WEIGHTED GRANULAR BEST MATCHING ALGORITHM FOR CONTEXT-AWARE COMPUTING SYSTEMS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF INFORMATICS

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

A. BAKi KOCABALLI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF INFORMATION SYSTEMS

JANUARY 2005

Approval of the Graduate School of Informatics

____

Assoc. Prof. Dr. Nazife Baykal

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

Assoc. Prof. Dr. Onur DEMİRÖRS

Head of Department

This is to certify that we have read this and that in our opinion it is fully adequate in scope and quality, as a thesis for the degree of Master of Science.

____

Dr. Altan KOÇYİĞİT

Supervisor

Examining Committee Members

Assist. Prof. Dr. Nazife BAYKAL ____________________________

Prof. Dr. Semih BİLGEN ____________________________

Prof. Dr. Asuman DOĞAÇ ____________________________

Dr. Altan KOÇYİĞİT ____________________________

Prof. Dr. Hayri SEVER ____________________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

A. Baki Kocaballı

ABSTRACT

WEIGHTED GRANULAR BEST MATCHING ALGORITHM FOR CONTEXT-AWARE COMPUTING SYSTEMS

Kocaballı, A. Baki
M.S., Department of Information Systems
Supervisor: Dr. Altan Koçyiğit

January 2005, 94 pages

Weighted granular best matching algorithm is proposed for the operation of context matching in context-aware computing systems. New algorithm deals with the subjective, fuzzy and multidimensional characteristics of contextual information by using weights and a granular structure for contextual information. The proposal is applied on a case: CAPRA – Context-Aware Personal Reminder Agent tool to show the applicability of the new context matching algorithm. The obtained outputs showed that proposed algorithm produces the results which are more sensitive to the user's intention, more adaptive to the characteristics of the contextual information and applicable to a current Context-aware system.

Keywords: Context-awareness, Context-aware Computing, Ubiquitous Computing, Pervasive Computing, Context Matching, Best Matching, Granular Matching, Reminder Agents.

ÖZ

BAĞLAMIN-FARKINDA SİSTEMLER İÇİN AĞIRLIKLI TANECİKLİ EN İYİ EŞLEŞTİRME ALGORİTMASI

Kocaballı, A. Baki
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Dr. Altan Koçyiğit

Ocak 2005, 94 sayfa

Bağlam farkındalıklı sistemler icin gereken bağlam eşleştirmesi operasyonu için ağırlıklı tanecikli en iyi eşleştirme algoritması önerilmiştir. Bu yeni algoritma bağlamsal bilginin değişken, bulanık ve çok boyutlu yapısını, ağırlıklar, ve tanecikli yapı mekanızmalarıyla ele almıştır. Metodun uygulanabilirliğini göstermek için öneri CAPRA – Bağlam Farkındalıklı Kişisel Hatırlatma Ajanı isimli bir hatırlatma ajanı üzerinde denenmiştir. Elde edilen sonuçlar uygulanan yeni eşleştirme algoritmasının kullanıcıların niyetlerine karşı daha duyarlı ve bağlamsal bilginin özelliklerine daha uyumlu ve şuanki bir bağlam-duyarlı sitemde uygulanabilir olduğunu göstermiştir.

Anahtar Kelimeler: Bağlam-Farkındalığı, Bağlamın-Farkında Sistemler, Ubiquitous Computing, Pervasive Computing, Bağlam Eşleştirmesi, Tanecikli Eşleştirme, Hatırlatma Ajanları.

To my family

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Altan Koçyiğit for his guidance, support and motivation throughout the development of this thesis; and for giving me freedom I needed for my graduate studies.

I express my thanks to all my colleagues at Informatics Institute for their help. Many ideas were come out during many discussions with Selda Eren. I would like to specially thank her. I would also like to extend my thanks to Meltem Sönmez Turan for her help in editing process.

Prof. Peter Brown’s works and ideas related to CAR helped me to establish a base for this thesis work. He is a very good inspirational researcher. A valuable discussion with Prof. Varol Akman had also enlarged my vision of context, and I appreciate it very much.

I would like to thank my dear partner Ceren for her love,support and understanding during the completion of this thesis.

I would like to express my deepest gratitude to my parents for their love, encouragement, and support throughout my education. This thesis would never have been completed without their love and endless support. I would also like to extend my thanks to my sister and brother for their morale support.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ACRONYMS

AS	: Activity Score
CAPRA	: Context-Aware Personal Reminder Agent
CAR	: Context-Aware Retrieval
DFD	: Data Flow Diagram
DPS	: Date Position Score
DS	: Date Score
DSS	: Date Spread Score
ebXML	: Electronic Business eXtensible Markup Language
GPS	: Global Positioning System
GSM	: Groupe Spécial Mobile
HCI	: Human Computer Interaction
HMM	: Hidden Markov Models
HPS	: Hour Position Score
HS	: Hour Score
HSS	: Hour Spread Score
IEEE	: Institute of Electrical and Electronics Engineers
IF	: Information Filtering
IP	: Internet Protocol
IR	: Information Retrieval
IR	: Infrared
KSOM	: Kohonen Self Organizing Maps
LAN	: Local Area Network
LS	: Location Score
METU	: Middle East Technical University
MIT	: Massachusetts Institute of Technology
PARC	: Palo Alto Research Center
PS	: People Score

RA	: Remembrance Agent
RCC	: Relevance Context Content
RDF	: Resource Description Framework
RF	: Radio Frequency
TEA	: Technology for Enabling Awareness
TFIDF	: Term Frequency Inverse Document Frequency
TMS	: Total Matching Score
TTS	: Total Time Score
VTT	: Technical Research Centre of Finland
WWW	: World Wide Web

CHAPTER 1

INTRODUCTION

"…There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment, instead of forcing humans to enter theirs, will make using a computer as refreshing as taking a walk in the woods." [1]

Mark Weiser described his vision on Ubiquitous and Pervasive Computing in 1991 by these words. Although during the past decade research on ubiquitous computing has increased tremendously, we are at the moment still far away from that vision but it seems more realizable and viable than before.

