# DYNAMIC CONTEXT MODELING BASED FCA IN CONTEXT-AWARE MIDDLEWARE

Zhou Zhong, Xin Lin and Junzhong Gu
*Institute of Computer Applications, East China Normal University, Shanghai, China*

Keywords: Dynamic Context Modeling, Context-aware Middleware, Formal Concept Analysis.

Abstract: This paper presents a proposal that aims to process dynamic Context modeling for mobile objects in the Generalized Adaptive Context-Aware Middleware (GaCAM). Context changes closely related to difference sensors in mobile environment, so the common data modeling in which data structures were fixed after models designed is inadaptable. The principal task is: dynamic Context modeling when the mobile entities holding Context models meet each other or meet a new environment. It is important to consider the differences between dynamic Context modeling and common data modeling. The proposal is Context modeling by applying Formal Concept Analysis (FCA) merging. Our approach creates a merged specialization/generalization hierarchy which captures the knowledge of Context source in mobile environment. The hierarchy can not only describe both Context concepts and related sensors, but also can help higher Context reasoning and activating potential Context event.

## 1 INTRODUCTION

With the Internet of Things developing, A new dimension has been added to the world of information and communication technologies (ICTs): from anytime, anyplace connectivity for anyone, we will now have connectivity for anything. Accordingly, Context-aware applications are becoming more and more popular, because they can dynamically adapt to specific user and thing situations to provide smarter services and reduce the frequency of required manual inputs. In this context, we are developing a new Context-Aware Middleware infrastructure called Generalized Adaptive Context-Aware Middleware (GaCAM), to support Context-aware application developing distributed, platform-independent and self-adaptive.

What is Context? Context is defined by Abowd et al. (A.K.Dey, 2000): "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." The benefits of having Context models include being able to abstract and represent relevant contextual information used in Context-aware systems. As well, this approach enables future use of model-driven approaches and reduces implementation effort.

The one of GaCAM's issues is self-adaptive about Context source access, Context modeling, Context storage, Context reasoning, and Context query/subscription.

Among the issues, self-adaptive Context models are essential to mobile environments. Because our final users with mobile devices or clothing sensors, RFID things and intelligent agents are movable, Context models carried with them should dynamically adapt to their situations. Another reason for dynamic Context modeling is that global Context model is too large to storage in local environment and high frequency of model learning for mobile and local users will increase event process delay.

As mentioned above, dynamic Context modeling is helpful in mobile environment, and we will implement it by applying a mathematical technique of information organization, Formal Concept Analysis (FCA). FCA algorithms are machine learning techniques that enable the creation of a common structure, which may reveal some associations between elements of the two original structures. The rest of the paper is organized as follows: Section 2 presents the backgrounds. Section 3 introduces the FCA concept Analysis. Section 4 introduces dynamic Context modeling strategy, and

Zhong Z., Lin X. and Gu J..
DYNAMIC CONTEXT MODELING BASED FCA IN CONTEXT-AWARE MIDDLEWARE.
DOI: 10.5220/0003433101030110
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 103-110
ISBN: 978-989-8425-56-0

the proposal is explained by an example of what can be achieved with GaCAM. Advantages of the method are discussed in Section 5. Finally, Section 6 gives the conclusions and identifies future lines of work.

# 2 BACKGROUND

## 2.1 GaCAM

Generalized Adaptive Context-Aware Middleware (GaCAM) is a rich middleware that makes effectively Context fusion, Context modeling, Context storage and Context reasoning in running time, and provide upper application with development toolkits and service interfaces. It decouples the Context processing with high level application developing and low level development complicated physical sensor programming and common virtual sensor programming, reducing the developers' burden.

