Ubisafe Computing: Vision and Challenges (I)

Jianhua Ma[1], Qiangfu Zhao[2], Vipin Chaudhary[3], Jingde Cheng[4], Laurence T. Yang[5], Runhe Huang[1], and Qun Jin[6]

[1] Hosei University, Tokyo 184-8584, Japan
[2] The University of Aizu, Fukushima 965-8580, Japan
[3] Wayne State University, MI - 48202, USA
[4] Saitama University, Saitama 338-8570, Japan
[5] St. Francis Xavier University, NS, B2G 2W5, Canada
[6] Waseda University, Saitama 359-1192, Japan
ubisafe@googlegroups.com

Abstract. In recent years, a variety of new computing paradigms have been proposed for various purposes. It is true that many of them intend to and really can gratify some of the people sometime, somewhere; a few of them can even gratify some of the people anytime, anywhere. However, at present, none of the computing paradigms intend to gratify all the people anytime, anywhere. With the rapid advance of information technology and the spread of information services, the IT disparity in age, social standing, and race of the people has been expanding and has become a critical social problem of the 21st century. Thus, we have a fundamental question: Can we construct, in a unified methodology, a computing environment that can gratify all the people in all situations, all places and all the time? We propose a novel and inclusive computing paradigm, named ***ubisafe computing***, for studying and providing possible solutions to the above problem. The ultimate goal of ubisafe computing is to build a computing environment in which all people and organizations can benefit from ubiquitous services anytime anywhere with assured and desired satisfaction without worrying or thinking about safety. That is, the ubisafe computing vision emphasizes two basic aspects: ubiquitous safety and ubiquitous satisfaction to all people in all situations. This paper presents the motivations for the ubisafe computing vision but focuses on one basic aspect of ubiquitous safety that covers reliability, security, privacy, persistency, trust, risk, out of control, and other watchfulness while considering novel, essential ubicomp or percomp features of unobtrusive computers, diverse users/people and life-like systems.

1 Introduction

Computers are becoming available anytime and anywhere in many different forms. They are distributed ubiquitously, pervasively and unobtrusively throughout the every-day environments in forms of small or large, visible or invisible, attached or embedded or blended, simple or complex, and so on. Wired or wireless networks connect these computers locally or globally, coordinated or ad hoc, continuously or intermittently, etc. Ubiquitous computing and networking has created tremendous opportunities to provide numerous novel services and applications that are built in both real world and cyber spaces. We are working, learning, traveling, entertaining,

L.T. Yang et al. (Eds.): ATC 2006, LNCS 4158, pp. 386 – 397, 2006.

and doing almost everything with the help of computers. Increasingly, all of us will live in a real-cyber integrated world in which countless physical objects including human bodies will be armed with computers and networks.

Due to the excitement for this new real-cyber integrated world, ubicomp has recently received wide attention from researchers worldwide. Inspired by Weiser's ubicomp vision [1], many new computing paradigms have been proposed, such as pervasive [2], AmI [3], universal, embedded, wearable, invisible, hidden, context-aware [4], context-sensitive [5], sentient [6], proactive [7], autonomic [8], amorphous [9], spray [10], organic [11], persistent [12], or "whatever it is called" computing. Indeed, these new computing paradigms have pushed ubicomp research further by identifying some specific problems and emphasizing some special aspects in the ubiquitous computing world. Although the computing paradigms are for varying purposes with different focuses and approaches, they share a common *any-oriented* vision, i.e., ubiquitous computers as well as services anytime, anywhere and by any means.

Besides the any-oriented service vision, what else are commonly shared or lacked in these computing paradigms? It is apparent that they certainly share safety problems that are severe due to the ubiquitous presence of computers. Also, they share a general goal to offer novel services with some level of satisfaction to their users. Eventually ubiquitous computing will be extended to everyone, which, as a whole, has not been addressed thus far by these computing paradigms. The IT disparity in age, social standing, and race of the people has been expanding and has become a critical social problem of the 21st century. Ubicomp as well as the above computing paradigms address the provision of novel services by arming the computers with a variety of real objects and environments in the real world that is intrinsically rich, changing and uncertain. However, the complexity of various real situations has not been fully realized and studied [13].