1.1. Overview: Pervasive/Ubiquitous Computing

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." [1]. Weiser with these words has described the essence of ubiquitous computing very clearly. Although in computer science literature and among the scientists ubiquitous computing and pervasive computing are used interchangeably, there is a nuance between them.

Actually, ubiquitous means everywhere. Pervasive means diffused throughout every part of something. Pervasive computing involves devices like handhelds - small, easy-to-use devices - through which we will be able to get information on anything and everything. That is the sort of thing that Web-enabled cell phones promise. Ubiquitous computing, though, eschews our having to use computers at all. Instead, it is computing in the background, with technology embedded in the things we already use. That might be a car navigation system that, by accessing satellite pictures, alerts us to a traffic jam ahead, or an oven that shuts off when our food is cooked.[1] We will use ubiquitous computing since it fits our vision better than pervasive computing.

Research on this field is mainly application-driven and focuses on three issues *natural interfaces*, *context-awareness* and *automated capture and access* [2]. Natural interfaces consist of speech, all kind gestures, hand writing and even signals in human brain. Automated capture and access is aimed to remove the burden of record keeping by continuously record live experiences (special events, meetings etc.) of humans and let humans to concentrate on just their business. This idea is parallel to the Weiser's vision of ubiquitous computing, stated as "The use of the computer should be unconscious - all attention should be given to the task at hand". This title also covers the adaptive user interfaces. Finally context-awareness is the key element of ubiquitous computing systems. Context is "that which surrounds, and gives meaning to something else".[2] Parallel to this, context-awareness is the awareness of any application or any devices to their existed context at any moment. Context could be any information that can be used to characterize the situation or condition of an entity which can be a person, place, or object [3].

---

[1] www.computerworld.com/news/2000/story/0, 11280, 41901, 00.html

[2] The Free On-line Dictionary of Computing - http://dictionary.reference.com

1.2. Context-aware Computing

As stated before context-awareness plays a very critical role in ubiquitous computing systems. In order to be context-aware, a system or application should adapt its behavior according to current context changing over time.

1.2.1. Emergence of Context-Aware Computing

Active Badge [4] is considered as the first context-aware application. It is developed in Olivetti Research Lab. at early 90's. In this project, the office personnel wore active badges and a network of sensors around the office space picked up the signals that badges transmitted. By this way tracking of office personnel is done and automatic call forwarding mechanism could be achieved.

ParcTab [5] was developed at the Xerox Palo Alto Research Center at the same period with Active Badges. It worked as a context-aware personal digital office assistant. It uses generally location, nearby resources and time.

After this period research on this field both Context-aware Computing and ubiquitous computing have increased tremendously. Many research initiatives at Georgia Institute of Technology, University o Lancaster, University of Canterbury, Xerox Research Center, MIT Media Library and many others have realized many context-aware research projects mainly application-driven. Tourist guides, remembrance and reminder agents and personal assistants are some of them.

1.2.2. Vision and Challenges

Since it was a relatively new field of research, there are many rooms for development. The aim is to realize effective and efficient provision and usage of contextual information.

When we consider the past decade, research and solutions were application specific and technologically dependent. This prevents the further steps in development and widespread applicability [6] and causes to reinvent and resolve the same problems.

Although there might be many approaches, we can examine the research directions in this field parallel to ubiquitous computing field as follows:

Firstly, to develop context-aware computing applications, it is required to have tools that are based on clearly defined models of context and of system software architecture. There are three main approaches for the architecture of context-aware systems;, and finally the blackboard model [7] based on *client-server* dialog, infrastructure-centered distributed services model [8], and finally the widget approach [9].

Another essential need is the common formal and reusable context representation format and ontology. There must be standards for the provision, usage and storage of contextual information. Ontology including a generic context model [10, 11, 12, 13, 14] for the sharing and processing of contextual information should be developed. This ontology should define all the terms, relations, constraints and ways of reasoning of contextual information [15].

Another key element in context-aware computing systems is the sensors which perceive the most of the physical context data. Requirements of sensing in ubiquitous computing systems can be defined with the following issues: design and usability, energy consumption, calibration, start-up time, robustness and reliability, portability, size and weight, unobtrusiveness, social acceptance and user concern, price and introduced cost, precision and openness [16].

Security and privacy is the very critical but very few of the research efforts considered this seriously. Exploiting the personal information is critical to successful proactivity and self-tuning which are the essences of ubiquitous computing systems [17]. Thus a comprehensive approach to the security issue setting up a balance between proactivity and privacy should be covered [18].

Context fusion is one of areas in which most of the research effort is done. Context/sensor fusion is used in Context-aware systems in two ways. First is to increase the reliability of the acquired context, this is necessary since the information from the sensors are not so accurate and have some error rate. Thus fusing the data from the different sensors will increase the reliability of the system. The second is to reach more abstract levels of contextual information from the lower ones. There are many methods used in fusion of context information. Neural networks [19, 20, 21, 22] and statistical approaches like Dempster-Shafer Theory and Bayesian–Inference [23, 13] are the ones mostly used. Further, more generic approaches covering larger sets of context elements considering scalability issues are required.

One of the least studied areas is the utilization of context history. Although context history is generally believed to be useful, it is rarely used [24]. The most promising thing is to find out the patterns of behavior of the users in system. Together with context history, context sharing also becomes crucial when the reliability issues concerned and heterogeneity of sensing mechanisms and smart spaces are considered.