GaCAM's characteristics are: 1) Reflective. Reflection refers to the function that the system can describe internal structure themselves, can represent the own behaviour, and can dynamically reconfigure their operation modes according to the operation environment changes. The advantage of Reflective middleware relative to traditional middleware lies in loose coupling in modules or components, namely the reflective middleware can be easily by expanding and reallocation.2) Adaptive. Adaptive demands middleware can adapt according to the changes of the environment automatically, so it requires that some Context middleware components such as sensors and services information should also be Context that can be sensed and processed in the mobile and changeable environments. In this paper, we mainly discuss how to dynamically building Context model for adaptive requirement. 3) Meta-data-descript. Metadata is the data about data. By the system describing the members and method in Context middleware components by metadata, it can help to realize the reflection mechanism. 4) Agent. As a part on the system, agents are behaviour entities that stay in certain space or follow some objective Context and identify object situations and solve the problems associated with the current situation. This flexibility is a great benefit when new application agents are implemented into the system. The flexibility lies primarily in the idea that the implementation of the agent's behaviour is hidden for other parties. Agent need to interact with the environment. Agent model in our middleware is like BDI (Belief-Desire-Intention) Model, and please refer to (Rao A., 1995) for details of BDI.

Aiming at the characteristics of Context-aware middleware above, multi-Agent based system architecture is founded. GaCAM agents are divided into two categories: management Agents and function Agents. Management Agents primarily are responsible for managing function Agent, and Management transactions include life cycle, Naming, Register and Security etc. Function Agents are to realize functions Agent program, and according to different function they are mainly divided into: Sensor Agent, Context evolutionary Agent, Context reasoning Agent, Context storage /access Agent, and Service planning Agent. Figure 1 shows GaCAM architecture with Context sources and applications.

## 2.2 Context Models

In previous work, researchers have proposed many Context model modalities such as key-value, XML, object, UML-ER, Ontology and so on (T. Strang, 2004 ), and fused Contexts formally or informally.

MIcontext (N. Savio, 2007) considers various kinds of contextual elements for mobile application design. But it does not consider the associations between contextual elements. SOUPA (H. Chen, 2004) is an ontology designed to support pervasive applications, and it models intelligent agents and other relevant information. SOUPA includes user Context such as beliefs, desires, intentions, and background information. Even though SOUPA is one of the most comprehensive Context models, it does not model dynamic Context changes. CoDAMoS (D. Preuveneers, 2004) is an ontology for creating Context-aware computing infrastructures, and it models user preferences and roles. However, it does not model the dynamic interactions between these elements. The approach in this thesis focuses on three user Contexts, namely user preferences, roles, and social relationships as well as the dynamic interactions between these contextual elements. SOCAM (T. Gu, 2005) proposes a dual-layer ontology inspired by CONON, and the required Context knowledge is reduced to the upper ontology, and the domain-specific onto logy can be dynamically bound. An MDE approach (C. TACONET, 2010) is proposed to define context-aware application models by UML meta-models. The advantage is that models may be applied for different platforms and technologies especially different context management technologies. But the work focuses on how to use context view Meta

