PAPER

A Two-Stage Composition Method for Danger-Aware Services Based on Context Similarity

Junbo WANG[†a)], ***Nonmember***, **Zixue CHENG**[††], ***Member***, **Lei JING**[††], **Kaoru OTA**[†], ***and*** **Mizuo KANSEN**[†††], ***Nonmembers***

SUMMARY Context-aware systems detect user's physical and social contexts based on sensor networks, and provide services that adapt to the user accordingly. Representing, detecting, and managing the contexts are important issues in context-aware systems. Composition of contexts is a useful method for these works, since it can detect a context by automatically composing small pieces of information to discover service. Danger-aware services are a kind of context-aware services which need description of relations between a user and his/her surrounding objects and between users. However when applying the existing composition methods to danger-aware services, they show the following shortcomings that (1) they have not provided an explicit method for representing composition of multi-user' contexts, (2) there is no flexible reasoning mechanism based on similarity of contexts, so that they can just provide services exactly following the predefined context reasoning rules. Therefore, in this paper, we propose a two-stage composition method based on context similarity to solve the above problems. The first stage is composition of the useful information to represent the context for a single user. The second stage is composition of multi-users' contexts to provide services by considering the relation of users. Finally the danger degree of the detected context is computed by using context similarity between the detected context and the predefined context. Context is dynamically represented based on two-stage composition rules and a Situation theory based Ontology, which combines the advantages of Ontology and Situation theory. We implement the system in an indoor ubiquitous environment, and evaluate the system through two experiments with the support of subjects. The experiment results show the method is effective, and the accuracy of danger detection is acceptable to a danger-aware system.

key words: *ubiquitous computing, context-aware services, danger-aware, composition of contexts, context similarity*

1. Introduction

Ubiquitous computing technologies are making our life more and more convenient. One of the hot topics in ubiquitous computing is context-aware services, which provide users personalized services, while giving consideration to the users' location, current time, available devices or facilities around the user, and preferences of the users, etc.

Unlike traditional systems, whose actions are only triggered by the user's input, a context-aware computing system can perceive the user's current context, and automatically provide services based on predefined reasoning rules. For example, users can define context reasoning rules for sleeping situation or cooking situation to assist their services in [4]. Context Studio is a personalization tool for defining context reasoning rules by end users [24]. In this paper, the reasoning rule consists of two parts, i.e., a predefined context and the corresponding services on the context.

Furthermore, accidents often happen when the specified people are close to some specified objects. For example, when a boy is playing, he may hurt himself by the surrounding dangerous objects, e.g., knife, electric socket, etc. Therefore, it is very important to detect the user's dangerous context quickly, and provide services automatically according to the context, which is called danger-aware service.

A danger-aware system needs to consider the relations among users and objects. For example, relations between a user and the surrounding objects, e.g., a child is close to a dangerous object. Or relations between the contexts of two users, e.g., a child is playing with a ball and his mother is close to the child with hot soup, or a person holding a particular flower is close to another person who has allergy to the flower.

For providing context/danger aware services, it is very important to define, represent, detect and manage contexts. One approach to define and represent context is based on context models, which can be categorized into 6 types by Thomas Strang [1]. Ontology Based Model [2], [4], [26] and [29] is testified as an appropriate method to represent the context, since some obvious advantages of ontology, such as interoperability of different devices, easily for sharing knowledge, etc. However, the shortcoming of those ontology-based context models is too complex to let end users design their services easily. Moreover, context model cannot always meet users, since the users' environment is always changing.

Composition of contexts is a promising approach to ease the definition, detection of contexts and provision of services. Various contexts can be automatically composed of small pieces of contexts, and services can be discovered based on the composition.

Kameas et al. proposed a very flexible composition mechanism, which see each object in the everyday life as a composeable smart object [5]. Smart objects can compose a context together by using abstract plugs, which is designed based on GAS Ontology [6], [28]. However, it did not con-

Manuscript received October 14, 2009.
Manuscript revised January 18, 2010.

[†]The authors are with Graduate School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu-shi, 965–8580 Japan.
[††]The authors are with School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu-shi, 965–8580 Japan.
[†††]The author is with University-Business Innovation Center, University of Aizu, Aizu-Wakamatsu-shi, 965–8580 Japan.
a) E-mail: d8101202@u-aizu.ac.jp

DOI: 10.1587/transinf.E93.D.1521

sider/represent relations between a user and its surrounding object explicitly and clearly.

Thomas et al. proposed a CoCo concept (Context Composition) [7], which can compose the information of context to simplify the development and deployment of context-aware services. However, this method is limited in composing information to represent context for one user. Multi-users' contexts are not be considered to discover services.

Analytic Hierarchy Process based context-aware service composition method were proposed by Koumoto et al. in [8], which compose context-aware services and select the more appropriate service based on user preference. However, it may bring large cognitive load for user to make such comparison when the items of comparison increase or change.

Moreover, the above researches have a common problem, that their reasoning mechanisms are not flexible, since they just can provide services exactly following the predefined context reasoning rules.

In other words, developers/users should predefine lots of reasoning rules for each possible context detected from sensors to get services, even though the possible contexts are very similar. Without a lot of predefined contexts reasoning rules, services may not be provided. They have intrinsic lack of flexibility for satisfying dynamically changing of the environment. Case based reasoning can be used to solve the problem. However, it also incurs a big trouble when managing a mass of cases [10].

In this paper, a two-stage composition method by using context similarity is proposed for danger aware system.

Firstly, we consider user context as either a simple one or a complex one. Simple one consists of one user and related objects, which is called single-user context in the paper. Complex one is composed by more than one related single-user contexts for providing services by considering the relations of contexts, which is called multi-user context.

Secondly, two-stage composition rules are presented. The first stage is composition of the useful information to represent the context for a single user. The second stage is composition of single user's contexts together to provide services by consider the influence among single user's contexts.

Finally the danger degree of the detected context is computed by using context similarity between the detected context and a predefined context. Danger degree means the level of possible risk. Developers/users assign danger degree to each predefined danger context/situation, and danger-aware system computes the danger degree of a detected context/situation based on the context similarity between them.

J. Yang et al. also proposed a concept similarity based context-aware service system in [9]. However, it has not provided a composition method and the method they used to get the concept similarity is too complex and costs too large time to realize in a danger-aware system.

In the proposed method, context is represented based on two-stage composition rules and a Situation Theory based Ontology, which combining the advantages of Ontology and Situation theory.

J. Barwise proposed Situation theory in [11], by which situations can be described with 3 primitives, i.e., individual, relations between individuals, and space-time information. However, the situation theory is mainly for linguistic and natural languages processing research [12], [13], and has not considered how to detect those descriptions with current ubiquitous technologies.

A hybrid context model based on Situation Theory and Ontology has been proposed in [25]. However, it just uses the concepts in a normal ontology to represent Situation Theory-based context. The above advantages of Situation Theory have not been imported into Ontology to improve normal Ontology when describing context.

To this end, a two-stage composition method based on context similarity is proposed. By using our methods, (1) system can automatically compose contexts by the small pieces of information, (2) provide more thoughtful services by considering the relation of contexts, and (3) compute danger degree of the detected context by calculating context similarity with the predefined context.

The rest of the paper is organized as follows. Section 2 gives example and model of the two stage composition method. Section 3 presents the design of the method in detail. Section 4 gives a danger-aware system based on the method. The evaluation results are represented in Sect. 5. Finally Sect. 6 concludes the paper and gives directions for future works.

2. Model and Example

2.1 An Example of Danger-Aware System

Accidents often happen when a specified people close to some specified things out sight of guardians' supervision. For example, as shown in the left side of Fig. 1, when a child, named John, is close to a knife, he/she may hurt himself. Therefore when designing danger-aware services, we need to basically consider who the user is, what the objects surrounding user are, what the relation between user and objects is and when/where it happen.

Moreover, some accidents may happen with the influ-

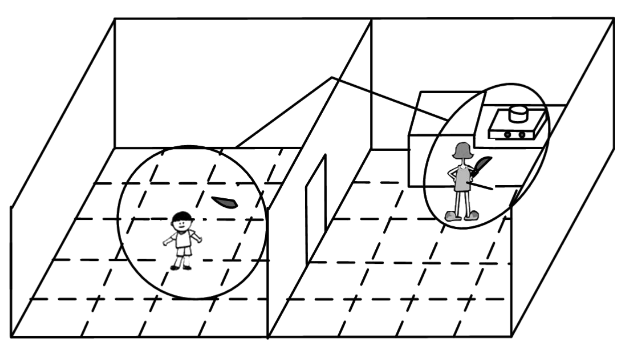


Fig. 1 An example of danger-aware system.

ence of others' contexts. For example, as shown in the right side of Fig. 1, his mother is cooking with a scoop in her hand in the right room. Service simply just based on the left side may incur additional accidents, since the mother may go to look to her child in a hurry with the scoop and/or without close the gas. Therefore, when providing the services, we need to not only consider one user's context, but also need comprehensive consider the relation with others' contexts.

Meanwhile, it is very hard for developers/users to specify all the possible dangerous contexts/situations. For example in the left room, John is close to a knife is a dangerous context/situation. However, Tom, i.e., another child, is close to a scissor is also a dangerous situation. For these similar dangerous contexts/situations, the developers/users need to define various times repeatedly, since the system may not work if the detected context/situation is not exactly equal to the specified contexts/situations.

2.2 Model

Figure 2 shows a model representing contexts based on two-stage composition, which is abstracted from the example in Fig. 1. It consists of composition operators and all the individuals in the Fig. 1. The operator represents the first stage composition, and represents the second stage composition. The composition operators consist of functions/rules for composing contexts and computing their danger degree based on context similarity, which will be presented in Sect. 3 in detail. The operator in the left side composes a single user context in the left room as shown in Fig. 1, and the operator in the right side composes another single user context in the right room together. We consider the environment as a special individual, which always be included in the first stage of composition. If the context represented after the first stage composition is recognized as dangerous situation, system provides warning services automatically. And the system also provides more thoughtful services by the comprehensive consideration of the relations between two contexts, i.e., child's context and mother's context. For example, the system alert mother not forgets to turn off the gas and hold scoop in a hurry.

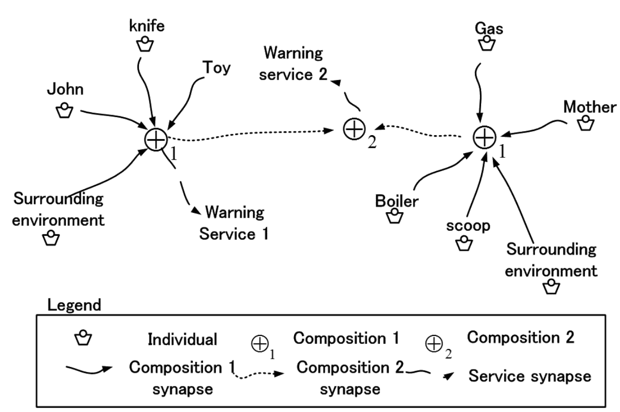


Fig. 2 Abstract model of composition.

In this paper, we assume there is no conflict between different users' contexts in danger-aware system. However, it doesn't mean these research problems/points aren't important in this research field. A research paper about how to solve conflict problem has been published in [27] by our group.

3. Our Methods

3.1 Basic Idea

Generally, for context-aware system firstly users/developers need to specify contexts/situations and bind services with them. The system compares the detected context/situation with specified ones to decide whether and what services should be provided. However, it brings a big burden to the users/developers for specifying all the possible contexts/situations. Moreover, it may not work if the detected context/situation is not exactly equal to the specified contexts/situations.