Finally, sophisticated and configurable context matching mechanisms are required for the better coupling of provided and desired context information and thus more adaptive servicing. In this thesis, we will try to investigate such a mechanism.

1.3. Scope, Aim and Method

The objective of the research presented is to show the current state of the art in context-aware computing research and to show more advanced methods using granular similarities and fuzzy matching methods with weights for context matching operations in ubiquitous context-aware computing systems.

The focus is on the matching of contextual information at the same level of abstraction. Further the following crucial issues: relative importance of contextual fields, priority, and granularity of the context information are also covered.

The methods used include the surveying of the current literature and work done, and implementing case prototypes to show the applicability of the proposed mechanisms.

1.4. Contributions

The main contributions are:

- An overview of the research efforts in the field of ubiquitous computing and context-aware computing including architectural approaches, models, context acquiring and fusion mechanisms and application areas of context-aware computing.
- A granular, weighted best match mechanism for context-aware computing systems.
- Enriching the capabilities of current reminder agents.

1.5. Thesis Organization

The thesis is structured in the following way. In Chapter 2, firstly, the notion of context, characteristics of contextual information, its dimensions and types are assessed. Then, representation and modeling of context and architectural approaches are introduced. Finally, application areas of context-aware computing and research initiatives are figured out. The aim of this chapter is to introduce a survey of the current research on ubiquitous and context-aware computing and also to define a frame of reference for the problem of context matching. In chapter 3, surveyed literature is focused on the context matching issue.

In Chapter 3, first the definition of context matching, operations and work done in this field are examined. Then our approach to the problem of context matching is investigated.

The implementation details and the evaluation of the prototype application CAPRA- Context-Aware Personal Reminder Agent is presented in Chapter 4.

In Chapter 5, summary of the proposals and contributions of this research are introduced. Also the problems, limitations and desired characteristics and future research directions related to the context matching mechanisms are also discussed.

CHAPTER 2

BACKGROUND AND RELATED WORK

Context-aware computing is mostly associated with and considered a key characteristic of ubiquitous computing; it provides proactivity and supports unconscious use of computing capabilities. Thus it will be more useful to examine the Context-aware computing in the context of ubiquitous computing.

In this section, a brief overview of ubiquitous computing will be introduced, and then we will start to investigate our main issue, "context", its definitions, different interpretations, dimensions and characteristics of contextual information. After giving modeling and representation approaches, infrastructural issues will be addressed. Finally, application areas of context-aware computing and research initiatives will be figured out.

2.1. On Ubiquitous/Pervasive Computing

Ubiquitous computing means making many computers available throughout the physical environment, while making them effectively invisible to the user.[1] Ubiquitous computing is held by some to be the Third Wave of computing. The

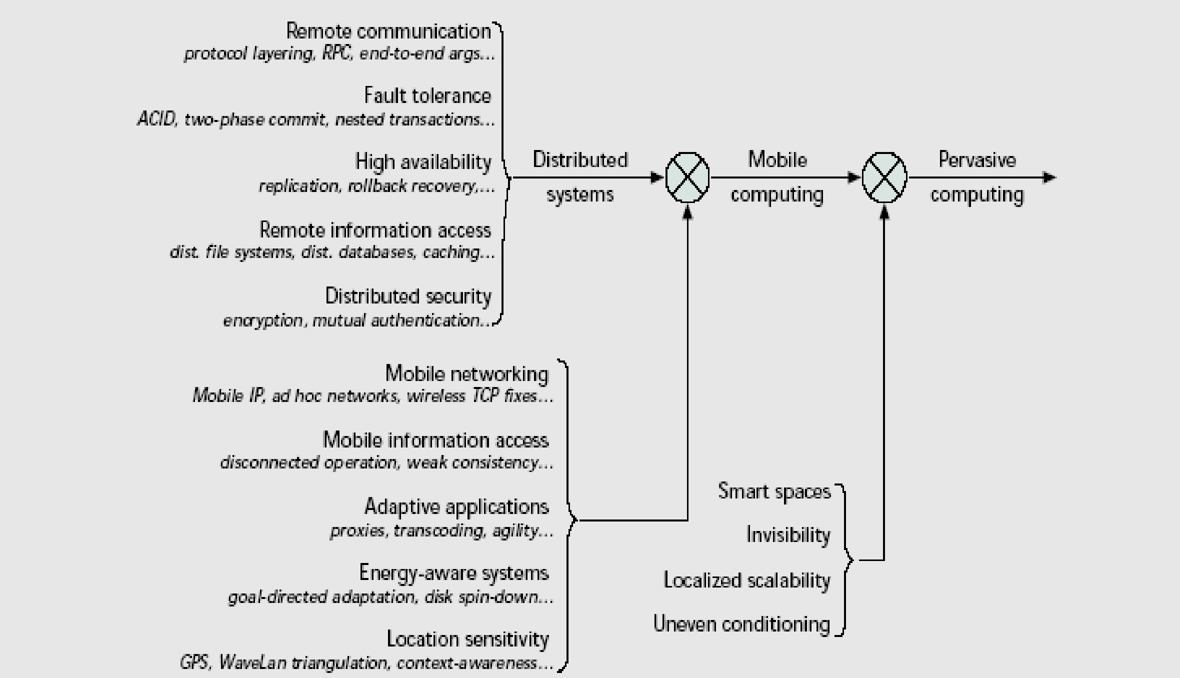
---

[1] The Free On-line Dictionary of Computing - http://dictionary.reference.com

First Wave was many people per computer; the Second Wave was one person per computer. The Third Wave will be many computers per person. [25]

2.1.1. Emergence and Evolution of Ubiquitous Computing

Researchers in the Electronics and Imaging Laboratory of the Xerox Palo Alto Research Center (PARC) in 1991 conceived the early concept of Ubiquitous Computing while they were preparing the proposal of fabricating large, wall-sized, flat-panel computer displays from large-area amorphous silicon sheets. Including this, three intertwined efforts – large wall-display, book-sized ParcPad, and the palm-sized ParcTab emerged within the Ubiquitous Computing program in PARC. Weiser and his colleagues reported their concept and many researches followed [26].



