. of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000)*, ACM, pp. 20-31, 2000.
16) Codd, E. F., "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM*, *13*, *6*, pp. 377-387, 1970.
17) Coutaz, J., Crowley, J. L., Dobson, S. and Garlan, D., "Context is key," *Communications of the ACM*, *48*, *3*, pp. 49-53, 2005.
18) Davies, N. and Gellersen, H.-W., "Beyond prototypes: Challenges in deploying ubiquitous systems," *IEEE Pervasive Computing*, *1*, *1*, pp. 26-35, 2002.
19) Dey, A. K., "Understanding and Using Context," *Personal and Ubiquitous Computing*, *5*, *1*, pp. 4-7, 2001.
20) Dobson, S. A. and Nixon, P., "More Principled Design of Pervasive Computing Systems," in *Proc. of the 9th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'04)*, Springer, pp. 292-305, 2004.
21) Dourish, P., "What we talk about when we talk about context," *Personal and Ubiquitous Computing*, *8*, *1*, pp. 19-30, 2004.
22) Grossniklaus, M., *Context-Aware Data Management. An object-Oriented Version Model*, Verlag Dr. Müller, 2007.
23) Güting, R. H., "An Introduction to Spatial Database Systems," *The International Journal on Very Large Data Bases (VLDB)*, *3*, *4*, pp. 357-399, 1994.
24) Harter, A., Hopper, A., Steggles, P., Ward, A. and Webster, P., "The Anatomy of a Context-Aware Application," *Wireless Networks*, *8*, *2-3*, pp. 187-197, 2002.
25) Held, A., Buchholz, S. and Schill, A., "Modeling of Context Information for Pervasive Computing Applications," in *Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, 2002.
26) Henricksen, K., Indulska, J. and Rakotonirainy, A., "Modeling Context Information in Pervasive Computing Systems," in *Proc. of the 1st International Conference on Pervasive Computing (Pervasive 2002)*, Springer, pp. 167-180, 2002.
27) Henricksen, K., Indulska, J. and Rakotonirainy, A., "Using context and preferences to implement self-adapting pervasive computing applications," *Software: Practice and Experience*, *36*, *11-12*, pp. 1307-1330, 2006.
28) Hoareau, C. and Satoh, I., "Query Language for Location-Based Services: A Model Checking Approach," *IEICE Transactions on Information and Systems E91 D4*, pp. 976-985, 2008.
29) Hull, R. and King, R., "Semantic Database Modeling: Survey, Applications, and Research Issues," *ACM Computing Survey*, *19*, *3*, pp. 201-260, 1987.

30) Ilarri, S., Mena, E. and Bobed, C., "Processing Location-Dependent Queries with Location Granules," in *Proc. of the 2nd OnTheMove Workshop on Pervasive Systems (PerSys'07)*, Springer, pp. 856-865, 2007.

31) Indulska, J., McFadden, T., Kind, M. and Henricksen, K., "Scalable Location Management for Context-Aware Systems," in *Proc. of the 4th IFIP WG6.1 International Conf. on Distributed Applications and Interoperable Systems (DAIS 2003)*, Springer, pp. 224-235, 2003.

32) Indulska, J., Robinson, R., Rakotonirainy, A. and Henricksen, K., "Experiences in Using CC/PP in Context-Aware Systems," in *Proc. of the 4th International Conf. on Mobile Data Management (MDM 2003)*, Springer, pp. 247-261, 2003.

33) Jiang, C. and Steenkiste, P., "A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing," in *Proc. of the 4th International Conf. on Ubiquitous Computing (UbiComp 2002)*, Springer, pp. 307-313, 2002.

34) Kaenampornpan, M. and O'Neill, E., "An Intergrated Context Model: Bringing Activity to Context," in *Proc. of the 1st International Workshop on Advanced Context Modelling, Reasoning and Management, in coordination with the 6th International Conf. on Ubiquitous Computing (UbiComp 2004)* Springer, 2004.