Thus, we have a fundamental question: Can we construct, in a unified methodology, an any-oriented computing environment that can gratify all the people in all situations with (almost) perfect safety and satisfaction? To study and provide possible solutions to the above question, we propose a novel and inclusive computing paradigm, named *ubisafe computing,* based on the visions of *ubiquitous safety* and *ubiquitous satisfaction* to diverse people in complex situations. The ultimate goal of ubisafe computing is to build a computing environment in which all people and organizations can benefit from the any-oriented ubiquitous services with assured and desired satisfaction without worrying or thinking about safety.

Due to the broad applicability of ubisafe computing, this paper is focused primarily on one aspect of ubisafe vision: ubiquitous safety. Although computer and network safety has been studied for several decades, we still have several basic questions to answer: (1) Do we really understand all kinds of new risks in using novel computers/networks that are attached, embedded or blended into real objects and environments? (2) Do we really have efficient and effective solutions to precisely predict and further prevent the risks under various situations in the complex computing environment? (3) Can we create risk-less computing environments in which all people can really enjoy ubiquitous services without any anxiety about safety problems covering reliability, security, privacy, persistency, trust, disaster, out of control, and so on? A series of challenges exist to make ubiquitous safe artifacts, systems, and environments

so as to let everyone benefit from ubiquitous services, and simultaneously guarantee their desired safety.

The rest of this article is organized as follows. The next section defines some basic concepts and terminologies used in this paper. In Section 3, we briefly review the state-of-art of safe/safety related computing technologies as well as corresponding fundamental characteristics, then discuss some essential changes and brand new features brought from the ubiquitous or pervasive computing trends, which necessarily call for broader vision for next generation safe computing. Section 4 presents several representative visions/scenarios for ubisafe computing in terms of safety and from different viewpoints. Section 5 further clarifies the safety related ubisafe concepts, and discusses the research challenges and issues towards the ubisafe vision. We conclude the paper in Section 6 with some final thoughts.

2 Definitions of Terminologies

To discuss the safe/safety problem in a unified way, we need a common language. In this section, we define some terminologies that will be used throughout this paper. The vocabulary given here is more safety related and by no means complete. It will be updated with the progress of ubisafe technology.

Since we are talking about ubiquitous computing environment(s), we first define the concept of ***u-objects*** (or ***u-things***). Anything in a ubiquitous computing environment is a u-object, whether it is a human user, a computer, a computer network, a cell phone, a car navigation system, a sensor, or an RFID tag. We can classify the u-objects hierarchically. An un-decomposable u-object is called a ***u-atom***, and a u-object made up from many u-atoms or smaller u-objects is called a ***u-complex***. A u-complex can be used to build a larger u-complex.

The u-objects can be also categorized based on their functions. For example, a basic computing element is called a ***u-element*** or ***u-artifact***, a computing system consisting of the basic elements is called a ***u-system***, and the computing environment containing all u-elements, u-systems, and other related u-objects is called the ***u-environment***. Note that these definitions are relative, because a u-system can be a u-element in a larger u-system. A u-environment can also be a u-system in a larger u-environment. A u-system or u-environment is a u-complex. A u-element may be a u-atom or a u-complex.

Since the human beings play an important and special role in a u-environment, we call a person involved in a u-environment/u-complex a ***u-person***. Since a u-person cannot be decomposed, he/she is a u-atom, although he/she can be much larger in size than a cell-phone, which may be seen as a u-system/u-complex consisting of many smaller processors. Note that a u-person can be an ordinary user, a programmer, a system manager, or someone else. We say a u-person is ***non-negative*** if he/she does not attack other u-objects (include u-persons and u-systems). We can also define (although not absolutely necessary) strictly ***positive u-persons*** as those who never even think about attacking other u-objects and may probably be able to help others. In our common sense, non-negative u-persons are good persons. Similarly, ***negative u-persons*** are those who (sometimes, often or always) try to attack other u-objects. We can also define positive, non-negative and negative u-objects in a similar way.