This figure shows how research problems in pervasive computing relate to those in mobile computing and distributed systems. New problems are encountered as one moves from left to right in this figure. In addition, the solution of many previously-encountered problems becomes more complex. As the modulation symbols suggest, this increase in complexity is multiplicative rather than additive — it is very much more difficult to design and implement a pervasive computing system than a simple distributed system of comparable robustness and maturity. Note that this figure describes logical relationships, not temporal ones. Although the evolution of research effort over time has loosely followed this picture, there have been cases where research effort on some aspect of pervasive computing began relatively early. For example, work on smart spaces began in the early 1990's and proceeded relatively independently of work in mobile computing.

Figure 1 Evolution of Pervasive Computing [17]

Towards pervasive computing, distributed systems and mobile computing are major initial steps. Although their internal characteristics cover the needs of pervasive computing, some new solutions have to be sought. Personal computing evolved to the distributed computing with emergence of networking with key features: remote communication, fault tolerance, high availability, remote information access, distributed security [17], as seen in Figure 1.

The appearance of full-function laptop computers and wireless LANs in the early 1990s led researchers to confront the problems that arise in building a distributed system with mobile clients. With the increasing integration of cellular technology with the Web, the field of mobile computing was thus born. Although many basic principles of distributed system design continued to apply, four key constraints of mobility forced the development of specialized techniques: unpredictable variation in network quality, lowered trust and robustness of mobile elements, limitations on local resources imposed by weight and size constraints, and concern for battery power consumption [27, 28].

2.1.2. Issues and Challenges

General issues for ubiquitous/pervasive computing systems could be examined in four aspects: effective use of smart spaces, invisibility, localized scalability, masking uneven conditioning [17].

- *Effective Use of Smart Spaces*: Smart spaces can be enclosed area corridor or meeting room or open well-defined areas like courtyard or quadrangle by embedding computer infrastructure and capabilities in to the space. Automatic adjustment of heating, cooling, and lighting levels in a room based on an occupant's electronic profile could be a typical example of smart spaces.
- *Invisibility*: It is related to Weiser's vision of ubiquitous computing defined as disappearance of technology. Parallel to this minimal user

distraction and unconscious use of computing capabilities should be satisfied.

- *Localized Scalability*: With the increasing sophistication and interaction of the smart spaces, there will be severe implications for a wireless mobile user like bandwidth, energy and distraction. When the number of users increases the problem will be more complicated. In a pervasive system, scalability work on the basis of the inverse square laws of nature, density of interactions should be decreased when the distance from that location increases. Otherwise, overwhelming of distant interactions with little relevance occurs.
- *Masking Uneven Conditioning*: Non-technical factors like organizational structure, economics, and business models play an important role in the penetration process of pervasive computing into the infrastructure. Thus the whole adaptation and penetration take different times and cause different environments with different "smartness". The possibility of the large distances between smartness of environments should be thought and covered. Wearable computing capabilities of users may manage this problem as a solution.

In addition to these key features, context-awareness is the essence of ubiquitous systems. Unlike the reactive mobile systems, ubiquitous systems should be proactive and this proactivity could be satisfied by the efficient and effective use of context by the entities.

2.2. On Context

Context is an important and differently perceived issue in many different fields especially in the field of artificial intelligence including Natural Language, Computational Linguistics, Categorization, Knowledge Representation and Reasoning, and also in Information Retrieval [29]. Research efforts done to represent and model context have been increased rapidly during the last decade.

Context communities[1] have been established and periodic conferences have been organized.

2.2.1. Notion of Context

There are many different definitions and interpretation of context according to the context or reader' interpretation. It is used to describe a multitude of things from descriptions, explanations and analysis. It seems impossible to unify different notions since the notion of context cannot be inseparable from its use [30]. It should be clear what to do with the certain context.

2.2.1.1. Different Definitions of Context

According to the Brézillon [31], context could be described as a concept with complex topology, an ontology, a shared space of knowledge, a consistent set of propositional assumptions, a semantic background, the environment of communication, a set of restrictions that limit the access to parts of a system, a set of preferences and/or beliefs, a window on a screen, an infinite but only partially known set of assumptions, the product of an interpretation, navigational paths in information retrieval, slots in object oriented programming, buttons which are functional, customizable and shareable and possible worlds.

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [3] Dey and Abowd claim that if a piece of information can be used to characterize the situation of a participant in an interaction, then that information is context.

[1] http://context.umcs.maine.edu

Schmidt, Aidoo, Takaluoma, Tuomela, Laerhoven define context as "knowledge about the user's and device's state, including surroundings, situation, and to a less extent, location" [40]. In order to describe contexts they use a three-dimensional space with dimensions *Environment*, *Self*, and *Activity*.

"Context is the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user." [24]. Kotz and Chen focuses on two aspects environmental determination and relevancy in their definition.

Schilit, Adams and Want [33] define context to be the constantly changing execution environment with computing environment, user environment and Physical environment. According to Schilit *et al. [33],* the important aspects of context are: where you are, who you are with, and what resources are nearby.