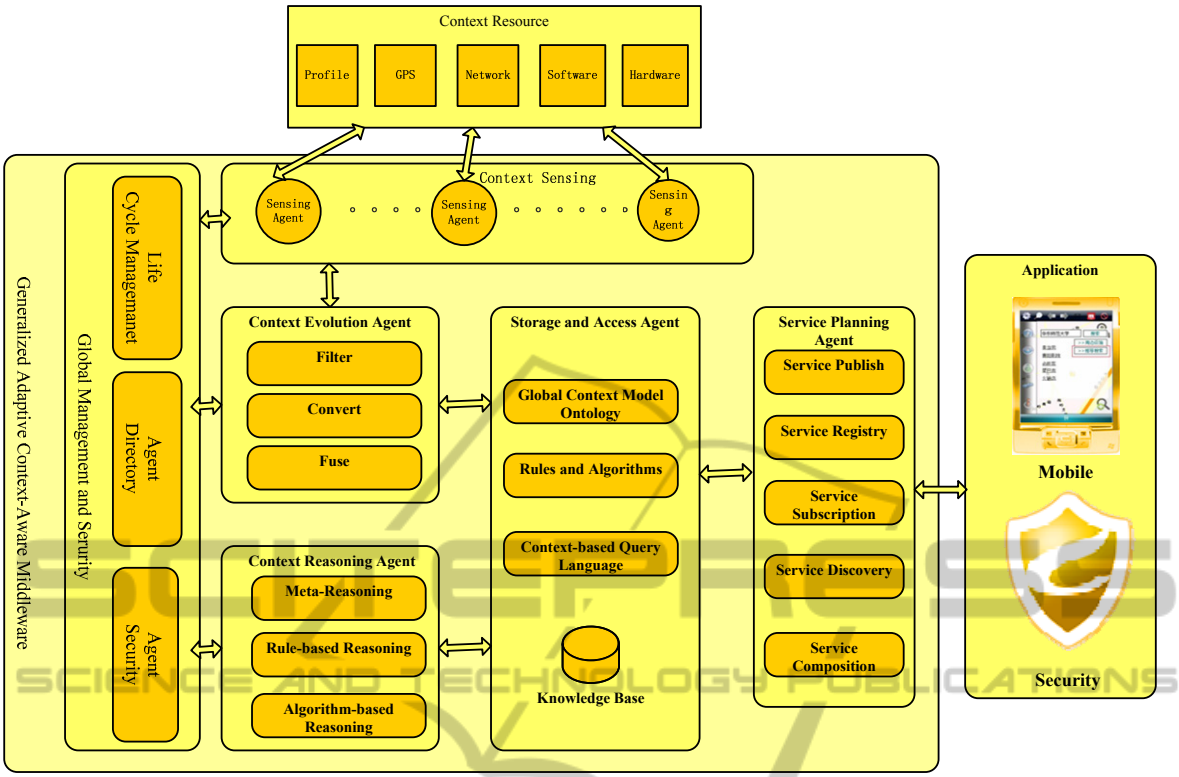


Figure 1: GaCAM architecture.

models in the development phrase for global design.

All of these works give us very integrity global Context models or global Context model designs which refer to their Context system and domain ontology is designed beforehand. However, how to process the dynamic modeling when Context relates with specific situations is not mentioned clearly. In the next section, we will discuss our mathematics basis of our dynamic Context modeling.

## 3 INTRODUCING OF FCA

Formal Concept Analysis (FCA) was introduced by (R. Wille, 1982) and is completely developed in (B. Ganter, 1999). FCA is the process of abstracting conceptual descriptions from a set of objects described by attributes (R. Wille, 1982). The FCA has been used in works related to symbolic data analysis and knowledge representation (R.Godin, 1995). Olivier Cur´e in (Curé Olivier, 2009) extends existing FCA-based systems for ontology merging. In our modeling, we also use FCA-based Merging for dynamically Context modeling, but we consider Context-aware characteristics so the merging method is easier than the FCA merging method that mainly processes texts in(Curé Olivier, 2009) , and we find that FCA merging enables the creation of a common lattice structure, which may reveal some associations between elements of the two original structures and the Context model using the lattice is also helpful to retrieving, so it is very appropriate for local Context modeling. We shall begin by introducing the basic notions defined by Wille.

**Definition.** A formal context $K$ is defined as a triple of sets, (*G, M, I*), where G is a set of objects, M is a set of attributes, and $I$ is a binary relation between G and M (i.e. $I \subseteq G \times M$ ). $(g,\ m) \in I$ is read “object g has attribute m”.

**Definition.** For $A \subseteq G$, we define $A' := \{m \in M \mid \forall g \in A : (g,m) \in I\}$ and, for $B \subseteq M$ , we define $B'$ $:= \{g \in G \mid \forall m \in B : (g,m) \in I\}$ .

A formal concept of a formal context $(G,M,I)$ is defined as a pair $(A,B)$ with $A \subseteq G, B \subseteq M$ , $A' = B$ and $B' = A$. The sets $A$ and $B$ are called the *extent* and the *intent* of the formal concept $(A,B)$. On $K$ a partial order relation $\leq$ can be defined through the following formula where $(A,B),(A',\ B') \in K$: $(A,B) \leq (A',B') \Leftrightarrow A \subseteq A' (\Leftrightarrow B \supseteq B')$ . This relation is a generalization/specialization hierarchy

relationship. The set of all formal concepts of context $K$ with the partial order $\leq$ is always a complete lattice, call the concept lattice (or Galois lattice).