To ease the development and enhance the spreading use of context-aware applications, we propose a two-stage composition method based on context similarity. Firstly, developers/users specify the possible dangerous contexts/situations, and assign danger degree to each possible danger context/situation. And then the system automatically detects the information and composes a context as shown in Fig. 3, by matching the data in registration DB to get the corresponding class in ontology for each individual. And then the system compares the detected context with predefined possible dangerous context/situation to get the danger degree of the detected context based on similarity between the two contexts. The context similarity is gotten based on the ontology designed in Sect. 3.2. Finally the system provides services based on danger degree of the detected context.

By using our methods, (1) system can automatically compose the contexts, (2) provide more thoughtful services by considering the relation of contexts, and (3) deal with dangerous contexts which have not been specified by using context similarity.

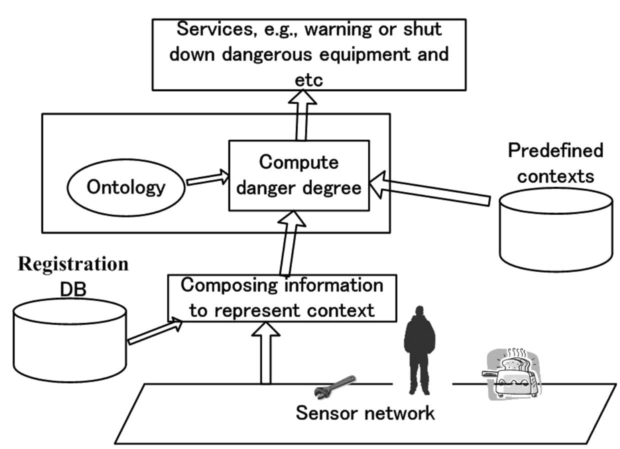


Fig. 3 Basic image of the method.

3.2 STB-Ontology (Situation Theory Based Ontology)

The situation theory is proposed by J. Barwise, for mainly discussing formally the situations represented by human languages. It is mostly used for natural language processing, linguistic issues, informational, and cognitive phenomena [12], [13].

The advantages of representation of context based on Situation Theory are,
(1) The variety of contexts can be represented by using only 3 primitives, i.e., individual (and its properties), relations between individuals, and space-time information. Therefore, we can have a uniform way for representation of context.
(2) The situation theory based representation can describe the context close to the real situations happen around us, since it is a kind of predicate logic and has the same representation power as nature languages.
(3) For composition of contexts, the most important thing is how to represent small pieces of information (simple and primitive contexts), and how to detect and compose contexts by those pieces, and provide services. By employing situation theory in composition of contexts, the context can be described as primitive as possible, and very close to the situations happen around us.

However, representing context just using Situation Theory also has some shortcomings. For example, it just can define context at instance level, but cannot define at an abstract level (e.g., class level). Thus developers/users have to define a lot of contexts/situations, even though many of them are very similar, which brings big load to developers/users.

On another hand, ontology has many advantages in representing contexts, e.g., formal to express the knowledge, and easy for knowledge sharing and reusing. Furthermore, by using the concepts of class, sub-class, and instance in ontology, the context information can be represented not only in instance level but also in abstract level (class level). These advantages can complement the representation based on situation theory. Furthermore, similarity of contexts can be defined on ontology, so the amount of specifying the contexts and corresponding services can be reduced.

Therefore, we combine the advantages of ontology and Situation Theory to build a STB-Ontology (Situation Theory Based Ontology) for describing contexts. STB-Ontology includes all the properties and advantages of normal ontology. Additionally, it borrows the structure/concepts in Situation Theory, i.e., individuals consisting of persons and objects, relations, and space-time locations, and creates the corresponding class and sub-classes for describing them in detail. STB-Ontology can be used to represent context, coupled with the help of various definitions in Sect. 3.3, defined based on the basic structure of Situation Theory.

Figure 4 shows the general features/concepts to represent context based on STB-Ontology. A context includes individuals, and relations, where individual class has subclasses, i.e., person, object and environment. Relation class

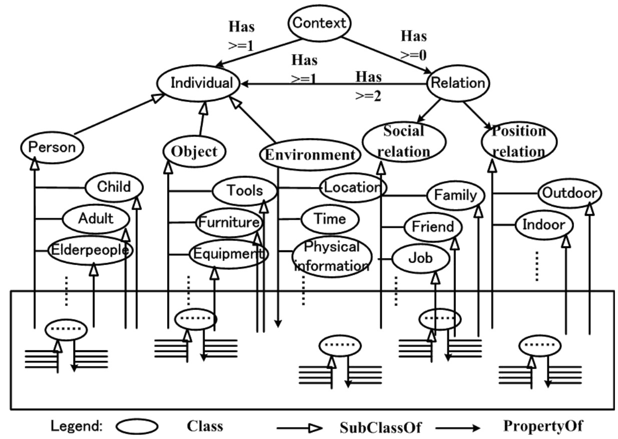


Fig. 4 Main concepts and structure of STB-Ontology.

consists of position relation and social relation, where position relations are detected by sensors, and social relations are gotten with matching individuals IDs with the knowledge in DB. Other relations can be extended easily. An instance of environment consists of the instances from the class Location, Time, Temperature and etc. Class Person, Object, Social relation and Position relation have various subclasses to represent each domain in detail. For the physical information in the environment individual, we just consider temperature and brightness as examples to show the design method. It can be extended easily for other parameter/concept.

3.3 Definitions

Firstly, we give some definitions to represent contexts, and based on which to compute danger degree.

Definition 1: Individual, Relation and Environment
I, P, O, E, and R are sets containing the instances of the class *Individual*, *Person*, *Object*, *Environment* and *Relation* in the STB-Ontology, respectively. $I = P \bigcup O \bigcup E$ and we use i to denote an individual.

We use $rel(i_1, i_2)$ to denote a relation between two individuals. It is an element of R, which can be position relation or social relation. $rel^p(i_1, i_2)$ is used to represent the position relation between two individuals and $rel^s(i_1, i_2)$ is used to represent the social relation between two individuals. For example, position relation can be *inSameRoom*, *inAdjoiningRoom*, *isCloseTo*, and so on. Social relation can be *isFatherOf*. $i_1, i_2 \in P \bigcup O$.

We use $REL(i_1, i_2) = <rel^p(i_1, i_2),\ rel^s(i_1, i_2)>$ to denote a relation pair.

We use e to denote an environment individual, $e \in E$. Based on the STB-Ontology, e consists of three elements, i.e., *time*, *location* and *physical information*. We describe e as follows,

$$e = <t, l, pinfo> \text{ where}$$

t represents time information, which is an instance of *time*

class,
l represent location information, which is an instance of *location* class,
pinfo = <$value^{temp}$, $value^{bri}$> represents physical information, e.g., temperature, brightness. $value^{temp}$, $value^{bri}$ are the values respectively. For example, <27, 1750> means the temperature is 27 centigrade, and brightness is 1750 lm. The units of temperature and brightness can be designed by applications. For simplicity, we only consider the above two physical parameters. Others can be represented similarly.

Definition 2: Single-user context
We call a context *single-user context*, if and only if it contains a person, all the related surrounding objects and environment factors. We use *suc* to denote a single-user context, which is a 4-tuple and can be described as follows,

$suc = <p, O, REL, e>$ where,

p represents the person, and $p \in P$.
$O = \{o_1, o_2, \ldots, o_k, \ldots, o_m\}$ is a finite set including all the related objects with p. The related objects can be decided by the distance between person and an object for satisfying the needs of different applications.
$REL \subseteq \{ REL(p, o_1), REL(p, o_2), \ldots REL(p, o_k), \ldots REL(p, o_m)\}$ is a finite set including the relations between p with each object in O.
e is the surrounding environment of p.

Example 1:
$suc_{011} = <p, O, REL, e>$
$p = John$
$O = \{knife, toy\}$
$REL = \{<isCloseTo(John, knife), null>,$
$<isCloseTo(John, toy), null>\}$
$e = <afternoon, living\ room, <27, 1050>>$

Definition 3: Multi-user Context
We call a context *multi-user context*, if it contains more than one user, all the related surrounding objects and environment factors of the users. *Multi-user context* is for providing services with consideration of relations among more than one user's context. For the simplicity of discussion, in this paper we mainly consider the situation just including two users, i.e., two *single-user context*s, in each *multi-user context*. It can be extended to more than two users.

We use *muc* to denote a *multi-user context*, which is a 3-tuple and can be described formally as follows,

$muc = <P, REL(suc_1, suc_2), S\_UC>$ where,

$P = \{p_1, p_2\}$ is a set of persons.
$REL(suc_1, suc_2) = <rel^p(p_1, p_2), rel^s(p_1, p_2)>$ is a 2-tuple which represents the relations of two single-user contexts. In this paper, we use the position and social relation between two persons belonging in P to represent the relation of two contexts.
$S\_UC = \{suc_1, suc_2\}$ is a set of *single-user context*s.

Definition 4: Predefined Danger Context
Predefined danger context is specified by developers/experts/users based on their experiences and knowledge. It consists of predefined single-user contexts and predefined multi-user context.

Predefined single-user context *pre-suc* has the same structure with single-user context in the Definition 2, where we add a prefix *pre-* before each element to show the difference with the ones in single-user context.

$pre-suc = <pre-p, pre-O, pre-REL, pre-e>.$

We use *pre-muc* to denote a predefined *multi-user context* which has the same structure with the *multi-user context* in the Definition 3, where we add a mark *pre-* before each element to show the difference of them.

$pre-muc = <pre-p, pre-REL(pre-suc_1, pre-suc_2), pre-S\_UC>.$

Definition 5:Danger degree
We use δ to denote the danger degree, which shows how much a situation is dangerous.

$\delta(suc)$ and $\delta(muc)$ are used to represent the danger degree of a *single-user context* and a *multi-user context*.

$\delta(pre\text{-}suc)$ and $\delta(pre\text{-}muc)$ are used to represent the danger degree of a *predefined single-user context* and a *predefined multi-user context*.

All of them are decimal fractions larger than or equal to 0 and less than or equal to 1. 1 means the most dangerous contexts/situations, and 0 means the least dangerous contexts/situations.

3.4 Computing Danger Degree Based on Context Similarity

In this section, we will discuss how to compute the danger degree of a context based on context similarity. Firstly, developers/users give danger degrees for some specified contexts/situations, e.g., the danger degree of the context "*Tom is close to a scissor*" is *0.85*. However, the detected context may not be exactly the predefined danger context, e.g., "*John is close to a knife*," where John is Tom's elder brother. It is very hard for developers/users to define every possible danger context. However, by using our method, the above detected context also can be recognized as a danger context automatically by comparing similarity with the predefined danger context.

3.4.1 Computing Danger Degree of Single-User Context

Firstly, we get the danger degree of single-user context *suc* by using the formula (1).

$$\delta(suc) = \begin{cases} \delta(pre-suc) & Sim_{suc}(suc, pre-suc) \geq Thre_{suc} \\ 0 & Sim_{suc}(suc, pre-suc) < Thre_{suc} \end{cases} \tag{1}$$

where $Thre_{suc}$ represents a threshold of similarity of two *single-user context*s, We define $Sim_{suc}(suc,pre\text{-}suc)$ is a formula to get the similarity of *two single-user context*s.

Based on the Definition 2 and 4, a *single-user context* and a *predefined single-user context* can be described as follows,

$suc = <p, O, REL, e>$ and $pre - suc = <pre - p, pre - O, pre - REL, pre - e>$,

And then we can get the similarity of them based on the following formula,

$$\begin{aligned} Sim_{suc}(suc, pre - suc) = w_{11} \cdot Sim_{sema}(p, pre - p) \\ + w_{12} \cdot Sim_{O\_REL} \\ + w_{13} \cdot Sim_e(e, pre - e) \quad (2) \end{aligned}$$

where w_{11}, w_{12}, w_{13} are weights of each items ($\sum_{i=1}^{3} w_{1i} = 1$).