A u-complex can be defined as a directed graph. Each node is a u-atom or a smaller u-complex. The nodes are related or connected by edges. The relation between two nodes can be passive (follow others' instruction) or active (take own action), positive (help) or negative (attack), steady (fixed) or dynamic (changed), and so on. Normally, all nodes in a u-complex should be non-negative or cooperative and there should be no negative relation between the nodes. In practice, however, we cannot expect that all nodes are reliable or trustable, especially when some nodes are u-persons or ***life-like*** agents as well as smart/intelligent u-things.

We say a u-object is ***absolutely safe*** if it does not have negative relation with any other u-objects in the same u-complex, or if it does not have any relation with any negative u-objects. For example, an absolutely safe u-person does not get attacked directly by any u-object. This u-person is surrounded by some kind of firewall, and all kinds of attacks/dangers are blocked and invisible. In practice, however, it is difficult to keep a u-object away from all kinds of attacks. We say a u-object is ***relatively safe*** or simply ***safe*** if it functions well even if there are some negative u-objects (or related to negative u-objects). A relatively safe u-object should have the ability to detect various attacks. When an attack is detected, this u-object can call some other anti-attack u-object(s) for help. An anti-attack u-object can be embedded into other u-objects, and can be used when needed.

Note that negative relation or negative u-objects is not the only source of risk. In many (or most) cases, the attack/dangers may come from the failure, mistake or trouble of a positive or non-negative u-object. This is why we must study reliability when we talk about safety since the two are closely related. We say a u-object is ***reliable*** if the quality of service provided by this u-object is acceptable to related u-objects in the same u-complex. A u-object is reliable only if it is safe. If it is not safe, it can be a troublemaker even if it is positive or non-negative. In fact, a u-object can be harmed or even damaged by its positively related u-objects. This can happen when the non-negative u-objects make some mistakes or have some trouble themselves. Thus, if a u-object strongly depends on some other non-negative u-objects, it is safe only if all these non-negative u-objects are reliable. We may say a reliable u-object is ***trustable*** to other u-objects only if it is non-negative and reliable. In this sense, most u-persons are not trustable. This is not because u-persons are negative, but because they often make mistakes, and thus they are not reliable in many situations.

In a u-environment, a reliable u-object should be able to provide services with high enough quality to other u-objects, anytime within the lifetime of the u-environment and anywhere in physical or virtual space spanned by the u-environment. For this purpose, a reliable u-object should not stop working during the lifetime of the u-environment. This is obviously too strong a requirement. In practice, we may just employ many u-objects to provide the same kind of services. These u-objects can work together in an asynchronous mode. Some can work, some can sleep, and some can even die. This is called fault tolerance in reliability engineering. It is actually a simple idea borrowed from nature. The question is: how to build a u-system that is reliable as a whole from un-reliable and un-safe u-elements under uncertain situations?

The above u-things related definitions and discussions are the base for depicting our ubisafe vision and presenting some ideas to achieve the ubisafe computing environment. But before that, let us first give an overview of existing representative

computing techniques as well as their trends and novel features in the next section. These techniques appear somewhat disparate but actually share many things in common, although they are called by different names and proposed and studied by different communities or groups.

3 Computing Trends and Profound Novel Features

Safe/safety related computing is not new and has been studied in various computer and computer-based systems for decades. It is related to many technical aspects such as reliability, security, fault tolerance, survivable computing, dependable computing and so on. Some non-technical aspects covering social and human factors have also been studied.

Trusted/trustworthy computing (TC) [14] recently garnered great attention and is intending to build a unified framework or general computing paradigm to cover or integrate various safety related aspects including security, privacy, reliability, risk, reputation, and so on. The United States Department of Defense has defined that a trusted system or component is one that can break the security policy. In fact, as discussed in the last section, a trusted u-object may be the most dangerous source to result in very serious or fatal security problem in a u-environment, in case the trusted u-object is not really reliable.

Trust is indeed very important and greatly expected especially in cooperation among hardware, software, services, etc. In our life experience, trust is only one of the key elements in cooperative processes. The cooperation is just one relationship between entities in the real world. Actually there are many other relationships, such as loosely coupled, mutual use, non-cooperation, competition, fight, and so on. No matter what relationships exist, what users often desire is that they can get things done satisfactorily and safely.