A possible confusion might arise from the double use of the word 'context' in FCA and in Context model. This comes from the fact that FCA and context model are two models for the concept of 'context' which arose independently. In order to distinguish both notions, we will always refer to the FCA context as 'formal context'. The Contexts in Context model are referred to just as 'Context'. There is no direct counter-part of formal concepts in Context model. Context classes /Context objects are best compared to FCA objects, and sensors / sensor Types are compared to FCA attributes.

# 4 DYNAMIC CONTEXT MODELING

## 4.1 Prerequisite

As mentioned above, relative agents will follow moving objects or stay in local environments. Sensor agents are responsible for transforming the raw data collected from real sensors into an identifiable Context classes/objects. Dynamic Context modeling happens when moving objects come into the new environment. Different from common modeling, it is important to consider sensor information into modeling, because Contexts and sensors are changing in mobile environment. Moreover, like previous works, it is necessary for global consistency to design a global Context model. Therefore, before introducing the modeling process, we firstly discuss sensor agents and global Context Model.

Sensor agent in our middleware named Virtual Sensor. Virtual sensors abstract similar sensor type from logic sensors and physical sensors. Each virtual sensor consists of two elements: Self Description Profile and Output Format template (Output). For example, Virtual Sensor about location includes GPS sensor, WIFI sensor, etc. Dynamic Context modeling process not accesses the real sensors directly, but gets information from virtual sensors. In order to simplify the expression, virtual sensor is referred to as 'sensor' in the following. Table 1 shows the example of Virtual Sensors designed in our middleware.

Table 1: Example of Virtual Sensors.

| Sensor Type | Example |
|---|---|
| Location | GPS, Wireless positioning device (WIFI),FID (from GIS Map) |
| Acoustic | Sound detector, Microphone |
| Thermal | Thermometers |
| Computational Environment | Device detector, Network monitor |
| Profile | RFID(item profile), User profile, Society Relation, Group profile |
| Log | System logger, User behaviour logger |
| Document | Search engine, Spider |
| Time | Time sensor, Date sensor, Zone sensor |
| Event Record | Outlook, Anti-virus monitor, Firewall |

Global Context model is also essential to tell what can be known. The global Context models of many Context-Aware Systems are quite similar because of the similar domain researched such as smart space and location based application etc. After some work of summary, the fundamental element classification framework in middleware categorizes the elements in the Context models into two main categories: Physical Context and Logic Context. Physical Context includes kinds of Context from physical sensors, and Logic Context includes kinds of Context from Logic sensors. Its sub Contexts are Social Relation Context, Activity Context, Computational Context and Information Context. Social Relation Context represents the people and their information changed with little frequency. Activity Context represents activities that people and things involved in. Computational Context represents software and hardware information from virtual sensors. Figure 2 summarizes the fundamental element classification framework. The global Context model is described as a full ontology with Context relations (relations not mentioned in this paper).

Context environments and mobile objects have their own knowledge about Contexts that are wanted to be shared for enriching Context each other. When dynamically modeling happens, the merging is necessary.

Meanwhile, Context environment changes when new certain Context or new certain sensor comes in. Therefore, we should consider remodeling the Context model to integrate new Context and estimate new sensor ability.