In the formula (2) Sim_{O_REL} is a value which means the similarity of objects and relations between the two contexts.

$$\begin{aligned} & Sim_{O\_REL} \\ & = \sum_{i=1}^{m} (w_{3i} \cdot Sim_i) \\ & = \sum_{i=1}^{m} w_{3i} \cdot (w_{21} \cdot Sim_{sema}(o_i, pre-o_i) \\ & \quad + w_{22} \cdot Sim_{sema}(REL(p, o_i), REL(p, pre-o_i))) \quad (3) \end{aligned}$$

where w_{21}, w_{22} and w_{3i} are weights, and $w_{21} + w_{22} = 1$, $\sum_{i=1}^{m} w_{3i} = 1$.

$Sim_e(e, pre - e)$ in formula (2) is for getting the similarity of environment shown as follows.

$$\begin{aligned} & Sim_e(e, pre - e) \\ & = w'_1 \cdot Sim_{sema}(t, pre - t) \\ & \quad + w'_2 \cdot Sim_{sema}(l, pre - l) \\ & \quad + w'_3 \cdot Sim_{value}(value^{temp}, pre-value^{temp}) \\ & \quad w'_4 \cdot Sim_{value}(value^{bri}, pre-value^{bri}) \quad (4) \end{aligned}$$

where w_1', w_2', w_3', w_4' are weights of each items in a e ($\sum_{i=1}^{4} w'_i = 1$), and $Sim_{value}(a, b)$ is for getting the similarity to two values, i.e., temperature value and brightness value.

$$Sim_{value}(a, b) = \begin{cases} 1 & |a - b| \geq Threshold \\ 0 & |a - b| < Threshold \end{cases} \quad (5)$$

Threshold can be decided based on the information need to be compared.

$Sim_{sema}(x, y)$ in the formula (2) and (4) is to get the semantic similarity of two concepts/classes in ontology.

We use the method presented in [14], [15] to compute similarity of two concepts in an ontology, i.e., formula (6).

$$Sim_{sema}(x, y) = 1 - \partial(x, y) \quad (6)$$

Where $\partial(x,y)$ denotes the weighted distance of two concepts x and y in ontology, and we can get it by using formula (7),

$$\partial(x, y) = [w(p(x, y)) - w(x)] + [w(p(x, y)) - w(y)] \quad (7)$$

$p(x, y)$ represents the most close common parent of x and y, and $w(x)$ is a weight value of concept x, which can be calculated by using formula (8).

$$w(n) = \frac{1}{k^{l(n)+1}} \quad (8)$$

Where $l(n)$ is the length of the path from root to concept n, and k is a predefined parameter which is larger than 1 to indicate the decrease rate (currently be set to 2) [14].

There is a problem when get the similarity of objects and relations in the two contexts in formula (3), i.e., the number of objects/relations in the two contexts may not exactly same. In this paper, we use *null* elements as objects/relations to fill the context which has fewer objects, so that the two contexts can have the same number of objects/relations.

For distinguishing null elements in the detected context and the predefined context, we use $pre - null$ to represent the null elements in the predefined context. And we designed two special formulas to get the semantic similarity when there is $null/pre - null$ as follows.

When some options have not been defined by developers/users, we also use $pre - null$ to represent it. The following two formulas are for all the elements in context.

$$Sim_{sema}(z, pre - null) = 1 \quad (9)$$

It means the similarity between any object/relation z in the detected single-user context including *null*, and $pre - null$ in predefined single-user context equal to 1. In other words, if the number of objects in the detected context is larger than predefined one, the redundant objects in the detected contexts will not impact of the reasoning result. For example, if a user predefine the situation if child A is close to the object B and C will be dangerous. Suppose the system detects a situation that child A is close to the object B, C and D. We set the $Sim_{sema}(D, pre - null)$ equal to 1to not produce an effect on reasoning the detected context as dangerous.

$$Sim_{sema}(null, z) = 0 \quad (10)$$

It means the similarity between *null* in the detected single-user context, and any object/relation z in the predefined single-user context not including $pre - null$ equal to 0. In other words, if the number of objects in the detected context is less than the predefined context, we set $Sim_{sema}(null, z)$ to decrease the total context similarity. For example, if the user also predefine the situation if child A is close to the object B and C will be dangerous. Suppose the system detects a situation that child A is close to the object B. We set the $Sim_{sema}(null, z)$ equal to 0 to decrease the possibility of reasoning the detected context as dangerous.

When computing Sim_i in the formula (3), we should consider the comparing order, i.e., we should compare which object in the detected context and which object in the predefined context to get the similarity. In this paper, we follow a principle based on the maximum semantic similarity. In other words, for the first object in the detected context, we find the object, not equal to $pre - null$, which has

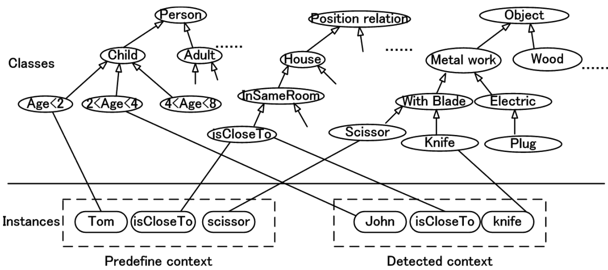


Fig. 5 An example of ontology.

a maximum semantic similarity with the first object in the detected context, and for the next object we find the object, not equal to $pre - null$ if there are the resting objects in the predefined context, which has a maximum semantic similarity with the second object in the detected context, and then compare them in this way until the last one.

Example 2:

Figure 5 shows an example of ontology. For simplify, we suppose there is just one object around the user.

Suppose the predefined dangerous context is

$$pre - suc = <Tom, \{scissor\}, \{<isCloseTo(Tom, scissor), null>\}, <afternoon, living\ room, <pre - null, pre - null>>>$$

where Tom is an instance of class *child(age<2)*, scissor is an instance of class *Scissor*, and *isCloseTo(Tom,scissor)* is an instance of class *isCloseTo* as shown in Fig. 5.

We assign the danger degree of this predefined context as 0.85.

Suppose the system detects a *single-user context suc*,

$$suc = <John, \{knife\}, \{< isCloseTo(John, knife), null>\}, <afternoon, livingroom, <27, 1050>>>$$

where John is an instance of class *child(2<age<4)*, knife is an instance of class *Knife*, and *isCloseTo(John, knife)* is an instance of class *isCloseTo* as shown in Fig. 5.

Firstly, we get the weights of the above related classes. *w(x)* is used to represent the weight of a class *x* in ontology. Based on formula (8), we can get,

$$w(Child) = 0.25$$
$$w(Child(age < 2)) = 0.125$$
$$w(Child(2 < age < 4)) = 0.125$$
$$w(isCloseTo) = 0.0625$$
$$w(WithBlade) = 0.125$$
$$w(Knife) = 0.0625$$
$$w(Scissor) = 0.0625$$

And the based on the formula (6), we can get,

$$Sim_{sema}(Child(age < 2), Child(2 < age < 4)) = 0.75$$
$$Sim_{sema}(isCloseTo, isCloseTo) = 1$$
$$Sim_{sema}(Knife, Scissor) = 0.875$$

Based on the formula (9), i.e., $Sim_{sema}(z, pre-null) = 1$, we can get $Sim_e(e, pre - e) = 1$.

Then by using the formula (2)

$$\begin{aligned}
&Sim_{suc}(suc, pre - suc) = \\
&w_{11} \cdot Sim_{sema}(Child(age < 2), Child(2 < age < 4))+ \\
&w_{12} \cdot (w_{21} \cdot Sim_{sema}(Knife, Scissor)+ \\
&w_{22} \cdot Sim_{sema}(isCloseTo, isCloseTo))+ \\
&w_{13} \cdot Sim_e(e, pre - e) \\
&= 0.4 \cdot 0.75 + 0.3 \cdot (0.5 \cdot 0.875 + 0.5 \cdot 1) + 0.3 \cdot 1 \approx 0.88
\end{aligned}$$

Where we suppose $w_{11} = 0.4$, $w_{12} = w_{13} = 0.3$, $w_{21} = w_{22} = 0.5$, and $Sim_{thre} = 0.8$.

Then $Sim_{suc}(suc, pre - suc) > Sim_{thre}$ Therefore we can get $\delta(suc) = \delta(pre - suc) = 0.85$ based on formula (1).

3.4.2 Computing Danger Degree of Multi-User Context

Formula(11) is used to get the danger degree of a *multi-user context*.

$$\delta(muc) = \begin{cases} \delta(pre - muc) & Sim_{muc}(muc, pre-muc) > Thre_{muc} \\ 0 & otherwise \end{cases} \tag{11}$$

which means when the context similarity between the detected multi-user context and the predefined multi-user context is larger than a threshold $Thre_{muc}$, the danger degree of the detected context is assigned same with the predefined context.

$$\begin{aligned}
&Sim_{muc}(muc, pre-muc) \\
&\quad = w_{41} \cdot Sim_{suc}(suc_1, pre-suc_1) \\
&\qquad + w_{42} \cdot Sim_{suc}(suc_2, pre - suc_2) \\
&\qquad + w_{43} \cdot Sim_{rel}(REL(suc_1, suc_2), \\
&\qquad REL(pre - suc_1, pre - suc_2))
\end{aligned} \tag{12}$$

where w_{41}, w_{42}, and w_{43} are weights, and the sum of them are 1.

$Sim_{suc}(suc_1, pre - suc_1)$ and $Sim_{suc}(suc_2, pre - suc_2)$ can be gotten based on the formula (2) in the Sect. 3.4.1.

$Sim_{rel}(REL(suc_1, suc_2), REL(pre - suc_1, pre - suc_2))$ is to get the similarity of two set of relations. We use $Sim_{rel}(suc_1, suc_2)$ to simplify $Sim_{rel}(REL(suc_1, suc_2), REL(pre - suc_1, pre - suc_2))$ in formula (13), then

$$\begin{aligned}
&Sim_{rel}(suc_1, suc_2) \\
&\quad = w_1 Sim_{sema}(rel^p(suc_1, suc_2), \\
&\qquad rel^p(pre - suc_1, pre - suc_2)) + \\
&\qquad w_2 Sim_{sema}(rel^s(suc_1, suc_2), \\
&\qquad rel^s(pre - suc_1, pre - suc_2))
\end{aligned} \tag{13}$$

Where w_1 and w_2 are weights of each item ($w_1 + w_2 = 1$) defined by developers, depending on applications. $Sim_{sema}(x,y)$ is the same with the formula (6), to get the similarity of two classes/concepts in an ontology.

Example 3: Suppose there is a predefined multi-user context which is specified by the developers/users,

$pre-muc = <pre-P, pre-REL(pre-suc_1, pre-suc_2), pre-$

$S\_UC>$.

Suppose $pre-suc_1$ is the same with *pre-suc* in example 3.

And $pre-suc_2 = <Mother, \{gas\},$
$\{<isCloseTo(Mother, gas), null>\},$
$<afternoon, kitchen, <30, 1550>>>$
and $d(pre-muc) = 0.85$.

Suppose the system detects a multi-user context muc shown as follows,

$muc = <P, REL(suc_1, suc_2), S\_UC>$.

Suppose suc_1 is the same with the detected single-user context in example 2, i.e., $suc_1 = suc$, and suc_2 is the same with $pre-suc_2$ in predefined multi-user context.

And then $sim_{suc}(suc_1, pre-suc_1) = 0.88$ by using the result of example 2, and $sim_{suc}(suc_2, pre-suc_2) = 1$ since they are same.

If we suppose $sim_{rel}(REL(suc_1, suc_2), REL(pre-suc_1, pre-suc_2)) = 1$, and $Thre_{muc} = 0.9$ in formula (11), w_{41}, w_{42}, and w_{43} are equal to 0.4, 0.4, and 0.2 respectively.