It is a fact that computers and networks have permeated more places and areas in the real world and our life. Thus, the computing environments and features are changing continuously. The computing technology has to accordingly evolve to fit the new environments and features. To predict the next possible computing evolutionary direction or stage must be, therefore, to first identify fundamental changes of computing environment and then find out basic features brought due to the trends of ubiquitous/pervasive computing. In terms of safety impacts, the following three profound features are considered the most essential.

A. Unobtrusive computers attached/embedded/blended to real objects/ environments

The computing systems (the u-systems) are developing in two extreme directions. One is to become bigger, so that the whole world can be covered. The other is to become smaller, so that ordinary u-persons are even not aware of their existence in surroundings. Talking about the latter, nowadays various kinds of computing chips/devices for information acquisition, storage, processing and communication, have become so small that they can be attached/embedded/blended (AEB) to real physical objects and environments. Such AEB computers are often unobtrusive and even invisible. These computers are parts of real object (artifacts, instruments, goods, etc.) to enhance their usages with adding some kinds of information functions. Due to

the small size and power consumption restriction, an AEB computer may have low CPU speed with very limited ROM/RAM and short communication distance. Thus, one AEB computer may be functioned only in a single simple or very limited task, and many of the AEB computers can be interconnected via networks and organized together to complete a large u-complex or high-level u-system. These u-systems will eventually be pervasive in real physical environments of the world. In this sense, the AEB computers are true u-objects, and they are making u-systems truly ubiquitous.

Perhaps the first of the most fundamental factors related to safe/ubisafe in terms of the ubiquity or pervasiveness of the unobtrusive computers is from *physical characteristic oriented* aspects since the u-objects (the AEB computers) would be in environments that may be open, changed, or leading to worse conditions, etc. Let us take some examples. (1) Suppose the working conditions of some u-objects are not good (such as being outside and suffering sunshine, rain and so on), the u-objects may sometimes not work normally or even fail with a high probability. How to quickly detect the anomaly/failure and then take proper actions to make the whole u-system and/or associated u-persons still safe? The hardware redundancy is one of the fault tolerant approaches to improve system reliability. The point is: What is a suitable redundant method to put these various u-objects together, and form a well-organized or even self-organized reliable u-system? (2) Suppose the u-objects are in some open space (indoor or outdoor) in which some people harboring malicious intent can also enter. They may possibly communicate with the u-objects, move/steal/damage them, or put some bad-intent u-objects in the same u-environment. How to guarantee that the system is working correctly/safely/reliably as well as serving true user privacy under such unavoidable malicious behaviors? (3) Generally, the strength of a cryptographic algorithm is related to its complexity, which needs more computations. Due to physical restrictions of size and energy, the computational and communicational resources of AEB computers are often very limited, and thus it is unfeasible to adopt very complex security schemes and protocols. The problem is how to use the limited computing resources to provide enough security/safety protection in a barely controllable u-environment?

B. Diverse users covering all (ubiquitous) people with different features/demands

Computers and their corresponding environments were originally designed for experts, later extended to technical people and now to ordinary people who posses or can gain certain knowledge about computer usages. One of the profound changes for AEB computers is that they are integrated into real things to form u-objects to serve various u-persons including very young children and even babies who have no computer related knowledge at all. That is, the u-persons are to be eventually extended to all people including babies, school children, aged people, disabled persons, men and women who have different professions, etc. However, the usages of AEB computers will be totally different from the conventional ones such as PCs, and the u-objects will be used by u-persons consciously or unconsciously. A person, whether he/she likes or dislikes, may not be able to escape completely from with the presence of computers since the u-objects are becoming ubiquitous in the real ambient environments surrounding us.