For the above prerequisite, we define the modeling process as follows:

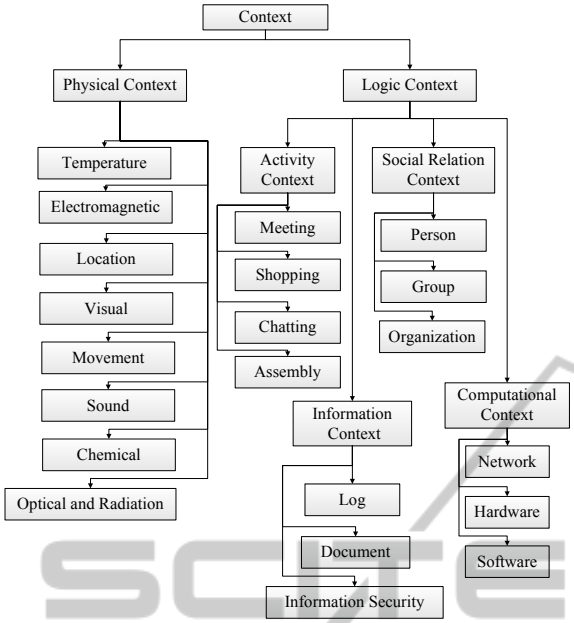


Figure 2: Global Context model classifications.

## 4.2 Initial

Firstly, the space range of the local environment or mobile objects is defined as Context set and sensor set available, where G= Context set, M=sensor set, formal context K= space. I can be illustrated with an example about a local environment in Table 2. In the example, the G includes three Context classes (A: Computer, B: Room, C: Person) and four objects (A1, A2, B1, B2) in which (A1, A2) are class A's objects and (B1, B2) are class B's objects. The M includes three types (α: RFID sensor, β: Location sensor, γ: Profile sensor) and four sensors (α1, α2, β1, β2) in which (α1, α2)'s type is α and (β1, β2)'s type is β. Because the modeling process is not involved Context name and sensor name, letters are instead of full names in order to simplify the expression.

Context class C and sensor type γ have no instances, it means that the space knows what are C and γ, but instances of C and γ don't appear currently. The situations of this kind are always widespread, when the local environment has known the class of certain Context of which the objects have not appeared and the type of certain sensor of which the sensors have not appeared. In the Table 2, if Context class has objects, then we will not list the class to avoid the lattice nodes too large, but it is no problem for presenting sensor types because the types presented can be helpful to abstraction of Context classes.

Table 2: The example of the local environment.

| | α | α1 | α2 | β | β1 | β2 | γ |
|---|---|---|---|---|---|---|---|
| A1 | ✓ | ✓ | | | | | ✓ |
| A2 | ✓ | | ✓ | | | | ✓ |
| B1 | ✓ | ✓ | | ✓ | ✓ | | |
| B2 | ✓ | ✓ | | ✓ | | ✓ | |
| C | | | | ✓ | | | ✓ |

Secondly, we can get Galois connection lattice from this table using FCA. The Galois connection lattice is showed by Figure 3. The lattice is a specialization/generalization hierarchy that presents the Context model of Contexts and relative sensors in the example, and all nodes are almost meaningful because of using sensor as the scale of attributes.

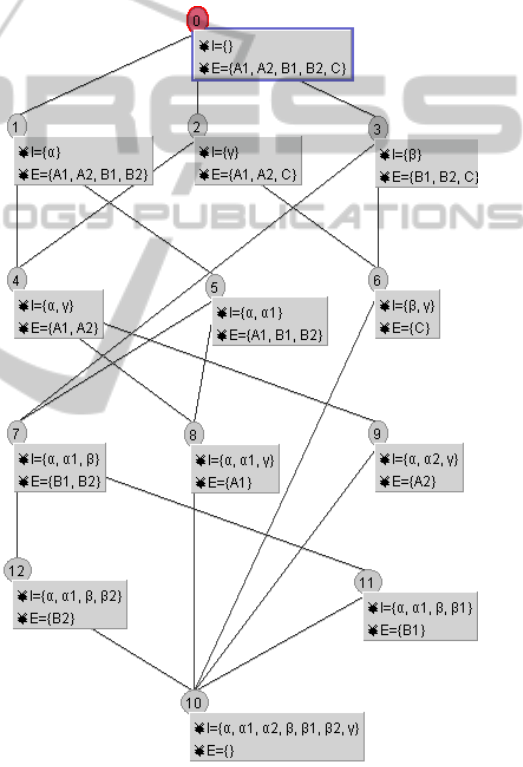


Figure 3: The Galois connection lattice of the example.