$Sim_{muc}(muc, pre-muc) = 0.4\times0.88+0.4\times1+0.2\times1 = 0.95$

And then we can get $\delta(muc) = d(pre-muc) = 0.85$ by using formula (11) since 0.95 is larger than 0.9.

3.4.3 Selecting Appropriate Predefined Context

In this paper, we look for predefined single-user context based on priority. *Person* in the context has the highest priority, and then is surrounding objects and relation between the person and objects based on the formula (3), and finally environment factor has the lowest priority.

In other words, firstly we search the predefined contexts which include the persons whose similarity with the user in detected context is not less than a specified threshold, i.e., $Thres_{per}$. Secondly in these contexts we search the ones which have the objects, whose similarity with the objects in detected context is not less than a specified threshold, i.e., $Thres_{obj}$. And then in these contexts we search the ones which have the relations, whose similarity with the relations in detected context is not less than a specified threshold, i.e., $Thres_{rel}$. Finally, we select one from them which have the maximum context similarity with the detected context.

For selecting appropriate multi-user context, we also follow the priority. The two single-user contexts have the highest priority, and then is relation of two single-user contexts. The system searches the predefined multi-user contexts by comparing the similarities of the first single-user context, the second single-user context and the relation of contexts with the corresponding thresholds respectively. Those similarities are computed by comparing a detected one and a predefined one respectively.

3.5 Composition Rules

Composition Rule 1: Composition of single-user context

Composition rule 1 is the first stage of composition for composing a user and his/her surrounding objects to represent single-user context. We use com_1 to denote the composition rule 1, which is a procedure shown as follows.

Environment factor always exists in a single-user context. Therefore, for each user, the single-user context always exists, even though there is no surrounding object.

Firstly, the system detects the position relation between user p and all the objects detected by the system in O^{all}, and stores objects which are close to the person p in O. And then for each object o_k in O, the system gets the position and social relation between o_k and p, and represents them with $rel^p(p, o_k)$ and $rel^s(p, o_k)$ respectively. Finally system constructs *suc* and gets the danger degree of *suc*, which is consist of danger degrees of single-user contexts using the function $get\_suc\_dd(suc)$.

There are some functions and variables employed in the procedure com_1. Function $detect\_p\_rel(p, o)$ is for detecting whether the relation between person p object o is belongs to the class/subclasses *isCloseTo*. It returns a Boolean value. Functions $get\_p\_rel(p, o)$ and $get\_s\_rel(p, o)$ are for getting the position relation and social relation between person p and object o respectively. They return the corresponding relations. If there are no relations, they return *null*. These three functions are different based on the different sensor networks, and we will implement these two functions in Sect. 4 based on our sensor networks.

The sets variables O, REL, all initially equal to empty, are used to store the surrounding related objects and relations between person and objects in O. The double variable $\delta(suc)$, initially equal to 0, is used to save the danger degree of single-user context *suc*. Boolean variable rel_close, initially equal to false, is used to get the return value of function $detect\_p\_rel(p, o)$ for judging whether the relation between p and o is belongs to the class/subclasses of *isCloseTo*.

Function $get\_suc\_dd(suc)$ is for getting the danger degree of *suc*. Firstly, the system finds the appropriate predefined danger context *presuc* based on the method described in Sect. 3.4.3. And then, the system calculates the context similarity $Sim_{suc}(suc, pre-suc)$ between detected context *pc* and predefined danger context *prec* using the formula (2) in the Sect. 3.4.1. Finally, we get the danger degree of the detected context by using the formula (1) in the Sect. 3.4.1.

Procedure com_1:
Input: p, e, surrounding objects $O^{all} = \{o_1, o_2, \ldots, o_i, \ldots, o_u\}$
Output: $\delta(suc)$
Variables:
$O := \phi; REL := \phi; rel\_close := false; \delta(suc) := 0;$
Begin
    For $\forall o_i \in O^{all}$
        $rel\_close := detect\_p\_rel(p, o_i)$
        if $rel\_close = true$ ***then***
            $O := O \bigcup \{o_i\}$
    For $\forall o_k \in O$
        $rel^p(p, o_k) := get\_p\_rel(p, o_k)$
        $rel^s(p, o_k) := get\_s\_rel(p, o_k)$
        $REL(p, o_k) := \{rel^p(p, o_k), rel^s(p, o_k)\}$
    $suc := < p, O, REL, e >$

$\delta(suc) := get\_suc\_dd(suc)$
Return $\delta(suc)$
End

Composition Rule 2: Composition of multi-user context based on single-user contexts
Composition rule 2 is the second stage of composition for composing single-user contexts to represent multi-user context by considering the relations/influences among contexts. For the simplicity of discussion, in this paper we mainly consider the composition of two single-user contexts, and it can be extended to more than two.

Services often are necessary by considering multi-contexts and the relations of contexts. For example, we suppose there are two contexts/situations and each of them is safe if they happen separately. However if they happen with some specified relations, e.g., close to each other, the whole context may be dangerous. For example, a mother is sewing with needle and his daughter is playing around. Or a child is playing and his mother is close to the child with hot soup. Or a person holding a particular flower is close to another person who has allergy to the flower. Or the situations in the Fig. 1, when the situation in the left room happens, the situation in the right room may change to dangerous.

Therefore, we propose the composition rule 2 to consider multi contexts and relations of contexts. We use com_2 to denote the composition rule 2. Firstly, the system gets the relation between two contexts by using the function $get\_context\_rel(suc_1, suc_2)$. In this paper, we use the position and social relation between two persons in the two contexts to represent the relations of two contexts. And then, the system gets the dangerous degree of multi-user context based on the dangerous degree of each multi-user context by using the function $get\_muc\_dd(muc)$.

There are several variables in the procedure com_2. Sets P, S_UC, $REL(suc_1, suc_2)$, all initially equal to empty, are used to store persons, single-user contexts, and relation of two contexts, respectively. Double variable $\delta(muc)$, initially equal to 0, is used to save the danger degree of the detected *muc*.

Procedure com_2:
Input: $suc_1 =< p_1, O_1, REL_1, e_1 >$,
$suc_2 =< p_2, O_2, REL_2, e_2 >$
Output: $\delta(muc)$
Variables:
$P := \phi; S\_UC := \phi; REL(suc_1, suc_2) := \phi; \delta(muc) := 0;$
Begin
$P := \{p_1\} \bigcup \{p_2\}$
$S\_UC := \{suc_1\} \bigcup \{suc_2\}$
$REL(suc_1, suc_2) := get\_context\_rel(suc_1, suc_2)$
$muc :=< P, REL(suc_1, suc_2), S\_UC >$
$\delta(muc) = get\_muc\_dd(muc)$
Return $\delta(muc)$
End

The function $get\_muc\_dd(muc)$ is designed based on Sect. 3.4.2. Firstly, the system compares suc_1 with $pre - suc_1$, suc_2 with $pre - suc_2$, and $REL(suc_1, suc_2)$ with $REL(pre - suc_1, pre - suc_2)$, respectively by using formula (2) and (10). If all of these similarities are larger than the corresponding thresholds, the system assigns the danger degree of *pre-suc* to the detected *muc*. Otherwise, danger degree of the detected *muc* is assigned as 0.

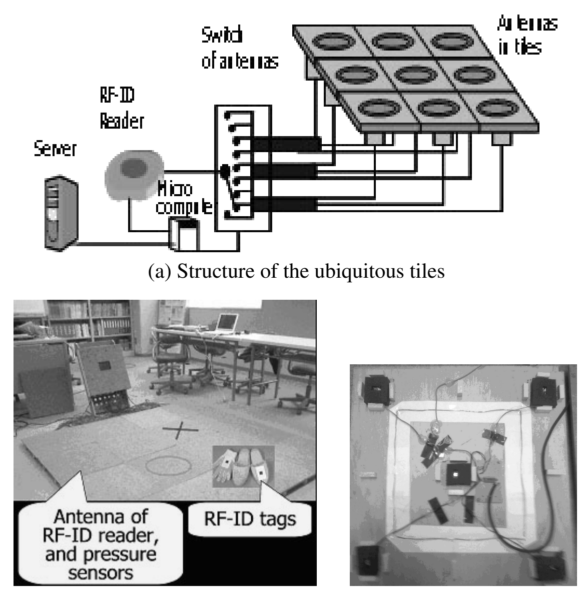
4. A Danger-Aware System Based on the Proposed Method

In this Section, we present a danger-aware system based on the two-stage composition method. Firstly, we present the hardware environment of the system, i.e., a U-tile sensor network. And then we present the software design, mainly including how to use ontology to define and detect a context, and how to compute context similarity.

4.1 Hardware Environment

This danger-aware system is built based on a U-tiles sensor network [3], [16]. It can detect the situations, by detecting RF-ID tags and their positions precisely in a room, by using a DB and reasoning techniques.

The ubiquitous tiles consist of some tiles on a floor, an RF-ID reader, and a switch of antennas controlled by a PIC program as shown in Fig. 6 (a). Figure 6 (b) shows the whole image of U-tiles sensor network. An antenna of the RF-ID reader is embedded on the back of every tile, as shown in Fig. 6 (c). The reader is periodically or selectively connected with the antenna of each tile, through the switch controlled by a Micro-computer program. When a person stands (an object is put) on the tile, the RF-ID tag on the person (or the object) will be read by the reader. Therefore, who (or what) is on which tile (where) can be detected. By searching a DB with the detected ID as a key, the person's properties can be achieved. If there are two persons (individuals) on different



(a) Structure of the ubiquitous tiles
(b) Surface side (c) Back side of each tile
Fig. 6 Ubiquitous tiles.

tiles, both IDs can be detected, and the distance between the two persons can be computed and the relation between the two persons can be retrieved from a DB with the two IDs as keys.

A computer server as shown in Fig. 6 (a) coordinates work of the whole system, including assisting users define a context, describing the information from the sensor network as contexts, finally computing danger degree and providing services. The detail software design in the computer server will be presented in Sect. 4.2.

We assume there is a smart room, and the floor and the horizontal surface of furniture are paved using U-tiles. Based on the above discussion, we can get the position and the relation of objects putted on the floor or the horizontal surface of furniture.

4.2 Software Design

4.2.1 Develop Environment

In this subsection, we mainly discuss the software design in the computer server, which accomplishes the works of (1) providing interfaces to assist users in defining contexts, (2) representing the information of contexts based on the definitions and composition rules in Sects. 3.3 and 3.5, and (3) computing danger degree of detected contexts based on ontology discussed in Sect. 3.4.

The following develop tools are used in the system,

- Develop environment: Eclipse,
- Develop language: Java SDK 1.6,
- External JARs: Jena-2.6.0, mysql-connector-java-5.0.6-bin.jar, RXTXcomm.jar,
- Ontology design software: Protégé3.1,
- Database: MySQL 5.0.

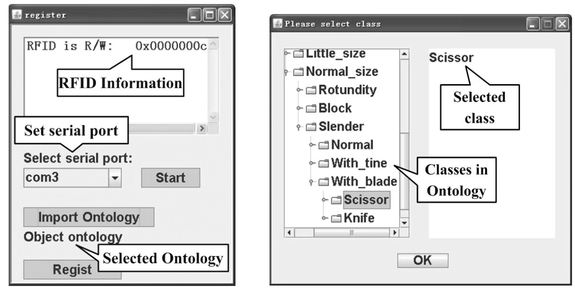
The software system is developed based on Java and eclipse. Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS, OWL, and SPARQL, and includes a rule-based inference engine. Jena is open source (See [18] for detail). In this paper, we mainly use APIs for ontology to compute the similarity between two concepts.