Perhaps the second most fundamental factor related to safe/ubisafe in terms of the ubiquity of various users is from *human characteristic oriented* aspects. It has been

realized that a very large proportion of safety incidents in IT related systems is caused by human mistakes or incorrect usages. An example is the recently publicized incident where a stock staff's mistaken data input led to huge money loss. The incorrect usages would become very common in a u-environment since some kinds of u-persons may have no computer knowledge at all, no intention to follow the predefined usage instructions, no awareness of possible dangers approaching, no ability to deal with occurring dangers, and so on. In addition, whether a circumstance is safe or not is relative, varied, and greatly dependent upon the associated people's situations and backgrounds such as ages, states, needs, etc. How to generally describe the complex scenarios of safety and correctly judge concrete safe/unsafe situations is really hard due to the relative and varied safety demands of diverse humans and their characteristics.

C. Life-like systems, i.e., smart/intelligent u-things from small to large scales

Most traditional computers such as conventional PCs and PDAs, although with many functions, are relatively *passive*. They often wait for the users' inputs, take some actions, and send outputs to the users. They usually have no information about users' physical situations and social statuses as well as ambient states. In contrast, the u-systems are becoming relatively *active* by sensing users and/or physical environments and possibly taking some autonomous actions according to the sensed information. Such active character is an outcome of the following three features of the u-systems: (1) computers are too small to be visible when they are attached/embedded/blended in u-objects; (2) too many computers exist to be interact-able simultaneously by a human user; (3) computer systems are too complex to be managed by human users especially for non-technical people. It is expected that the active u-objects may possess some *smart* behaviors, such as context-aware, reactive, proactive, adaptive, automated, autonomic, organic, sentient, perceptual, cognitive, thinking, or intelligent. The u-systems with the above behaviors seem becoming *life-like* systems. A large scale u-systems may include many small scale u-elements and other u-objects, all of which may form various kinds of relationships, passive or active, positive or negative, loosely or tightly, static or dynamic, locally or globally, etc.

Perhaps the third most fundamental factor related to safe/ubisafe in terms of the ubiquity of active/smart u-systems is from *life-like system characteristic oriented* aspects. Being life-like, a u-system should be able to sense necessary information, *i.e.*, so-called contexts. However, the sensed contexts are usually some approximations to states of the real environment surrounding a u-system because the real world is constantly changing, intrinsically uncertain, and infinitely rich. Therefore, the contextual information acquired may not be sufficient and precise enough to characterize a real environment. Due to the incomplete and uncertain contexts, it would be not rare for u-systems to have misjudgments and incorrect decisions, which may probably result in un-safety of their users (other u-objects or u-persons). It is also expected that u-systems can somehow understand their users' needs. The question is how much can be expected to correctly and promptly know the users' true needs? Detecting users' physical states and activities is one thing, while knowing users' needs is a much harder task. When a u-system involves many associated u-objects and u-persons, an event occurring in one u-object may generate a sequence of cascaded events or

consequences to others. How to know if a small local event initiated by something will make other things or even the whole system unsafe?

It can be seen from the above that AEB computers and networks based u-systems have to face many fundamental and hard issues that are novel but crucial to build ubiquitous safe computing environments to offer ubiquitous safe services to all people, at all places, at all times, and under all situations. These call for radically rethinking safety related computing, and natural emergence of ubisafe computing.

4 Ubisafe Vision Related to Ubiquitous Safety

One of main purposes of ubisafe computing is to provide a unified solution for solving various safety problems related to all kinds of u-objects. In the last section, we discussed three fundamental safety-related factors and some new unsafe sources faced by tiny u-objects, by u-systems built from these tiny u-objects, by global u-systems, by life-like systems, and so on. There will be infinitely many issues if we consider the safety problems faced by all kinds of u-objects. Thus, instead of talking about specific u-objects, we talk about the safety problem using a general language in this section.

First of all, the ultimate goal of ubisafe computing is to build a u-environment in which any u-person, an ordinary user, a programmer, a system manager, or others, can get satisfactory services safely anytime and anywhere in any situations. Other non-human u-objects should also be safe in order to guarantee the safety of u-persons. However, as will be discussed latter, some u-objects should be in un-safe states/positions so as to provide a safe environment to guarantee the safety of u-persons. Ideally, we should provide a u-environment in which all ***u-persons*** are ***absolutely safe***. From the definition of Section 2, a u-person is absolutely safe if it is not directly related to any negative u-objects. That is, all kinds of attacks or risks are (should be) invisible to an absolutely safe u-person. When all u-objects in a u-environment are both non-negative and reliable (thus trustable), there will be no risks and attacks. This kind of u-environment is *an extremely ideal vision for ubisafe*.