Actually it is not possible to evaluate the validity of these definitions context-independently; all definitions are valid in their 'context of use'. Notion, definition and scope of the context depend on the context. However when we investigate the definitions and approaches to describe context, we see generally two point of views engineering and cognitive [34].

2.2.1.2. Two Sides of Context

Brézillon claims that the notion of context is dependent in its interpretation on a cognitive science versus an engineering (or system building) point of view, the practice viewpoint versus the theory one. He explains these two sides as: "The *cognitive science view* is that context is used to model interactions and situations in a world of infinite breadth, and human behavior is a key in extracting a model. The *engineering view* is that context is useful in representing and reasoning about a restricted state space within which a problem can be solved." [34]

2.2.2. Nature of Contextual Information

In order to reach a better understanding of contextual information, some essential characteristics of contextual information should be investigated.

2.2.2.1. Context Information Exhibits a Range of Temporal Characteristics

Temporal characteristics of context are related to static and dynamic properties of the context elements based on their frequency of change. Static context information is invariant such as a person's date of birth or relationship with his/her colleagues. Dynamic context changes frequently such as location or activity of a person. It is desired to obtain the dynamic context elements automatically from the sensing mechanisms [11].

2.2.2.2. Context Information is Imperfect

Contextual information might be incorrect, inconsistent or incomplete due to the internal characteristics of the pervasive systems. It might be incorrect due to highly dynamic structure of pervasive environments so the information might become quickly outdated. In addition to this, sensory mechanism and higher level context fusion mechanism might produce faulty results [11].

2.2.2.3. Contextual Information has Many Alternative Representations and Levels of Abstraction

Contextual information as stated before can be represented in various ways and in various levels of abstraction. In pervasive context-aware systems most of the context elements are gathered by sensors and typically sensory data is in raw format, however applications might need contextual information at higher levels of abstraction [11]. For example, a room could be represented as the coordinates or as the room-410 or Jack's office. This example is related to the different representation of the location information; on the other hand abstraction of the contextual information could also be different. For example location information of a person could be room-410 or MM-building or Middle East Technical University etc. All of the information is correct but their granularities are different. There is an increasing abstraction in this example. A leveled representation or hierarchical structure with relationships between the levels might be established [20].

2.2.2.4. Context Information is Highly Interrelated

Elements of context information might be highly interrelated. Some of the elements might be derived from the other element or elements of context. For example, current activity of a person could be derived from the schedule, time or historical context data. Henricksen and Indulska [11] calls this property dependency.

2.2.2.5. Context and Content May Not Be So Different

In a fieldwork study in the project stick-e notes [35], it was observed that the difference between context and content became less distinct. Ecologists using the prototype system had started to be just as interested in the context of the note. Thus they use simply a set of fields that describe a situation instead of two separate context and content parts.

2.2.2.6. Context is an Attribute not an Entity

Contextual information is not useful or meaningful on its own. It should describe or specify some real or virtual entity. For example location information does not mean anything if it does not locate someone or does not belong any entity. It should work as an attribute of entities specifying them in the system [35].

2.2.3. Dimensions of Context

Indeed context has infinite dimensions, if it is thought to be independent. It is certain that if we would like to utilize contextual information, we should clearly define its use and frame of reference.

Schmidt broadly defines the dimensions of context as follows:

- *Identity - e.g. identity of entity.*
- *spatial information - e.*g. location, orientation, speed, and acceleration
- *temporal information* - e.g. time of the day, date, and season of the year
- *environmental information* - e.g. temperature, air quality, and light or noise level
- *social situation* - e.g. who you are with, and people that are nearby
- *resources that are nearby* - e.g. accessible devices, and hosts
- *availability of resources* - e.g. battery, display, network, and bandwidth
- *physiological measurements* - e.g. blood pressure, hart rate, respiration rate, muscle activity, and tone of voice
- *activity* - e.g. talking, reading, walking, and running
- *schedules and agendas*

A good minimal set of necessary context is defined by Abowd and Mynatt [2] with five W's.

- **Who**: The identity of one particular user/entity and possibly together with identities of nearby people/entities.
- **What**: What user is doing? Perceiving and interpreting human activity.
- **Where**: Location information coupled with other context elements such as time.

- **When**: time as an index to a captured record, a baseline of behaviors.
- **Why**: Why is user doing it? Discovering user intention.

Schmidt [12] used six categories indexed by time slices to define context dimensions. The model is depicted in Figure 2.