For designing ontologies, we use the software of Protégé3.1. It is a free, open-source platform that provides a suite of tools to construct domain models and knowledge-based applications with ontologies (See [19] for detail). Developers/users also can download the existing ontologies designed by others from internet.

4.2.2 User Interfaces

The system mainly includes four interfaces, i.e., registration interface, single-user context definition interface, multi-user context definition interface, and U-tile interface.

Figure 7 shows the registration interface. The interface in Fig. 7 (a) is used to set serial port, show RF-ID tag information, import ontology, and save information. Figure 7 (b) is a pop-up interface for users to select a class from



(a) Set up interface (b) Ontology select interface

Fig. 7 Registration interface.

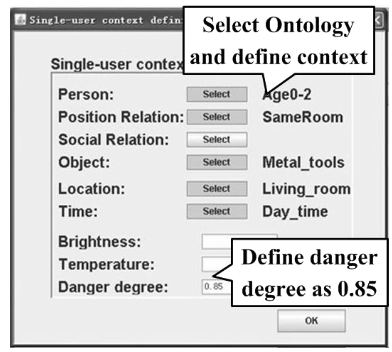


Fig. 8 Single-user context definition interface.

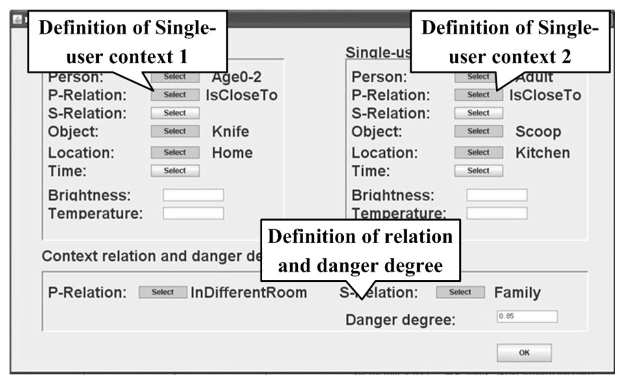


Fig. 9 Multi-user context definition interface.

the corresponding ontology. The left part of Fig. 7 (b) shows class information in the selected ontology, and the right part shows the selected class.

Figure 8 shows the interface for defining single-user contexts. When the user pushes one of select buttons, the system provides the interface for the user to select class similarly to Fig. 7 (b). At the right of select buttons, it shows the selected class by users. Through the edit boxes in the Fig. 8, users can input the value of brightness, temperature, and danger degree, respectively. If the user selects/inputs nothing for some items, e.g. social relation and brightness, the system sets them as null.

Figure 9 shows the interface for defining multi-user context. The left part is for defining single-user context 1 and the right part is for defining single-user context 2. Both

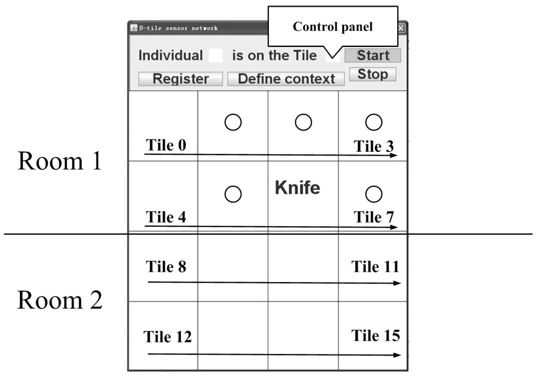


Fig. 10 Sensor network interface.

of them are similar to Fig. 8. The lower part is for defining the relation of two contexts, where two select buttons are linked with a pop up interface similarly to the Fig. 7 (b) for selecting classes in ontology. The right of two buttons shows the selected classes.

Figure 10 is the U-tile interface, which follows the appearance of U-tile sensor network, and shows the position information of objects/persons on the tiles. In each raw from left to right, the tile ID increases in order. When there are objects on tiles, the names and positions of objects/persons will be shown automatically. For example in Fig. 10 there is a knife on the tile 6.

4.2.3 Implementation of Functions

In this sub-Section, we implement the functions to get the position and social relation in the two composition rules, and the method for computing context similarity.
We implement these functions based on U-tiles sensor network and information stored in DB. However, there are also many ways to implement these functions. For example, we can implement them in the RD-based indoor positioning system, such as SpotON [20] and RADAR [21] or in the ultrasonic based indoor positioning system such as Active Bat [22] and Cricket [17]. Our method composition can be used as a building block in construction of context-aware services in these positioning systems. Here, we use U-tiles, an indoor positioning sensor network in order to show the feasibility of implementing our method.
(1) The function to get the position relation of two individuals.

We implement the function to get position relation of two individuals, i.e., $get\_p\_rel(i_1, i_2)$, in the two composition rules as follows. The inputs of the function $get\_p\_rel(i_1, i_2)$ are tile IDs of the two individuals represented by using integer variables. The outputs are position relations represented by string variable. In this system, we mainly get three kinds of position relation, i.e., *isCloseTo*, *isInSameRoom*, *isInDifferentRoom*. Others can be added in the future in the similar way.

We use Fig. 10 to show the implementation of function $get\_p\_rel(i_1, i_2)$. We assume the whole U-tiles are divided into two parts. The upper part, i.e., tile ID is from 0 to 7, is room 1 (Kitchen), and the lower part, i.e., tile ID is from 8 to 15, is room 2 (Living room). Firstly, the system judges whether they are in the same room based on tile ID. If they are in the same room, the system continues to judge whether they are close to each other. The surrounding tiles direct around a tile, including the tiles on the diagonal, are considered as the relation of *isCloseTo*. As shown in the Fig. 10, objects on the tile 1 to tile 3, tile 5 and tile 7, marked using circle, are recognized as *closeto* the knife on the tile 6 and then it returns relation *isCloseTo*, otherwise it returns relation *isInSameRoom*. If they are recognized in the different room, it returns relation *isInDifferentRoom*.
(2) The function to get the social relation of two individuals.

In this system, we detect the social relation between two individuals using the function $get\_s\_rel(i_1, i_2)$. The input of this function are tag IDs of two individuals represented by two integer variables. The output is a social relation of two individuals represented by a string variable. Firstly, the system gets class information of two individuals from a registration table in the DB. Then the system searches in a *relation table* to get relation information based on two classes and returns the corresponding relation.
(3) The function to get the semantic similarity of two concepts/classes in ontology.

Finally the system computes danger degree using functions for getting similarity of two concepts/classes in the ontology. It is named $get\_c\_similarity(c1, c2)$. The inputs of function are two concepts/classes in the ontology represented by two string variable, i.e., $c1$ and $c2$. Output is the similarity of two concepts/classes represented by a double varable, i.e., *sim*, initially equal to *zero*.

Firstly, the system gets all the classes in the ontology by using the function $getAllClasses()$, which is designed based on APIs in the Jena. The classes are saved in the String set $CLASS$, which is initially equal to *empty*. And then it gets common parent classes of two concepts, i.e., $c1$, $c2$, and saves them in the String set C_CLASS, initially equal to *empty*. After that it finds the common parent which is most close to two concepts based on the depth of class. Finally it computes the similarity of two inputted concepts/classes by using the weight of the $c1$, $c2$ and the common parent which is most close to two concepts.

The depth of a class means the distance from the root class to the class. The depth of class c is calculated by using the function $getDepth(c)$, which is also designed based on APIs in the Jena. Integer set $DEPTH$, initially equal to *empty*, to save the depth of every class. Integer variable *maxDepPosition*, initially equal to 0, is used to save the position of the common parent which is most close to $c1$ and $c2$. Function $getMaxP(DEPTH)$ is for getting the position of the maximum value in the set $DEPTH$. Double variables w_parent, w_c1, and w_c2, initially equal to 0, are used to save the weights of the $c1$, $c2$ and the common parent which is most close to two concepts respectively.
Function $get\_c\_similarity(c1, c2)$:
Input:$c1, c2$

Output: *sim*
Variables:
$sim := 0; CLASS := \phi; C\_CLASS := \phi; DEPTH := \phi;$
$maxDepPosition := 0; dc\_class := ``\ ";$
$w\_parent := 0; w\_c1 := 0; w\_c2 := 0; d := 0; i := 0; j := 0; k := 0;$
Begin
  CLASS:=getAllClasses();
  For $\forall$ $CLASS[i] \in CLASS$
    If *(hasSubClass(CLASS[i], c1)* ***and***
     (hasSubClass(CLASS[i], c2)) ***then***
      $C\_CLASS := C\_CLASS \bigcup \{CLASS[i]\}$
  For $\forall$ $C\_CLASS[j] \in C\_CLASS$
    DEPTH[j] = getDepth(C_CLASS[j]);
  $maxDepPosition := getMaxP(DEPTH)$
  $dc\_class = C\_CLASS[maxDepPosition]$
  $w\_parent := 2^{(getDepth(dc\_class)+1)}$
  $w\_c1 := 2^{(getDepth(c1)+1)}$
  $w\_c2 := 2^{(getDepth(c2)+1)}$
  $d := (1/w\_parent - 1/w\_c1) + (1/w\_parent - 1/w\_c2)$
  $sim := 1 - d$
  Return *sim*
End

5. Evaluation

In order to investigate if the system is effective as a danger-aware system, experiment 1 was performed with the support of 10 subjects. In the experiment 1, a questionnaire was answered by subjects after using the system. Based on the result of the experiment 1, we analyze the system in the following three aspects, 1) whether it is necessary as a danger-aware system, 2) whether the context can be easily defined by users, and 3) the effectiveness and possibility of the system. After that, we perform an additional experiment to evaluate the response time of the system.

Moreover for deeply evaluating the proposed method, we performed the experiment 2 which is focusing on, 1) the effectiveness when the method is used in various real danger scenarios, 2) the accuracy of danger detection, and 3) influence by adjusting weights.

5.1 Experiment 1

Subjects and Method

Ten subjects attended this experiment. Nine subjects are graduated students. Another subject has a part-time job at our University. All the subjects performed the experiment one by one.

Firstly a five minutes' introduction was given to each subject. The introduction was mainly focusing on an outline of the system and how the system works.

Secondly, a five minutes' demonstration was given to each subject to show how to register person/object information, and how to define contexts.

Thirdly, each subject spent around ten minutes to use the system by themselves. Firstly, some RF-ID tags are given to each subject. One is attached to the shoes that the subject wears, and others are attached to the objects. After that, subjects register person/object information using the interface as shown in Fig. 7. Secondly, each subject defines a context by themselves. Finally, the objects attached with RF-ID tags are put on the U-tile sensor network, and the subject walks on the U-tile sensor network freely.

Fig. 11 A subject did the experiment with his daughter.

In addition, a subject did the experiment with his five-year-old daughter as shown in Fig. 11. A context of "A child is close to a knife in the living room at anytime" was defined by the subject, and a danger degree of 0.9 was set with the predefined context by the subject. After the definition, we put a scissor attached with the corresponding RF-ID tag on one of tiles. Then his daughter played on the U-tiles sensor network wearing the shoes attached with an RF-ID tag, by which the daughter can be recognized as a child. The display showed the positions of the daughter and the scissor. When she was close to the scissor, the detected context was recognized as similar to the predefined context, and warning service was provided by audio device.

She was very happy and exciting when doing the trial. Quickly being close to the scissor, and then quickly being far from the scissor.

We asked the subjects to answer a questionnaire after they used the system.

Q1. Did you easily use our interface to define a situation and its corresponding service?
Q2. Do you think there was no stress when you were using the system?
Q3. Did the system provide warning services timely, when you thought you had been approaching the dangerous situation?
Q4. Did you feel/think you are really in a dangerous situation, when the system provided warning service? (Please answer the question in the children' or elder people' point of view you defined)?
Q5. Do you think the situation you met in the experiment when providing services is the similar/expected one to your predefined situation?
Q6. Do you think the system is useful in real life?
Q7. Do you think the system can help avoid elder people and children from dangers?