The reason why we use lattices as the formal presentation of the dynamic Context model is that only aiming at the class description of global model is not appropriate for the dynamic environment, where individual-relations between objects and sensors are also very important. The lattice can also contain and present these individual relations.

## 4.3 Merging

The merging schematic is showed by Figure 4. To complete the example above for explaining merging

process, we assume that a moving object with its sensors also carry Contexts showed by Table 3. Its space can be set as the formal context K' (G', M', I'). G' includes three Context classes (C: Person, E: Device) and four objects (C1, C2, E1, E2) in which (C1, C2) are class C's objects and (E1, E2) are class E's objects. M' includes three types (α: RFID sensor, β: Location sensor, γ: Profile sensor) and four sensors (γ1, β3, β4) in which (β3, β4)'s type is β and γ1's type is γ. When the moving object meets local environment, contexts and sensor agents will be computed by management agent in the local environment. Contexts and sensors may be complementary, so extra Context information can be got because of adaptive agent characteristic of sensors. In the example, (A1, A2) get the γ1 attribute and (E1, E2) get the α2 attribute. Table 4 is the whole table for Context modeling, and then we can get a merging model K'' dynamically from FCA, the result Lattice is showed by Figure 5.

## 5 ADVANTAGES OF THIS METHOD

In addition to solve dynamic Context modeling in mobile environment, the method has the following advantages:

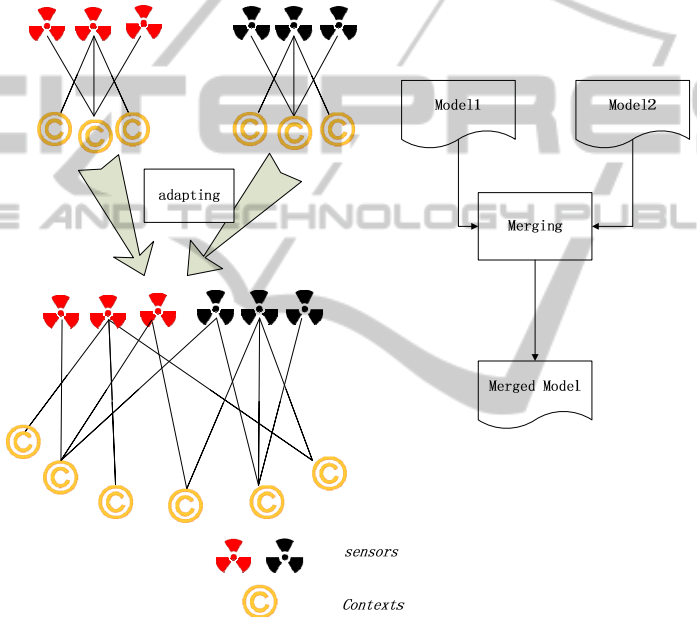


Figure 4: The merging schematic.

Table 3: Contexts of another mobile object.

| | β | β3 | β4 | γ | γ1 | α |
|---|---|---|---|---|---|---|
| C1 | ✓ | ✓ | | ✓ | ✓ | |
| C2 | ✓ | | ✓ | ✓ | ✓ | |
| E1 | ✓ | ✓ | | | | ✓ |
| E2 | ✓ | | ✓ | | | ✓ |

Table 4: Whole model after modeling.

| | α | α1 | α2 | β | β1 | β2 | β3 | β4 | γ | γ1 |
|---|---|---|---|---|---|---|---|---|---|---|
| A1 | ✓ | ✓ | | | | | | | ✓ | ☑ |
| A2 | ✓ | | ✓ | | | | | | ✓ | ☑ |
| B1 | ✓ | ✓ | | ✓ | ✓ | | | | | |
| B2 | ✓ | ✓ | | ✓ | | ✓ | | | | |
| C1 | | | | ✓ | | | ✓ | | ✓ | ✓ |
| C2 | | | | ✓ | | | | ✓ | ✓ | ✓ |
| E1 | ✓ | | ☑ | ✓ | | | ✓ | | | |
| E2 | ✓ | | ☑ | ✓ | | | | ✓ | | |