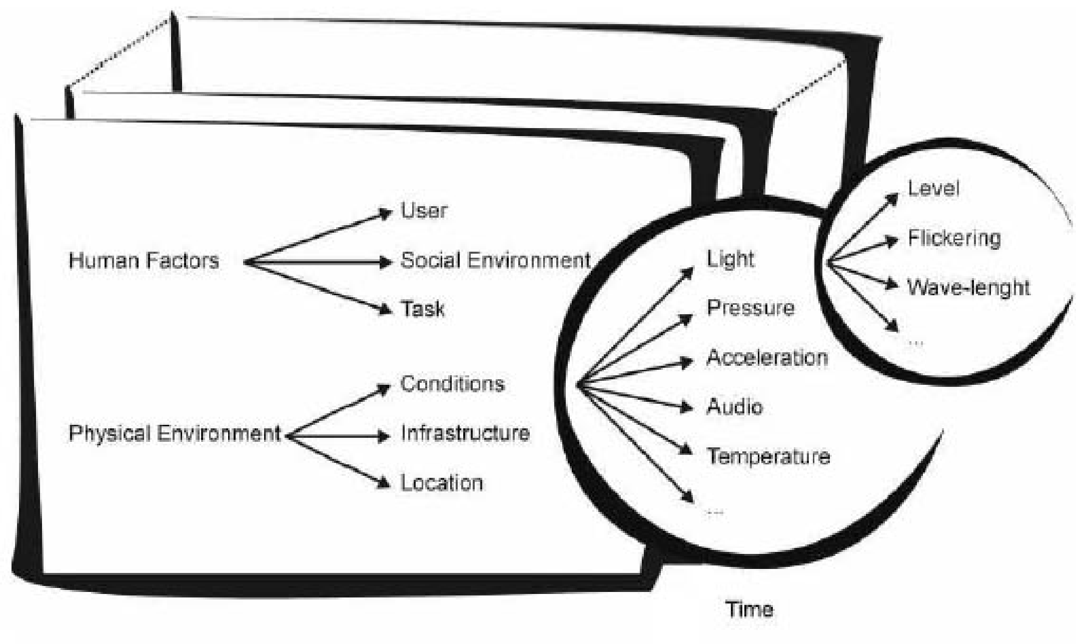


Figure 2 Schmidt's Context Model [12]

2.2.4. Context-awareness

A system is context-aware if it can extract, interpret and use context information and adapt its functionality to the current context of use. [36] Context-awareness can be examined as active/passive or hard/soft context-awareness [11].

2.2.4.1. Active and Passive Context-awareness

If an application proactively acts, i.e. automatically adapts its behavior according to changing current context, this is active context-awareness. However, passive context-awareness occurs if an application just makes the current context available to the user without action [24]. Active context-awareness is a more desired type of context-awareness when it is considered in the scope of pervasive computing vision with the key aspect proactivity and unobtrusive usage.

2.2.4.2. Hard and Soft Context-awareness

Goslar, Bucholz and Schill [92] proposed two context-awareness; hard context-awareness-hardware or sensor based and soft context-awareness-software agents based. Hard context-awareness is suitable for smart environments whereas soft context-awareness works best in comprehensive environments with plenty of information representing different parts of the physical context that can be reused such as Enterprise Application suites. Ideally both types of context-awareness should work complementary.

2.3. Representation and Modeling of Context

In order to provide a standard mechanism to obtain context and to utilize contextual information, a common representation format and a context model is required. However, due to the nature of contextual information, it is not so straightforward or even not possible to create a comprehensible and generic model including every conceivable object of the real world or imagination [37].

Moreover, all processes required in a context-aware computing system, gathering, transferring, storing and interpreting context information should be supported by a representation format [38]. Various formalisms explicitly represent context in system building including logic, rule sets, conceptual graphs and semantic networks [34].

2.3.1. Requirements of Context Representation

Common representation format should be structured to manage huge amount of contextual information, interchangeable to provide efficient transfers among the components of the system, composable/decomposable to provide distributed maintenance, uniform for the ease of interpretation of context profiles, extensible to support future needs of parties and standardized to be exchanged among the different entities of system [38].

2.3.2. Proposed Context Models

So far many models for Context-aware Computing systems have been proposed. However the proposed models are mostly application specific or lacks to support required characteristics of a desired generic context model.

Context toolkit [39] and the multi-sensor architecture [40, 41] provide frameworks that support the abstraction of context information from sensors. Context toolkit provides some basic components, widget, aggregators and interpreters to gather and process the context data in a standardized way. Widgets are responsible from the hiding the details of how the context perceived by providing a uniform interface. Aggregators have the ability to aggregate context information of real world entities such as users or places like meta-widget and finally interpreters take low-level context information and interpret it to higher level contexts [42]. The multi-sensor architecture with TEA Project [43] provides a layered architecture with sensors, cues, contexts and applications layer. It takes inputs called cues from various sensors and combines them into higher level context by using neural networks with a certainty measure for the context values [40].

Another approach to represent context is based on creating a context ontology proposed by VTT [13]. The ontology includes a schema with the structure and the properties for all the ontology's concepts, and a client-usable, extendable vocabulary presenting the terms for describing context information. They used RDF for description syntax. Context in this ontology is described by six properties: context type, context value, confidence, source, time stamp and attributes. First two are required properties for every expression. Moreover, a framework together with the context ontology to describe and utilize user context for handling context information in Ambient Intelligence environments is proposed by Doğaç, Laleci and Kabak [14]. They introduced a way to exploit user context for Web service discovery and composition through ebXML registries with necessary mechanisms for the privacy and security of user context. Gaia [15] is another project using ontology and context predicates to represent context. The structure of these predicates is ContextType (<Subject>, <Verb>, <Object>) and examples are as follows:

- Location (chris , entering , room 3231)
- Temperature (room 3231 , "=" , 98 F)
- Sister(venus , serena)
- StockQuote (msft , ">" , $60)

A more formal and comprehensive model was proposed by Henricksen et al [11]. Their model seems to address the shortcomings of previous models, by providing mechanisms for reasoning and association of context, temporal characteristics, and quality and dependency issues.

The most ignored parts in these models are the privacy and security issues and support for historical context. The most of the models focuses only location and time context and ignores the rest, and again most of them except Dey, Abowd and Wool [44], do not deal with the overlapping and excluding context cases. Scalability is another big issue that has not been covered. It seems impossible to cover the whole set of context and seems certain that to stop modeling in a reasonable point.

2.3.3. Research Questions

Although many attempts have been made, there is no widely accepted model and there are many problems to be covered. What elements of context at which precision should be considered to represent context? What are the common pieces and properties of context? How can the relations between different contexts elements be represented and measured? How can the overlapping and excluding context conditions be handled? What are the effective mechanisms for reasoning and processing context? How to integrate the more cognitive aspects of context into the model?