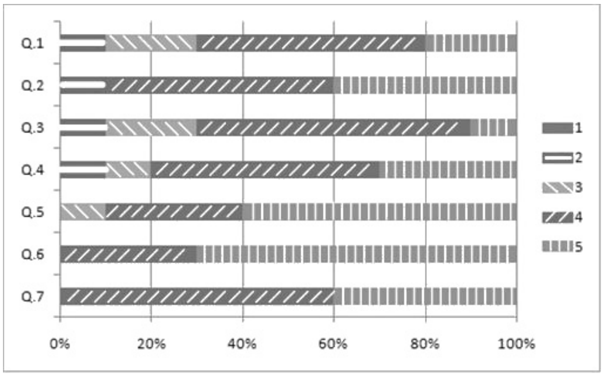


Fig. 12 The results of the questionnaire.

The result of the experiment
The subjects were asked to mark each question with 5 scales as follows, 5: Agree, 4: Partially Agree, 3: Neutral, 2: Partially Disagree, 1: Disagree. The results of the answers are shown in Fig. 12. From Fig. 12 and additional comments written freely by the subjects, we can get some hints for evaluation and improvement of the system.

We performed an interview to four subjects before they used the system one by one. Firstly, we ask him depict some dangerous situations he can image orally. And then we describe basic structure of Situation Theory to him, i.e., 3 primitives. After that we show the demo and let him define contexts by himself. Finally the following two questions are asked to him.
(1) Do you think the defined contexts are match with the dangerous situation you said orally?
(2) Do you think the structure of the defined context can include the information you used to describe a dangerous situation?

All the answers from them are "Yes" based on the above two questions. From this survey, we think the context defined based on Situation Theory is close to the real situation happen around the user. However, two subjects said even though they agree the expressive force of the context represent based on Situation Theory, the current version need more detail design to describe all the situations around them. From this point of view, we will enhance the detail design in the future.

Need for such a danger-aware system
Regarding Q.6 and Q.7, all of subjects give positive answers.

For Q.6, seven of ten subjects agree the system is useful in real life, and other three subjects partial agree.

For Q.7, four of ten subjects agree on that the system can help avoid elder people or children from dangers, and other six subjects partial agree.

From the freely comments, most of subjects think such a system is basically useful in the real life. Some subjects reported that they think this system can avoid some accidents, e,g., a child is scalded by a hot-water bottle or another child is hurt by a socket. Some subjects hope such kind of systems can become a product as soon as possible to avoid dangers in home. From the survey, we can see that such a system is really needed in the real life.

However, some subjects think the warning services should be categorized as several levels, so that the users can decide the danger degree just by selection of the different kinds of warning services. From this point of view, we think that service design should be given in more detail in the future.

Among the ten subjects, four subjects have children, from two-year old to eight-year old. We invite them attend to the experiment to evaluate in the father/mother' point of view. A special question is asked to them as follows, Do you think there are many dangerous situations for children at home?

Four subjects answered "YES". From this survey, we also think such a danger-aware system is need in the real life.

The system can be easily used by general users.
For Q.1, seven of the ten subjects give positive answers, one subject gives a negative answer, and two subjects stand neutral.

Among them, two subjects agree on that can easily use the interface, five subjects partially agree, one subject partially disagree, and two subjects stand neural.

For Q.2, nine of the ten subjects give positive answers, and one subject gives a negative answer.

Among them, four subjects agree on that there is no stress when using the system, and five subjects partially agree and one subject partially disagree.

All of subjects finished designing the services by themselves. From the freely comments, a subject reported that he think the service is also can be designed by some of their family after simple training.

However, some subjects think the interface should be more friendly, such as using more pictures instead of letters in the user interface. And one subject said he has some inconvenience when defining a context, since there is almost no icon. From this point of view, we think that to let users use the system more easily, the interface should be also be improved.

The effectiveness and possibility of such a danger aware system
For Q.5, nine of the ten subjects give positive answers, and one subject stands neutral.

Among them, six subjects on that the detected situation is similar/expected to the predefined one, three subjects partially agree, and one subject stands neutral.

From the freely comments, most of the subjects think the system works well. Some of the subjects think this kind of mechanism can save time for defining contexts, and system can well work. A subject said the definition of contexts based on classes by using STB-Ontology is very useful and easily to define and understand a context after shortly introduction.

From the result of Q.4, eight of the ten subjects give positive answers, and one subject stands neutral, and one subject give a negative answer.

Among them, three subjects on that they feel dangerous if a child meets the detected situations, five subjects partially agree, and one partially disagree.

For Q.3, seven of the ten subjects give positive answers, one subject gives a negative answer, and two subjects stand neutral.

Among them, four subjects on that they can easily use the interface, three subjects partially agree, one subject partially disagree, and two subjects stand neural.

In the free commands, some subjects said the system can provide warning services rapidly, but for danger-aware system it should provide service immediately especially for children. Some other subjects said the distance between the child and dangerous objects is set too short when providing warning services in the experiment. In a real product, different kinds of distance between a user and a dangerous object should be designed/set for providing different warning services. For example, when the distance is 5 meters, some kind of services should be provided, and when the distance becomes 3 meters other different kind of services should be provided. From this point of view, the response time in the hardware and software should be reduced for faster reaction, and service based on different distances and different users should be well designed in detail.

Based on the result of the experiment, we found the timely service is very crucial to the users. Therefore, we performed an additional experiment to evaluate the response time to trigger a service when detecting a dangerous context.

In this experiment, a dell Vostro 200 desktop computer is used as a server, whose configuration is as follows,
(1) Intel (R) Core(TM)2 Duo CPU, E4500 2.2 GHz
(2) 2.0 GB memory
(3) Windows XP SP3

The response time of the system mainly includes two parts. The first part is the response time in hardware, and the second part is the response time in the reasoning mechanism.

In the back of each tile, we set an RF-ID antenna, by which RF-ID reader can read an RF-ID tag information put on the tile. Based on the specification from TI Company [23], the read cycle time of S2000, i.e., the RF-ID reader, is about 200 ms. Meanwhile, compared to the read cycle time of RF-ID reader, the time consumed in CPU for reading information can be ignored.

For reading the information from each tile by using one RF-ID reader, we use relays to build a switch for connecting the RF-ID reader with each antenna. The switching between different tiles takes about 500 ms.

We put an object attached with an RF-ID tag on a tile, and a person approach the object from each of the surrounding tiles respectively. For each of the surrounding tiles, we test ten times. For each test, the number of the classes in Ontology is progressively increased by almost 50 from 50 to 300, and the number of predefined contexts in DB is also increased by almost 10 from 1 to 70. Almost a quarter of the predefined contexts in DB are similar with the detected context. The total average response time in the reasoning mechanism is almost 800 ms.

From the result of the experiment, the total response time of the system to provide services is about 1500 ms, i.e., 200 ms+500 ms+800 ms. With the result, we think the system is competent to be a danger-aware system for avoiding normal dangers. For example, a child is crawl to a stove, a hot bottle or a knife. With the 1500 ms response time, it is enough for warning his/her guardians. However, the response time also can be reduced by improving search algorithm. The delay in the hardware can be reduced by using faster RF-ID reader and electronic switch replacing of relays.

5.2 Experiment 2

Objectives
Evaluate the proposed method in the aspects of
2-a) the effectiveness when the method is used in some typical dangerous situations,
2-b) the accuracy of danger detection,
2-c) the influence by adjusting weights.

In this experiment, we will evaluate the method in the following typical danger situations.

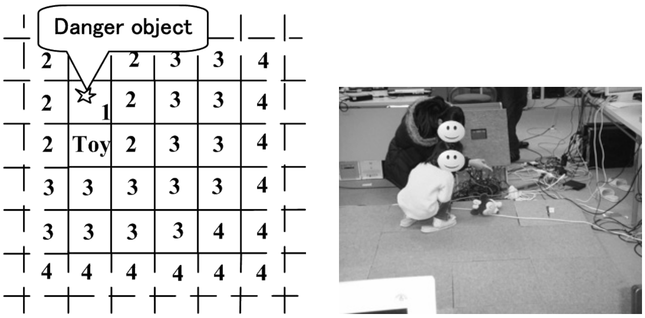
Danger situation 1:
There is more than one object around the user, but just one of them may cause a danger. For example, we consider scenario 1: "*A child is playing with toys, but there is a knife nearby forgotten by someone*".

Danger situation 2:
The danger caused by more than one object together. For example, we consider scenario 2: "*A boy gets a key from his mother, and may insert it into a socket (electrical outlet)*".

Danger situation 3:
Danger caused by the contexts of the two users. For example, we consider scenario 3: "*When scenario 1 happens, his mother is cooking and with a scoop in her hand*."

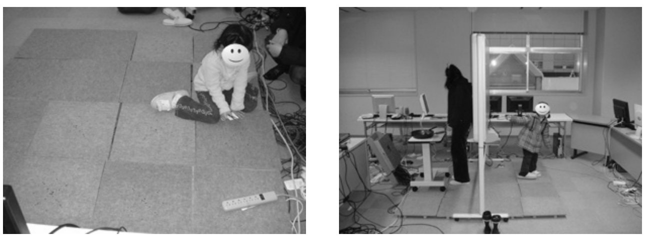
Experiment environment and process
Seven persons attended the experiment. Five of them were playing the role of human observers. They can support us to estimate the danger level for each detected context in the three scenarios, since each of them has experience to take care his/her child. Two children were asked to play on the tiles to help the human observers with estimating how much dangerous the situations are. The children wore shoes attached with different RF-ID tags, representing the information of a two-year-old child and a four-year-old child, respectively. Though the children are not exactly the mentioned ages, we asked the human observers image the ages of the children are 2 and 4 when estimating the danger levels.

For the scenario 1, we set the experiment environment as follows. Firstly, we put a toy and a dangerous object around the toy on the tile as shown in Fig. 13. Then we ask a child to play on the tiles. Finally, we change the dangerous



(a) (b)

Fig. 13 Experiment setting for scenario 1.



(a) (b)

Fig. 14 Experiment setting for scenario 3.

Table 1 Danger type definition.

Danger Level	Level4	Level3	Level2	Level1
Danger Degree	[0, 0.7)	[0.7, 0.8)	[0.8, 0.9)	[0.9, 1]
Danger Type	Not danger	A little Danger	Danger	Very Danger

object by other different objects to evaluate the system. We divide the tiles around the dangerous object into four areas as shown in Fig. 13 (a). When the child is close to the object within different distances (areas), e.g. the tiles marked with different numbers in Fig. 13 (a), the system provided warning services with different sounds. Figure 13 (b) shows a mother was teaching her child safety knowledge in the scenario 1.

For the scenario 2, firstly we put a key and a socket on the tiles and let the children play on the tiles. After that we also change the two objects by other objects to evaluate the system. Figure 14 (a) shows the child got little pliers and be close a socket.

For the scenario 3, we set the experiment environment as shown in Fig. 14 (b). We use a white board to divide the whole sensor network into two spaces to represent two rooms. In the left room, we set the experiment environment similar with a kitchen to support the subjects to image the real situations in daily life easily. Then the child played in the right room and her mother cooked in the left room as shown in Fig. 14 (b). When the child was in a dangerous situation, besides a warning sound for the situation, the system also provided another warning message by sound to let the mother pay attention to her surrounding environment in order to avoid additional dangers due to the reaction to the first warning message.

In the above three scenarios, the system recorded the context similarities and danger degrees of the detected contexts for further analysis.

System set up

Firstly, the whole extent of danger degree is divided into various ranges and different ranges represent different levels and types of danger as shown in the Table 1.