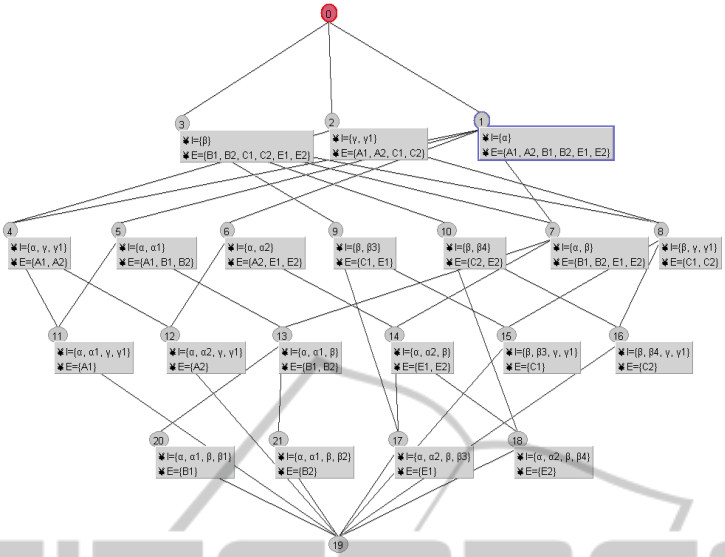


Figure 5: The result lattice of the whole model after merging.

## 5.1 Decouple between Class and Attribute

Context classes and Context classes' attributes are decoupling. The method is different from normal FCA modeling in which FCA's attributes are simple attribute names. Dynamically, the Context objects of the same class are discrepantly described from discrepant ability sensors of the same class, and also could be different because of information security and environment interference factor. So when Context objects move, it is well designed that Context classes and Context classes' attributes are decouple. Context objects/classes should have composite description from output template of different sensors/sensor types' dynamically. Using sensors/sensor types as the scale of attributes in modeling is also avoiding such meaningless abstraction that the results still need optimizing after merging. For example, the two classes, book and person, which both have an attribute named "name", are generalized into an abstract class. Obviously, this case is meaningless.

From the lattice, we can find it easy to check replaceable sensors of Contexts and easy to change more powerful sensor when some sensor does not work.

## 5.2 Assistance in Reasoning

The lattice can be helpful to get higher level relation and activate semantic event. While Context class relations are predefined, individual relations are dynamically ascertained by sensors. Like human beings' five senses, the Context-aware middleware can use sensors to ascertain relations. For example, using location sensors to ascertain the distance of two objects is similar to using eyes of human being. Therefore, relation reasoning is dependent on sensors to some extent. In our modeling, sensors' types and objects are attributes of FCA. Contexts which have the same sensor type may have relations dependent on the sensor type. Comprehensive sensing by human beings' five senses with consciousness can tell human beings higher level relation or semantic event, such as having a meeting or listening to the lecture. Like profile sensors, certain sensors can realize similar functions of consciousness about society and culture knowledge. For example, we can know an object with its location is a person by combination of location sensor and profile sensor. To some extent, the more different sensor attributes an object has in FCA, the concreter it is. Meanwhile, it may have higher level relations and involved events. So to one FCA context which has determinate sensor types, the objects of it may have relations with each other. We can ascertain current relations by retrieving the lattice and getting the sensors output.

For example, the node 3 from Figure 5 has only one attribute, β: Location sensor type. We assume that it can ascertain two relation {spatial part of, near} derived from β: Location sensor type. Its objects {B1: Room1, B2: Room2, C1: Person1, C2: Person2, E1:Device1, E2:Device2} may have these relations to be ascertained. We not only want to know spatial relations, but also higher relations than spatial relations. Similarly, we assume that it can

ascertain the higher level event that Person using computer in derived from {β: Location sensor} combined respectively with {α: RFID sensor, γ: Profile sensor}. The node 7 has the attributes {α: RFID sensor, β: Location sensor} and the node 8 has the attributes {β: Location sensor, γ: Profile sensor, γ1}, so the objects of the two nodes are helpful for reasoning. Further, reasoning agent can traverse perceptual nodes from top to bottom and check whether these objects have relative sensors sensing and get the values of them to ascertain current interrelations of objects.