35) Kang, Y.-B. and Pisan, Y., "A Survey of Major Challenges and Future Directions for Next Generation Pervasive Computing," in *Proc. of the 21th International Symposium on Computer and Information Sciences (ISCIS 2006)* Springer, pp. 755-764, 2006.

36) Katsiri, E. and Mycroft, A., "Applying Bayesian Networks to Sensor-Driven Systems," in *Proc. of the 10th IEEE International Symposium on Wearable Computers (ISWC'03)*, IEEE, pp. 149-150, 2003.

37) Kindberg, T., Barton, J. J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B. and Spasojevic, M., "People, Places, Things: Web Presence for the Real World," *MONET*, *7*, *5*, pp. 365-376, 2002.

38) Kjeldskov, J. and Skov, M. B., "Exploring context-awareness for ubiquitous computing in the healthcare domain," *Personal and Ubiquitous Computing*, *11*, *7*, pp. 549-562, 2007.

39) Korpipää P. and Mäntyjärvi, J., "An Ontology for Mobile Device Sensor-Based Context Awareness," in *Proc. of the 4th International and Interdisciplinary Conf. Modeling and Using Context (CONTEXT 2003)*, Springer, pp. 451-458, 2003.

40) Lehmann, O., Bauer, M., Becker, C. and Nicklas, D., "From Home to World - Supporting Context-aware Applications through World Models," in *Proc. of the 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom 2004)*, IEEE, pp. 297-308, 2004.

41) Levene, M. and Loizou, G., "The Nested Relation Type Model: An Application of Domain Theory to Databases," *Computer Journal*, *33*, *1*, pp. 19-30, 1990.

42) Loke, S. W., "Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective," *The Knowledge Engineering Review*, *19*, *3*, pp. 213-233, 2004.

43) McCarthy, D. R. and Dayal, U., "The Architecture Of An Active Data Base Management System," in *Proc. of the 1989 ACM SIGMOD International Conference on Management of Data*, ACM, pp. 215-224, 1989.

44) Moore, P., Hu, B. and Wan, J., “Smart-Context: A Context Ontology for Pervasive Mobile Computing,” *The Computer Journal*, pp. 1-17, first published on March 4, 2007, doi:10.1093/comjnl/bxm104.

45) Munoz, M. A., Rodríguez, M., Favela, J., Martinez-Garcia, A. I. and González, V. M., “Context-Aware Mobile Communication in Hospitals,” *Computer*, *36*, *9* pp. 38-46, 2003.

46) Noy, N., “Order from chaos,” *Queue*, *3*, *8*, pp. 42-49, 2005.

47) Pascoe, J., Ryan, N. and Morse, D., “Issues in Developing Context-Aware Computing,” in *Proc. of the 1st International Symposium on Handheld and Ubiquitous Computing (HUC’99)*, Springer, pp. 208-221, 1999.

48) Perich, F., Joshi, A., Finin, T. and Yesha, Y., “On Data Management in Pervasive Computing Environments,” *IEEE Transactions on Knowledge and Data Engineering*, *16*, *5*, pp. 621-634, 2004.

49) Peters, S. and Shrobe, H. E., “Using Semantic Networks for Knowledge Representation in an Intelligent Environment,” in *Proc. of the 1st IEEE International Conference on Pervasive Computing and Communications (Per-Com’03)*, IEEE, pp. 323-329, 2003.

50) Pils, C., Roussaki, I. and Strimpakou, M., “Location-Based Context Retrieval and Filtering,” in *Proc. of the 2nd International Workshop on Location-and Context-Awareness (LoCA 2006)*, Springer, pp. 256-273, 2006.

51) Ranganathan, A., McGrath, R. E., Campbell, R. H. and Mickunas, M. D., “Use of ontologies in a pervasive computing environment,” *The Knowledge Engineering Review*, *18*, *3*, pp. 209-220, 2003.

52) Raptis, D., Tselios, N. K. and Avouris, N. M., “Context-based design of mobile applications for museums: a survey of existing practices,” in *Proc. of the 7th Conference on Human-Computer Interaction with Mobile Devices and Services (Mobile HCI 2005)*, ACM, pp. 153-160, 2005.