However, in practice, some of the u-objects are neither non-negative nor reliable. Most u-persons are non-negative but not reliable. A non-negative u-person can also "attack" other u-objects and make trouble due to their mistakes (although he/she is a good person, and does not intend to hurt others). The risks/attacks may come from negative u-persons or u-objects; they can also come from a u-person him/herself. Thus, to guarantee the absolute safety of all u-persons, we must have a specialized u-systems to detect, prevent, and avoid the risks/attacks. For these u-systems, the risks/attacks should be visible, observable, predictable, and counter-able. Thus, *a modified vision for ubisafe* is as follows. The goal is to build some anti-risk/attack u-systems in the u-environment that are so powerful that any u-person can be isolated from risks/attacks from outside; and all kinds of risks/attacks from a u-person him/herself can be predicted and prevented. In this u-environment, all u-persons can receive guaranteed safe services anytime, anywhere, and do not have to think about the safety problem at all.

This poses another question: shall we trust the anti-risk/attack u-systems completely? The answer unfortunately is NO. In fact, no system (existing one or to be developed) can predicate and detect all kinds of risks/attacks produced by many

different u-objects. The most powerful computer systems employed in some of the largest banks cannot even predict some trivial mistakes made by some users. Although an anti risk/attack u-system may make mistakes with a low probability, the cost or consequence as the result of the mistake might be very high or serious. Thus, to make a safe u-environment, it is not a good idea to make just a few giant anti-risk/attack u-systems. It might be better to distribute the risks to many u-objects. In this case, it is inevitable that some of the u-objects related to a u-person are not absolutely safe. If so, neither will the u-person be absolutely safe. Thus, instead of requiring absolute safety, we should build a u-environment in which all u-persons are relatively safe, *i.e.*, getting requested safety level of service. *This is a more practical vision for ubisafe*.

To make all u-persons in a u-environment relatively safe, it is necessary to embed some small-scale risk/attack detection/warning u-systems into the u-objects that are connected or related to each u-person. These small-scale u-systems serve as some of kind of firewalls, and they are not powerful enough to make all kinds of risks/attacks invisible, but should at least be able to detect the risks/attacks. Proper actions can be taken by more powerful u-systems whenever necessary. Note that it is impossible and not necessary to guarantee the safety of all u-objects. If we want to guarantee the safety of u-persons, some u-objects must be un-safe. That is, in order to isolate u-persons from attacks, some u-objects must receive the attacks, and they may face dangers all the time.

In the above discussion we highlighted some vision for ubisafe in terms of safety. The most practical one is to guarantee the relative safety of all u-persons in the u-environment. So far we have not considered the specific risks or attacks faced by a u-person. To make the discussion more concrete, we present some scenarios.

In most cases a u-person is an ordinary user. Let us take a scenario in which a u-system is a health-care system, the u-person is a patient with, say heart disease. In a ubisafe u-environment, the u-system should monitor the u-person anytime and anywhere, so that whenever his/her condition changes, proper advice/instruction/action can be provided. Here, the risk mainly comes from the u-person himself (even if there is no attack from other u-objects). What we meant by ubisafe computing is to provide an infrastructure in which the u-person can get high quality safety service anytime and anywhere. That is, the related u-system should be available anytime and anywhere and it should not stop working anytime.

A similar scenario is the kids-care system. Here, the u-persons are young kids, say less than 10 years old. The risks may come from the kids themselves due to inappropriate actions (*e.g.* trying to jump down from a high place or go across a road with heavy traffic); or come from negative u-persons (*e.g.* people who want to kidnap the kids). The u-system should be able to monitor the kids anytime anywhere, and take proper actions or give proper instructions to protect the kids.

Another simpler scenario is the case when the u-system is a home computer, and the u-person is a human user. There are many possible risks and attacks. The person may play Internet game for too long, and get tired or sick; the personal information might be observed by some other u-objects through spy-ware; the password of his bank account might be stolen by some fishing email; and so on. Ubisafe computing should provide some way to protect the u-persons as well as their privacy/properties.