MES (H. Chaker, 2010) also tries to assist a user in his business tasks requiring information. MES takes contextual factors (user, environment, business tasks) into account. Compared with MES, our solution takes sensor factor extra and every mobile or sensing body can be an abstract user. Also, we prefer discovering to tasking when coping with context, and tasks are carried by BDI-style Agent. MES's reasoning is based on task process. Our reasoning is based on sensing ability. Therefore, our reasoning is appropriate to recommendation.

# 6 CONCLUSIONS AND FUTURE WORK

Our study in this paper shows that our dynamic Context modeling in the Context-aware middleware is feasible and necessary for providing Context-aware applications. We've implemented a Context modeling agent as an infrastructure to support Context-aware applications, decoupling and dynamically modeling between sensors and Contexts. The work of this paper is part of our ongoing research project – Generalized Adaptive Context-Aware Middleware (GaCAM). Because FCA is not appropriate for large data, our next step is finding the threshold value of Context quantity to define the scale of local storage space. Meanwhile, it is helpful to explore novel approaches to both improve the assist ability in reasoning and reduce the time cost.

# ACKNOWLEDGEMENTS

This work is partly supported by Science and Technology Commission of Shanghai Municipality (STCSM) project (No. 09510703000 and No. 08511500303).

# REFERENCES

A. K. Dey, G. D. Abowd, 2000. "Towards a better understanding of context and context-awareness," In *Proceedings of CHI 2000 Workshop on the what, Who, Where, and how of Context-Awareness, The Hague*, *The Netherlands*.

T. Strang, C. Linnhoff-Popien, 2004. A Context Modeling Survey. In *the Sixth International Conference on Ubiquitous Computing*, *Nottingham/England*.

N. Savio, J. Braiterman, 2007. "Design sketch: The context of mobile interaction," In *Proceedings of International Journal of Mobile Marketing*, *Singapore*.

H. Chen, F. Perich, T. Finin and A. Joshi, 2004. "SOUPA: Standard ontology for ubiquitous and pervasive applications," In *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, *Boston, MA*.

D. Preuveneers et K. De Bosschere, 2004. "Towards an extensible context ontology for ambient intelligence," In *Proceedings of the 2nd European Symposium on Ambient Intelligence*, *Eindhoven, The Netherlands*.

T. Gu, H. K. Pung and D. Q, 2005. "A service-oriented middleware for building context-aware services," *Journal of Network and Computer Applications, vol. 28, pp. 1-18.*

A. Rao, M. Georgeff, 1995. *BDI Agents from Theory to Practice, Technical Note 56, AAII.*

R. Wille, 1982. "Restructuring lattice theory: An approach based on hierarchies of concepts". In *Ordered Sets, pages 225-470. Reidel, Dordrecht-Boston*.

B. Ganter and R.Wille 1999. "Formal concept analysis: mathematical foundations". Springer.

R. Godin, G.Mineau, R.Missaoui and H.Mili, 1995. "Méthodes declasification conceptuelle basées sur les treillis de galois et applications". *Revued'Intelligence Artificielle, 9(2):105-137*.

C. Olivier, 2009. "Merging Expressive Ontologies Using Formal Concept Analysis" In *OTM 2009 Workshops, Lecture Notes in Computer Science, Volume 5872*.

C. Taconet, Z. Kazi-Aoul, 2010. "Building context-awareness models for mobile applications" *Journal of digital information management (JDIM), vol. 8, pp. 78-87*.

H. Chaker, M. Chevalier, C. Soulé-Dupuy, A. Tricot, 2010. "Improving information retrieval by modelling business context". *International Workshop on User Profiles in Multi-application Environments (MultiA-Pro 2010)*, *IEEExplore digital library, pp. 121-126.*