We design an *object ontology* to evaluate our method. A part of the ontology is shown in Fig. 15 (a), which classifies the normal tools used in daily life in detail based on material, size, shape, character, and so on gradually, to show the evaluation procedure. $Thres_{obj}$ is set to 0.95.

Figure 15 (b) and (c) show parts of *person ontology* and *position relation ontology* respectively. $Thres_{per}$ is set to 0.8 and $Thres_{rel}$ is set to 0.9. Generally speaking, danger degree is decreased with increasing of distance, i.e. the longer distance between a person and a dangerous object, the less dangerous. Therefore, in the position relation ontology, we divide the dangerous area into 4 zones, i.e. z_1, z_2, z_3 and z_4. The less the zone's number is, the more dangerous. Moreover, we consider the distance between the dangerous object and the person in a zone is almost increased exponentially, i.e. the distance between the object and the person in $z_i + 1$ is approximately 2 times of the distance between the object and the person in z_i. In our implementation, the area 1, 2, 3 and 4 in Fig. 13 (a) match to z_1, z_2, z_3 and z_4 respectively. Moreover, the minimum sensible value is 50 cm since the length of edge of a tile is around 50 cm. The value can be adjusted by changing the size of tiles for satisfying requires of different applications.

Meanwhile, we suppose the environment of the detected context is the same as the predefined context to simplify the evaluation. We set the weights in the formula (2) and formula (3) in the Sect. 3.4.1 as follows, i.e., $w_{11} = 0.2$, $w_{12} = 0.6$, $w_{13} = 0.2$, $w_{21} = 0.5$ and $w_{22} = 0.5$. Moreover, w_{41}, w_{42}, and w_{43} in the formula (12) are set as 0.4, 0.4, and 0.2 respectively. For simplicity of discussion w_{3i} is set as 1/m, where m is the number of objects in predefined context.

For simplicity of discussion, we just give one threshold for each formula to compute danger degree in the Sect. 3.4, i.e., formula (1) and formula (12). In fact, the whole extent of danger degree can be divided into various ranges to represent different levels of danger in detail, by introducing more thresholds. In this experiment, we use four intervals to represent thresholds to compute danger degree as shown in Table 2.

Experiment result

After the experiment of scenario 3, we performed a short interview to the two mothers of children. Both of them said she felt rather nerves and worried her daughter when hearing the warning sound for her daughter. One of them mentioned she might forget to close gas or with some dangerous objects in hand, e.g. kitchen knife. Another mother said she

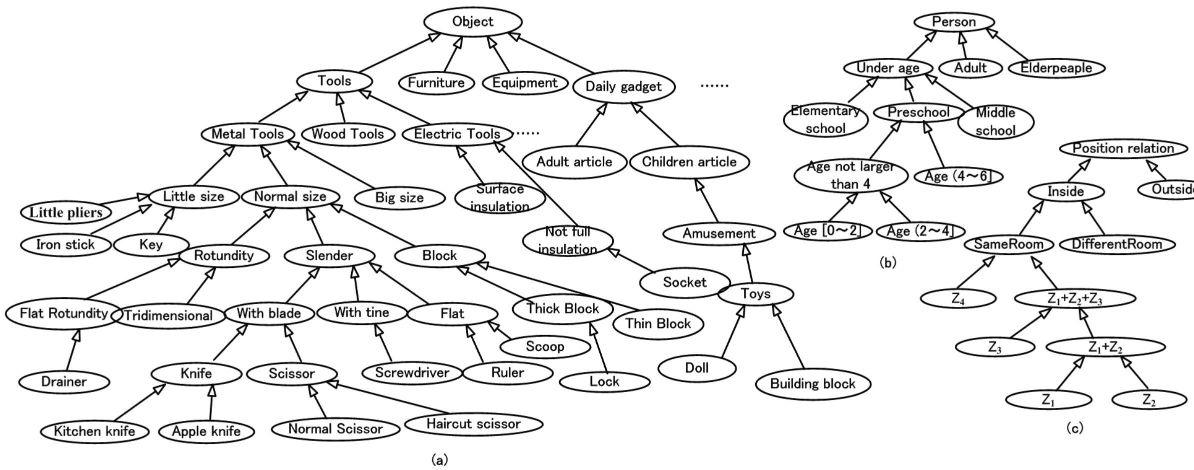


Fig. 15 The Ontologies of *Object*, *person* and *relation*.

Table 2 Danger degree.

Context similarity	Threshold $Thre_{suc}/Thre_{muc}$ [0,0.9)	[0.9, 0.925)	[0.925, 0.95)	[0.95, 0.975)	[0.975, 1]
The danger degree of *suc* ($\delta(suc)$)	$\frac{6}{10}(\delta(pre-suc))$	$\frac{7}{10}(\delta(pre-suc))$	$\frac{8}{10}(\delta(pre-suc))$	$\frac{9}{10}(\delta(pre-suc))$	$\frac{10}{10}(\delta(pre-suc))$
The danger degree of *muc* ($\delta(muc)$)	$\frac{6}{10}(\delta(pre-muc))$	$\frac{7}{10}(\delta(pre-muc))$	$\frac{8}{10}(\delta(pre-muc))$	$\frac{9}{10}(\delta(pre-muc))$	$\frac{10}{10}(\delta(pre-muc))$

※ For the simplicity of discussion, we assume pre-suc is the most dangrous one among the similar contexts based on these ontologies in this table.

Table 3 The experiment result of scenario 1.

The detected contexts No.	Person	Object and relation	Area	CS	DD	DL	E_dl
1	The two-year-old child	Close to a scissor	1	100%	0.85	2	2
2			2	99%	0.85	2	2
3			3	97.6%	0.85	2	3
4			4	94.8%	0.68	4	4
5		Close to a knife	1	99.5%	0.85	2	2
6			2	98.6%	0.85	2	3
7			3	97.1%	0.77	3	4
8			4	94.3%	0.68	4	4
9		Close to a screw-driver	1	98.5%	0.85	2	2
10			2	97.6%	0.85	2	3
11			3	96.1%	0.77	3	3
12			4	93.3%	0.68	4	4
13		Close to a ruler	1	98.5%	0.85	2	3
14			2	97.6%	0.85	2	3
15			3	96.1%	0.77	3	4
16			4	93.3%	0.68	4	4
17		Close to a lock	1	0	0	4	4
18			2	0	0	4	4
19			3	0	0	4	4
20			4	0	0	4	4
21	The four-year-old child	Close to a scissor	1	98.8%	0.85	2	3
22			2	97.8%	0.85	2	4
23			3	96.4%	0.77	3	4
24			4	93.6%	0.68	4	4
25		Close to a knife	1	98.3%	0.85	2	2
26			2	97.4%	0.77	3	3
27			3	95.9%	0.77	3	4
28			4	93%	0.68	4	4
29		Close to a screw-driver	1	97.3%	0.77	3	3
30			2	96.4%	0.77	3	3
31			3	95%	0.77	3	4
32			4	92%	0.6	4	4
33		Close to a ruler	1	97.3%	0.77	3	4
34			2	96.4%	0.77	3	4
35			3	95%	0.77	3	4
36			4	92%	0.6	4	4
37		Close to a lock	1	0	0	4	4
38			2	0	0	4	4
39			3	0	0	4	4
40			4	0	0	4	4

was not sure whether will forget something, but the warning message by considering her situations was very good and helpful in such a situation. However they also commented that the warning message by sound should not be perceived by children, since it might let children more nerves. Meanwhile, the warning message should reflect the current situation of the user automatically. In future we will design the service more appropriately.

The experiment result is shown in the following three tables. The detected danger degree and danger level are recorded by the system and the estimated danger level is made by human observers based on the common knowledge on dangerous situations.

Scenario 1:

For scenario 1, the predefined context is set as "*A two-year-old child is close to a scissor within 0.5 m*" and the predefined danger degree is set to 0.85. Table 3 shows the experiment result of scenario 1. In the following tables, we use CS, DD, DL, E_dl to stand for context similarity, detected danger degree, detected danger level, and estimated danger level respectively.

Scenario 2:

The predefined context is "*A two-year-old child is close to a metal key and a socket within 0.5 m*". The danger degree is set as 0.85.

Scenario 3:

The predefined context is "*A two-year-old child is close to a scissor within 0.5 m and his mother is cooking with a scoop in the different room*". The danger degree is set as 0.85.

The effectiveness in the above three scenarios

From the result in Tables 3, 4 and 5, we can see the system work well in the above three scenarios. Contexts are dynamically composed, and the system computes context similarity and danger degree, and finally provides services based on different danger degree. The results almost match the danger level estimated by the human observers. Based on three predefined contexts, the system recognizes almost 50 contexts in the different danger levels.

Table 4 The experiment result of scenario 2.

The detected contexts					CS	DD	DL	E_dl
No.	Person	Object and relation		Area				
1	The two-year-old child	Close to	Iron stick	1	99 %	0.85	2	2
			socket	1				
2			Iron stick	2	98.1%	0.85	2	3
			socket	2				
3			Iron stick	1	96 %	0.77	3	3
			socket	4				
4			Iron stick	4	93.8%	0.68	4	4
			socket	4				
5		Close to	Iron stick	1	0	0	4	4
			doll	1				
6	The four-year-old child	Close to	Iron stick	1	97.8%	0.85	2	3
			socket	1				
7			Iron stick	2	96.9%	0.77	3	4
			socket	2				
8			little pliers	1	94.8%	0.68	4	4
			socket	4				
9			little pliers	4	92.6%	0.68	4	4
			socket	4				
10		Close to	little pliers	1	0	0	4	4
			doll	1				

Table 5 The experiment result of scenario 3.

No.	The detected contexts	Similarity	CS	DD	DL	E_dl
1	A two-year-old boy is close to/in the area 1 of a scissor	100%	100 %	0.85	2	2
	His mother is cooking with a scoop	100%				
	in the different room	100%				
	Mother and son					
2	A two-year-old boy is close to/in the area 1 of a knife	99.5%	97.3 %	0.77	3	3
	His mother is cooking with a scoop	100%				
	in the same room	87.5%				
	Mother and son					
3	A two-year-old boy is close to/in the area 3 of a knife	97.1%	96.4 %	0.77	3	4
	His mother is cooking with a scoop	100%				
	in the same room	87.5%				
	Mother and son					
4	A two-year-old boy is close to/in the area 1 of a doll.	0	60 %	0.68	4	4
	His mother is cooking with a scoop	100%				
	in the different room	100%				
	Mother and son					

In the three tables, when the distance (number of area) is increased, the detected danger degree is severally decreased. For example, in the context 5 through 8 in the Table 3, when the child is in the area 1 and 2, the contexts are recognized as danger contexts. When the child changes to the area 3 and 4, the contexts are recognized as a little danger and not danger, respectively. It meets the common knowledge that in a certain range the closer distance the more dangerous. Through the experiment result, we can see the method is effective to compute danger degree based on different distances.

Meanwhile, the context similarity varies with the changing of the users. For example in the Table 4, in the context 1 and the context 6, the context similarity is decreased from 0.99 to 0.978 since the detected user is not same with the predefined user. However the danger degrees are almost same since when the distance is very short, i.e. area 1, even for a four-year-old child it is also dangerous. In the Table 1 we also can see, based on the predefined context for a two-year-old child, the system inferred danger degrees for a four-year-old automatically. And the results almost match the estimated danger levels. Therefore, through the experiment results, we can see the method is effective even the user is not exactly same as the predefined one, but similar one to the predefined one.

Moreover, the context similarity varies with the changing of objects. For example in Table 3, when the object is changed from a scissor to a screw-driver, in context 1 and 9, the context similarity is changed from 1 to 0.985. However both of contexts are recognized as danger context since the context similarity is within the same range. Similarly, in the Tables 4 and 5 for a certain distance, the context similarity varies with the changing of objects. If the object similarity is less than a threshold, the system cannot compute danger degree based on this predefined context. Then the system needs find a much more appropriate predefined context. In the above three tables, the danger degree is set to 0 if there is no appropriate predefined contexts. From the experiment results we can see, the system works well when the object is not same as the predefined one but similar one to it. When the detected object is far from the predefined one, the system will try to find other predefined contexts.