Besides the above scenarios, there are many situations in which various safety-related problems are important and fatal. Imagining that the safety-critical systems, such as the traffic systems, the flight control systems, the power plant, the financial system, and so on, are all controlled by computers, what will happen if some of them are attacked by some negative u-objects? What will happen if the attack comes from the mistakes of a positive u-person? As described in the last section, since the u-objects are becoming smaller in one extreme, and the u-systems are becoming larger in another, it is becoming difficult to have everything under control.

5 Ubisafe Computing Issues and Challenges

Now let us talk about some technical issues related to ubisafe computing, mainly from safety aspects. In fact, the most important thing is to detect the risk/attack. In the case of heart disease, we can understand the change of the patient's condition using some sensors, and the sensor outputs can be obtained via wired or wireless communications. This is relatively easy. In the case of kids-care, however, this is very difficult. How can we know a kid is going to jump from a high place? In the case of home computer, how can we know the user is too tired and should rest? Some risks look trivial but they are actually very hard to detect by computing machines. Some artificial intelligent (AI) and soft computing techniques are useful, but it seems inadequate to solve these kinds of challenging problems related to real daily life.

Another challenge is that, how can we detect new risks/attacks? A simple example is to detect a virus and kill it from the computer. Usually, we can remember all virus patterns, and can detect a virus through pattern matching. However, this method cannot prevent new viruses from infecting our computers. Thus, before killing the virus, our computers must be infected first. One approach to solve this problem is to learn from the immune system of our bodies. The basic idea is to produce some small risks or weak attacks, and distribute them to all u-objects. These risks/attacks serve as the vaccine and the u-objects, especially the anti risk/attack u-systems can learn from them and improve their immune system. The questions are: How to produce the vaccines and how to improve the immune system? These questions are related to artificial evolution (AE) computation, or biological computing in general.

To offer ubiquitous services, it is necessary to acquire contextual information ubiquitously from the ambient and surroundings, both in physical and cyber worlds. As mentioned in Section 3, the AEB computers can serve both as data collectors and processors. However, since they are not so capable, we must have some mechanism to integrate the information. There can be two approaches. First, we can just transmit the results (after some limited pre-processing) of many AEB computers to a relatively large-scale host computer, and ask the host computer to make the final decision. As mentioned before, this kind of centralized, giant-system approach can have serious safety problems. The second approach is a distributed approach, in which all AEB computers are used to construct a network, and this network can make decisions directly. To allow quick decisions and reactions, the AEB computer network should be decomposable, so that each part (or any part) can be autonomous, and can make decisions based on local information. This is a self-symmetric network, in which each part (any part) is again a complete system that can make decisions. Results obtained from

cellular automaton (CA) might be helpful to build up such a network. Further, since many nodes (a node is an AEB computers or a collection of AEB computers) of the network may provide incorrect results (due to limited processing capability) or may not provide any information at all (due to some kind of failures), the network should be able to make decisions based on incomplete and ambiguous information.

In practice, the network considered above can be a hybrid of many different computers. The construction of this kind of heterogeneous and asynchronous systems might be easy (just put some kind of computer somewhere in the u-environment); their control, management, and maintenance can be a very hard task. This is another challenge for ubisafe computing. The point is, even if each node or each sub-network may change during the lifetime of the network, the network as a whole should be safe, and should be able to provide high quality services to all its users. These issues are also addressed in security computing, non-stop computing, persistence computing, autonomic computing, organic computing, amorphous computing, and so on.