2.4. Context-aware Computing Infrastructure

In order to develop context-aware systems, clearly defined models of context and of system software architecture are needed. So far many research efforts have been realized to create models, representations and architectures for context-aware systems [44, 45, 46, 47, 48, 49]. Typically these architectures try to provide standard mechanisms for the basic operations of context-aware computing systems; gathering, translating, fusing and storing of context information. Some of them also provide necessary security mechanisms [18, 50].

2.4.1. Context Acquiring Mechanisms and Sensing the Context

In Context-aware computing systems acquisition of context information is realized in two ways [12]. The first one is to establish smart spaces for provision of context information to application and devices like Active Badges or GPS system [51]. In this approach, sensor infrastructure is embedded into the environment. The second approach is to embed the sensing mechanisms on the entities and each entity acquires its related context by itself. Wearable computing studies try to combine these kinds of sensing mechanisms on their systems. The addition of sensors to wearable computers allows them to be more adaptive to the changing context [52].

There are many types of context acquiring mechanisms, which are generally sensor based. Sensing efforts in ubiquitous computing studies have been mainly concentrated on location sensing so far. This is because of the easily foreseeable profits of knowing the location information. So it is reasonable to examine the location context differently from the other context elements.

2.4.1.1. Acquiring the Location

Although it might change according to current context and needs of applications, location information is considered as the most important context element in context-aware computing systems. Actually there are two areas for location sensing outdoor and indoor [24].

Outdoor location sensing is realized mostly by GPS with an accuracy of 5-20 meters for civil use changing according to the position and number of satellites [16]. However, GPS does not work indoor. GSM network [53] and radio beacons [54] are alternatives to GPS for outdoor location sensing.

There are more alternatives in indoor location sensing mechanism. RF transmitters [55], IR systems [51, 56, 57], ultrasonic and radio signals [58], smart floor systems [59, 60] and video recording systems [61] have been used methods for indoor. In addition to being unobtrusive, cheap, scalable and robust; provision of fine-grain location information and frequent update rate very critical for indoor location tracking systems [62].

There are some other location tracking systems using such as IEEE 802.11 standard and mobile IP but the information they provide is very coarse. In order to increase the precision and accuracy of the location information, sensor fusion methods could be considered [63].

2.4.1.2. Acquiring the Other Context Elements

Other than the location, there might be many context elements to be perceived in the environment. Perception of some of these elements either derived by other system parameters and other applications [64] or directly from the sensors [16].

The available sensing mechanisms are optical/vision sensors capturing light intensity, the density, the reflection and color temperature and type of light and also some image recognition services, audio sensors identifying loudness, background noise and base frequency, motion detectors showing the changes on movement, bio-sensors measuring pulse, skin resistance, blood pressure and some other specialized sensors for touch, temperature, air pressure and so on. Constraining issues on these sensing mechanisms are design and usability, energy consumption calibration, start-up time, robustness and reliability [16].

2.4.1.3. Detecting the Context Changes

It is not clear that when a certain kind of context ends and the other contexts starts. Actually, it is related to the consideration of notion of context as continuous or discrete. Context change event is important in Context-aware Computing systems because provision of contextual information depends on this event mainly. A decision of when to deliver context information is very critical. There are many unanswered questions in determining context change.

- In any point value, of anyone of the items constituting the current context may change; In that case, should we consider this context as changed?
- How much change of a value of context element results in an overall context change?
- Should we define specific thresholds for every element of context for determination of a change event?

- Besides all other context elements, time element always changes. There is no fixed moment for time context. What is the ideal granularity for time context? When to consider two time contexts different?

2.4.1.4. Sensor/Context Fusion

There are three types of sensor fusion: first the complementary type sensor fusion, fusing the sensor data to create more complete model, second the competitive type sensor fusion, fusing sensor data that represent the same measurement to increase reliability, and thirdly cooperative sensor fusion, fusing sensor data where observations of one sensor depend on that of another sensor [23]. Incorporating and fusing more of the context elements to obtain higher levels of context is another very critical and popular research field. Symbol clustering Maps [36], Kohonen Self Organizing Maps-KSOM with HMM [65], Naïve Bayesian Classifiers [13], Dempster Shafer Theory [23], Operator Graph Method [67] and Rule based systems [64] are the proposed methods for the context fusion.

Desired characteristics of the classifications algorithms in fusing the context data are online learning, adaptivity, variable topology, soft classification, noise resistance, limited resources and simplicity [21].

2.4.2. Fundamental Operations for a Context-aware System

In order to be context-aware, a system should accomplish the following tasks:

- *gather the information*: current context information should be obtained from the sensors or the smart infrastructure,
- *translate it into applicable format*: most of the sensory data in raw format but applications need processed or symbolic data,
- *combine context to higher level*: in addition to the low level contexts, abstraction to higher level contexts is also desired by the applications,

- *automatically take action (proactivity)*: applications should automatically take action according to the changing context and preferences,
- *make the information accessible to user*: besides proactivity context information should be available to the user to make passive context-awareness possible,
- *provide the necessary security mechanisms*: all precautions should be taken to realize the privacy and security of contextual and intimate information.

2.4.3. Architectural Approaches for context

Many research efforts have been made to create a generic pervasive computing architecture; Winograd [7] summarized the current architectural approaches in three categories: widget model, networked services model and blackboard models. He proposes some of the important key issues to evaluate these architectural approaches: efficiency, configurability, robustness, simplicity.

2.4.3.1. Widgets

A context widget provides applications with access to context information from their operating environment. They insulate applications from context acquisition concerns. Context toolkit [9] is a conceptual and programming focused framework. Separation of acquiring and using the context, transparent and distributed communications, constant availability of context acquisition, context storage and history and resource discovery are the provided mechanisms by context toolkit architecture. The system as stated before consists of sensors, widgets, aggregators, interpreters and the applications. The components and structure of the architecture depicted in Figure 3.