Accuracy of danger detection

For the simplicity of discussion, we mainly discuss the accuracy of danger detection based on Table 3. Results shown in other tables can be analyzed in the same way.

From the experiment result we can see, 25 of the detected contexts exactly match the estimated danger levels. 15 of the detected contexts not exactly match the estimated danger level. Among the 15 cases, 14 of the detected contexts are different with the estimated one by 1 level and just one is different by 2 levels. However, all the danger levels of them are detected as more dangerous than the corresponding estimated danger level, which means there is no possibility that when the user is dangerous but system not provide services e.g. warning. We think it is acceptable for

Table 6 The experiment result of scenario 1 by adjusting weights.

The detected contexts				CS	DD	DL	E_dl
No.	Person	Object and relation	Area				
1	The two-year-old child	Close to a scissor	1	100%	0.85	2	2
2			2	99%	0.85	2	2
3			3	97.7%	0.85	2	3
4			4	94.9%	0.68	4	4
5		Close to a knife	1	99.4%	0.85	2	2
6			2	98.5%	0.85	2	3
7			3	97%	0.77	3	4
8			4	94.4%	0.68	4	4
9		Close to a screw-driver	1	98.3%	0.85	2	2
10			2	97.4%	0.77	3	3
11			3	95.8%	0.77	3	3
12			4	93.2%	0.68	4	4
13		Close to a ruler	1	98%	0.85	2	3
14			2	97.4%	0.77	3	3
15			3	96%	0.77	3	4
16			4	93.2%	0.68	4	4
17		Close to a lock	1	0	0	4	4
18			2	0	0	4	4
19			3	0	0	4	4
20			4	0	0	4	4

a danger-aware system.

Moreover, the accuracy of danger detection can be improved by increasing more predefining contexts, dividing danger degree into more levels by setting more thresholds and adjusting parameters, e.g., weights.

For example, in Table 6 we change the weights as follows $w_{11} = 0.2$, $w_{12} = 0.65$, $w_{13} = 0.15$, $w_{21} = 0.55$ and $w_{22} = 0.45$, which improves the effectiveness of object by increasing weight w_{21}. Then the detected context 10 and 14 will meet the estimated danger level as shown in the Table 6, since the object similarity play a greater role when computing context similarity. Meanwhile it also not brings too much influence to the other detected contexts. An optimal setting of parameters can be found by analyzing a large number of data.

6. Conclusion

In this paper, a two stage-composition method based on context similarity is proposed to simplify representation and detection of contexts, and provision of services. The first stage is composition of the useful information to represent the context for a single user. The second stage is to consider contexts of multiple users, which are composed by multiple single users' contexts, to provide services more thoughtfully. Moreover, similarity between a specified context and a detected context is defined and procedures for computation of the similarity are given. Finally the danger degree of the detected context is computed based on context similarity.

By using the proposed method,

(1) The useful information and contexts compose together automatically to provide services.

(2) Developers/users can define a context not only in an instance level, but also in a class level.

(3) When an instance of contexts detected by a sensor network does not belong to a class on which the context is predefined, or the subclasses of the class, the system can compute the similarity between the detected one and the predefined ones to provide services.

Finally, we implement the proposed method in a danger-aware system and evaluate it through two experiments. The result of experiment 1 shows the method is effective to provide warning services, and meanwhile easily used by users. It also shows the proposed method take small enough time to provide services. The result of experiment 2 shows the method is effective in three typical danger situations with three scenarios. Meanwhile, the method is effective with the varieties of user, object and relation based on predefined contexts. Moreover, the experiment result shows the method is acceptable for danger-aware system on the accuracy of danger detection. And it also can be improved by adjusting various parameters.

In future we will apply the proposed method to other danger-aware systems to evaluate and enhance the method and we will also improve the danger aware system in the following aspects, 1) detailed service design for the different dangerous situations, 2) reducing the response time in the hardware and software, 3) more easy-to-use interface, and 4) more accurate to detect the different type dangers. Moreover, we will improve our method to remove the assumptions mentioned in Sect. 5.

Acknowledgement

We would like to say thanks to the anonymous reviewers, who carefully reading our paper, and kindly giving us valuable comments and we also would like to say thanks to all subjects who attended to our experiment and gave our useful suggestions.

This research is partially supported by a funding from JSPS (Japan Society for the Promotion of Science) NO.1865025.

References

[1] T. Strang and C. Linnhoff-Popien, "A context modeling survey," Proc. 1st International Workshop on Advanced Context Modelling, Reasoning and Management, pp.34–41, 2004.

[2] H. Chen, T. Finin, and A. Joshi, "Using OWL in a pervasive computing broker," Proc. Workshop on Ontologies in Open Agent Systems (AAMAS 2003), 2003.

[3] J. Wang, M. Kansen, Z. Cheng, T. Ichizawa, and T. Ikeda, "Designing an indoor situation-aware ubiquitous platform using rule-based reasoning method," Proc. 2008 International Computer Symposium, Taiwan, Nov. 2008.

[4] X.H. Wang, D.Q. Zang, T. Gu, and H.K. Pung, "Ontology based context modeling and reasoning using OWL," Proc. 2nd IEEE Conference on Pervasive Computing and Communications (PerCom2004), pp.18–22, Orlando, FL, USA, March 2004.

[5] A. Kameas, S. Bellis, I. Mavrommati, K. Delaney, M. Colley, and A. Pounds-Cornish, "An architecture that treats everyday objects as communicating tangible components," Proc. First IEEE International Conference on Pervasive Computing and Communications (PerCom'03), 2003.

[6] E. Christopoulou, C. Goumopoulos, I. Zaharakis, and A. Kameas,

"An ontology-based conceptual model for composing context-aware applications," Proc. 6th International Conference on Ubiquitous Computing, 2004.

[7] T. Buchholz, M. Krause, C. Linnhoff-Popien, and M. Schiffers, "CoCo: Dynamic composition of context information," Proc. First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), 2004.

[8] Y. Koumoto, H. Nonaka, and T. Yanagida, "A proposal of context-aware service composition method based on analytic hierarchy process," Proc. First KES International Symposium onIntelligent Decision Technologies (IDT'09), 2009.

[9] J. Yang, Y. Bai, F. Wang, and Y.H. Qiu, "To select the services in context aware systems using concept similarity," Proc. International Symposium on Electronic Commerce and Security, IEEE, 2008.

[10] A. Kofod-Petersen, "Challenges in case-based reasoning for context awareness in ambient intelligent systems," Proc. 1st Workshop on Case-Based Reasoning and Context Awareness (CACOA'06), 2006.

[11] J. Barwise and J. Perry, Situation and Attitudes, MIT Press, Cambridge, MA, 1983.

[12] J. Barwise and J. Etchemendy, "Model-Theoretic Semantics," in Foundations of Cognitive Science, ed. M.I. Posner, pp.207–243, MIT Press, Cambridge, MA, 1989.

[13] E. Tin and V. Akman, "Computational situation theory," ACM SIGART Bulletin, vol.5, no.4, 1994.

[14] Y. Ganjisaffar, H. Abolhassani, M. Neshati, and M. Jamali, "A similarity measure for OWL-S annotated Web services," Proc. 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006.

[15] J. Zhong, H. Zhu, J. Li, and Y. Yu, "Conceptual graph matching for semantic search," Proc. 2002 International Conference on Computational Science (ICCS2002), Amsterdam, April 2002.

[16] Z. Cheng and T. Huang, "A situation-aware support model and its implementation," Proc. Japan-China Joint Workshop on Frontier of Computer Science and Technology, 2006.

[17] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications," MOBICOM2001, July 2001.

[18] http://jena.sourceforge.net/

[19] http://protege.stanford.edu/

[20] J. Hightower, R. Wang, and G. Borriello, "SpotON: An indoor 3D location sensing technology based on RF signal strength," Proc. UW CSE Technical Report 2000-02-02, University of Washington, Seattle, WA, Feb. 2000.

[21] P. Bahl and V.N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," Proc. IEEE INFOCOM2000, vol.2, March 2000.

[22] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," Proc. Mobicom 1999, Aug. 1999.

[23] Texas Instrument, Series 2000 Reader System Reference Guide, TI Company, 2000.

[24] P. Korpipää, J. Häkkilä, J. Kela, S. Ronkainen, and I. Känsälä, "Utilising context ontology in mobile device application personalisation," Proc. 3rd international conference on Mobile and ubiquitous multimedia (2004), pp.133–140, 2004.

[25] A. Kalyan, S. Gopalan, and V. Sridhar, "Hybrid context model based on multilevel situation theory and ontology for contact centers," Proc. 3rd Int'l Conf. on Pervasive Computing and Communications Workshops (PerCom 2005 Workshops), 2005.

[26] S.J.H. Yang, "Context aware ubiquitous learning environments for peer-to-peer collaborative learning," Proc. Educational Technology & Society, vol.9, no.1, pp.188–201, 2006.

[27] J. Wang, K. Ota, L. Jing, M. Kansen, and Z. Cheng, "A smart gate for composing multi-individuals to provide context-aware services," Proc. IWAC2009, pp.658–663, Aizu, Japan, Sept. 2009.

[28] E. Christopoulou and A. Kameas, "GAS ontology: An ontology for collaboration among ubiquitous computing devices," International Journal of Human -Computer Studies, vol.62, no.5, pp.664–685, 2005.

[29] H. Chen, F. Perich, T. Finin, and A. Joshi, "SOUPA: Standard ontology for ubiquitous and pervasive applications," Proc. First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), Baltimore County, 2004.

Junbo Wang received his B.S. in Electrical Engineering and Automation in 2004, and his M.S. in Electric circuits & systems from in 2007, at the YanShan University, China, respectively. He is now pursuing a Ph.D. degree in the School of Computer Science and Engineering at the University of AIZU, Japan. His current research interests include Ubiquitous Context-aware Platforms, Ubiquitous Learning, and Sensor Network.

Zixue Cheng received B.Eng. degree from Northeast Institute of Heavy Machinery in 1982; his Ph.D from Tohoku University, Japan in 1993. He was an assistant professor from 1993 to1999, an associate professor from 1999 to 2002, and has been a full professor, since 2002, in the School of Computer Science and Engineering, the University of Aizu, Japan. His current interests include Distributed Algorithms, Distance Education, Ubiquitous Learning, Context-aware Service Platforms, and Functional Safety for Embedded Systems.

Lei Jing received his B.E. degree in Electrical and Mechanical Engineering from the Dalian University of Technology, China in 2000. M.E. and Ph.D degree in Computer Science from the Yanshan University, China in 2003 and University of Aizu, Japan in 2008 respectively. From 2008 to 2009, he was currently an Asia Jinzai adjunct assistant professor at UBIC, University of Aizu. From 2010, he is a special researcher of the University of Aizu. His research interests include sensor networks, context-aware computing and ubiquitous learning.

Kaoru Ota received the B.S. degree in computer science and engineering from the University of Aizu, Japan, in 2006, and M.S degree in computer science from Oklahoma State University, USA in 2008. She is working toward Ph.D degree in computer science at the University of Aizu. Her current interests of research are mobile agents in wireless sensor networks and ubiquitous learning.

Mizuo Kansen was born in 1963. He received his LL.B from Sophia University in 1987, and his M.S. in Computer Science and Engineering from Aizu University in 2003. He is currently an assistant professor of University-Business Innovation Center at the University of Aizu. He is a member of IPSJ and JSAI. His research interest are e-learning system and case-based reasoning system.