There are lots of challenges to create such expected ubisafe environments. Many new issues and hard challenges are basically from the three essential ubicomp/percomp features: unobtrusive AEB computers, diverse people and life-like u-things. The most fundamental and urgent research is to study all possible unsafe sources of various u-objects from the three aspects of physical, human and life characteristics. The safety related issues become harder when the above three aspects are interwoven with the diversity and complexity of the real world. Ubisafe, or "you-be-safe" or "all-be-safe", is ideal. How about the two fighting sides in a military battle? Here ubisafe becomes "you be unsafe, I am safe". Most people in the world are good but bad persons are not few and they can also use the ubisafe technologies/ environments to do bad things. It is absolutely necessary to study how to prevent malicious uses of the ubisafe technologies to do bad/criminal acts "safely". It would not be enough to only rely on engineering technologies to fully guarantee all safety in the ubiquitous world, which is certainly needed to combine social forces including law, regulation, ethics, and so on, but these are beyond our engineering research.

6 Concluding Remarks

In this paper, we have tried to present the vision for ubisafe computing mainly from safety aspects, although it is very difficult to do so due to both novelty and complexity of the ubisafe concept. Through the discussions presented in this paper, we should at least agree with the following points. First, a unified theory or methodology is needed for systematically solving a variety of conventional and new safety problems faced by different u-objects. Second, the safety guarantee (although relatively) should be offered to all (non-negative) u-persons no matter what the age, sex, race, profession, culture, preference and so on. Third, the study of ubisafe computing itself may motivate some revolutionary progresses in computing technology. We hope that in the not far future, all persons can become (non-negative) u-persons, and all of them can fully enjoy the u-environment provided by ubisafe computing – without talking about the safety problem any more. This is similar to invisible computers that disappear from human eyes as dreamed by Mark Weiser. We are working on the ubiquitous satisfaction aspect of ubisafe vision, which would be more challenging.

Acknowledgements. Dr. Ismail K. Ibrahim and Dr. Thomas Grill were involved in our early discussions on the ubisafe computing vision which the authors greatly appreciate. We would also like to thank Dr. Frank Hsu, Dr. Zhong Chen, Dr. Bernady Apduhan, Dr. Joseph Landman, and Dr. Qing Li for the helpful discussions on ubisafe concepts, terms, problems, issues, etc.

References

1. Weiser, M.: The Computer for the Twenty-First Century. Scientific American, September (1991) 94-104
2. Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. IEEE Personal Communications, August (2001) 10-17
3. The European Union Report on AmI: Scenarios for Ambient Intelligence in 2010. ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf (2001)
4. Shchilit, B.N., Adams, N., Want, R.: Context Aware Computing Applications. Proc. of Workshop on Mobile Computing, Systems and Applications, CA, December (1994)
5. Yau, S., Karim, F., Wang Y., Wang B., Gupta, S.K.S.: Reconfigurable Context-Sensitive Middleware for Pervasive Computing. IEEE Pervasive Computing, 1(3) (2002) 33-40
6. Addlesee, M., Curwen, R.W., Hodges, S., Newman, J., Steggles, P., Ward, A., Hopper, A.: Implementing a Sentient Computing System, IEEE Computer. Vol. 34, No. 8 (2001) 50-56
7. Tennenhouse, D.L.: Proactive Computing. Communications of ACM, 43(5) (2000) 43-50
8. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. IEEE Computer, Vol. 36, No. 1, January (2003) 41-50
9. Abelson, H., Allen, D., Coore, D., Hanson, C., Rauch, E., Sussman, G.H., Weiss, R.: Amorphous Computing. Communications of the ACM 43, No. 5 (2000) 74-82
10. Mamei, M., Zambonelli, F.: Spray Computers: Frontiers of Self-organization for Pervasive Computing. 3rd Italian Workshop From Objects to Agents, September (2003)
11. Müller-Schloer, C.: Organic Computing – On the Feasibility of Controlled Emergence. Proceedings of CODES + ISSS, ACM Press, September (2004) 2-5
12. Cheng, J.: Persistent Computing Systems as Continuously Available, Reliable, and Secure Systems. Proceedings of 1st International Conference on Availability, Reliability and Security, IEEE Computer Society Press, April (2006) 631-638
13. Ma, J.: Smart u-Things – Challenging Real World Complexity. IPSJ Symposium Series, Vol. 2005, No. 19 (2005) 146-150
14. Mundie, C., de Vries P., Haynes, P., Corwine, M.: Trustworthy Computing. http://www.microsoft.com/mscorp/twc/twc_whitepaper.mspx (2002)