53) Roussaki, I., Strimpakou, M., Kalatzis, N., Anagnostou, M. and Pils, C., “Hybrid context modeling: A location-based scheme using ontologies,” in *Proc. of the 4th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW’06)*, IEEE, pp. 2-7, 2006.

54) Salber, D., Dey, A. K. and Abowd, G. D., “The context toolkit: aiding the development of context-enabled applications,” in *Proc. of the SIGCHI conference on Human factors in computing systems (CHI’99)*, ACM, New York, pp. 434-441, 1999.

55) Satoh, I., “A Location Model for Pervasive Computing Environments,” in *Proc. of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, IEEE, pp. 215-224, 2005.

56) Satoh, I., “A location model for smart environments,” *Pervasive and Mobile Computing*, *3*, *2*, pp. 158-179, 2007.

57) Satyanarayanan, M., “Pervasive Computing: Vision and Challenges,” *IEEE Personal Communications*, *8*, *4*, pp. 10-17, 2001.

58) Schilit, B., Adams, N. and Want, R., “Context-aware computing applications,” in *Proc. of the Workshop on Mobile Computing Systems and Applications* IEEE, pp. 85-90, 1994.

59) Schmidt, A., Beigl, M. and Gellersen, H.-W., “There is more to context than location,” *Computers & Graphics*, *23*, *6*, pp. 893-901, 1999.

60) Silberschatz, A., Korth, H. F. and Sudarshan, S., "Data Models," *ACM Computing Surveys*, *28*, *1*, pp. 105-108, 1996.

61) Sohn, T., "Context-aware computing support for the educationally disadvantaged," in *Proc. of the IEEE Symposium on Human Centric Computing Languages and Environments (HCC 2003)*, IEEE, pp. 293-294, 2003.

62) Strang, T. and Linnhoff-Popien, C., "A Context Modeling Survey," in *Proc. of 1st International Workshop on Advanced Context Modelling, Reasoning and Management, in coordination with the 6th International Conference on Ubiquitous Computing (UbiComp 2004)*, Springer, 2004.

63) Strimpakou, M., Roussaki, I., Pils, C., Angermann, M., Robertson, P. and Anagnostou, M. E., "Context Modelling and Management in Ambient-Aware Pervasive Environments," in *Proc. of the 1st International Workshop on Location- and Context-Awareness (LoCA 2005)*, Springer, pp. 2-15, 2005.

64) Weiser, M., "The Computer for the Twenty-First Century," *Scientific American* pp. 94-100, 1991.

65) Widom, J. and Ceri, S., "Introduction to Active Database Systems," in *Active Database Systems: Triggers and Rules for Advanced Database Processing*, pp. 1-41, Morgan Kaufmann, 1996.

66) Ye, J., Coyle, L., Dobson, S. and Nixon, P., "Ontology-based models in pervasive computing systems," *The Knowledge Engineering Review*, *22*, *4*, pp. 315-347, 2007.

67) Zimmermann, A., Specht, M. and Lorenz, A., "Personalization and Context Management," *User Modeling and User-Adapted Interaction*, *15*, *3-4*, pp. 275-302, 2005.

Christian Hoareau: He earned his B.Sc. and M.Sc. in Computer Science from the University of La Réunion, France in 2003 and from Pierre and Marie Curie University, France in 2005, respectively. Since 2006, he has been a Ph.D. student at the Graduate School for Advanced Studies, National Institute of Informatics, Japan. His research interests include model checking, context-dependent query processing and system software for ubiquitous computing.

Ichiro Satoh, Ph.D.: He received his B.E., M.E, and Ph.D. degrees in Computer Science from Keio University, Japan in 1996. From 1996 to 1997, he was a research associate in the Department of Information Sciences, Ochanomizu University, Japan and from 1998 to 2000 was an associate professor in the same department. From 2001 to 2005, he was an associate professor in National Institute of Informatics (NII), Japan. Since 2006, he has been a professor of NII. His current research interests include, distributed and ubiquitous computing.