Wu has improved this model by integrating a sensor fusion mediator to the model and realized an uncertainty scheme [23].

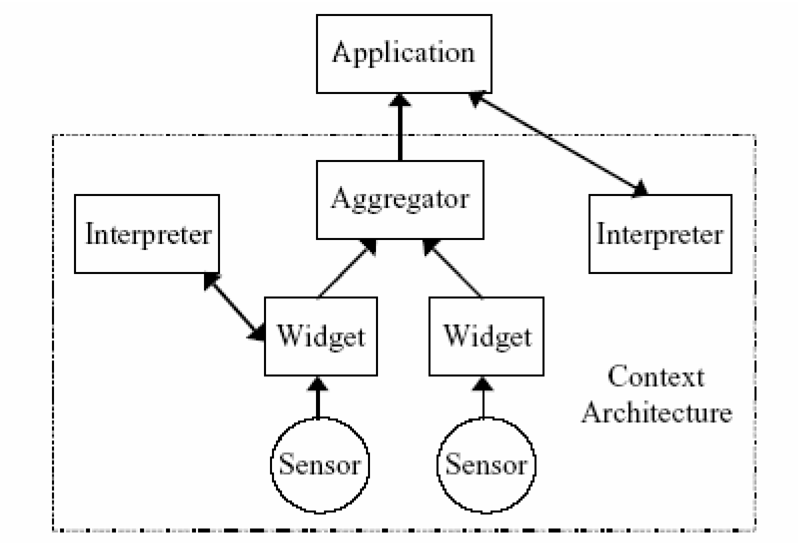


Figure 3 Context Toolkit Architecture

2.4.3.2. Networked Services

In service-based architecture, clients look for location of a service and then set up a connection as needed. Each component contains appropriate code to create connections, marshal outgoing and incoming messages, manage failures and error messages, etc. This adds complexity to each component, and in turn makes them more independent [7]. Resource discovery is an important issue for this approach. Most of the proposed systems have been used service based model so far [17, 27, 37, 36, 67].

2.4.3.3. Blackboards

The blackboard architecture adopts a data-centric rather than process-centric point of view. In blackboard systems, a process posts messages to a common shared message board; entities can receive messages if the previously specified pattern is satisfied. The nature of the pattern matching varies among different blackboard systems. All communications go through a centralized server [7].

Callbacks in the widget model take advantage of tight coupling for efficiency, but require complex configuration and are not robust to component failures. On the other hand, service based model puts much less emphasis on efficiency and tight control, and correspondingly more on configurability (service discovery) and robustness. Finally, the blackboard model is the most loosely coupled model and therefore pays a price in communication efficiency [7].

2.5. Context-aware Computing Applications

During the last decade, many research projects mainly application driven have been realized on context-aware computing field and more generally on pervasive /ubiquitous computing field. Context-aware computing applications have many application areas and special characteristics.

2.5.1. Characteristics of Context-Aware Computing Applications

Pascoe [35] figured out the typical characteristics of context-aware computing applications as follows:

- There are many classes of context-aware applications. Contextual information has been used in many different application areas. Actually almost any application might be context-aware by getting the benefits of their context and it is certain that with the availability of standards in this systems and availability of necessary infrastructures, new classes of application will emerge.
- Context-aware applications tend to be resource hungry. Mechanism such as continuous acquisition and provision of context, formatting and fusing context data needs considerable amount of memory and processor power.
- Context-awareness has a high development cost. Due to the lack of generic architectures and standards, developing the context-aware

applications need a great amount of effort for development starting from the beginning and sophisticated operations needed.

- The computing environments are diverse. Unlike desktop computing, the world of handheld and ubiquitous devices is a very heterogeneous one, with many different computing platforms each with their own special operating systems. This causes a major drawback in developing new software.
- The greater the context the greater the application. Usable context information is not confined with few context element, an application needs many context elements even the contexts of other nearby entities. Parallel to this, scope and scale of the application becomes larger.

2.5.2. Categorization of Features for Context-Aware Applications

There are three categorizations proposed for context-aware applications. Schilit [67] categorized the features under four types. *Proximate selection (a user interface technique)* applications retrieve information for the user manually based on available context. Applications that retrieve information for the user automatically based on available context are classified as *automatic contextual reconfiguration*. *Contextual command* applications are parallel to the passive context-awareness [24] execute commands for the user manually based on available context. *Context-triggered actions* are related to proactivity and active context-awareness concept and refer to services that are executed automatically when the right combination of context exists.

The second taxonomy of context-aware features is proposed by Pascoe [68]. *Contextual sensing* which is again parallel to passive context-awareness and is the mechanism to detect contextual information and to present it to the user; *contextual adaptation* which is related to the proactivity and active context-awareness concepts and is the ability to execute or modify a service automatically based on the current context; *contextual resource discovery* which enables context-aware applications to locate and make use of the resource and services fitting the their current context; and finally contextual augmentation which is the ability to associate data and entities with the current context.

Dey *et al.* [3] used three categories for identifying context-aware features in context-aware computing systems. First is the *presentation* of information and services to a user; second is automatic *execution* of a service; and third one is *tagging* of context to information for later retrieval. The categorization of Dey *et al* seems more generic and clear than the previous ones.

2.5.3. Context-aware Computing Applications

It is useful to examine the context-aware computing applications under the Brown's six categories [69]. Although there are much more applications currently exist, the listed applications are the example applications and cover and represent the whole set of study.

2.5.3.1. Proactive Triggering

These kinds of applications detect the current context and triggers information and services matching to the context. Proactive triggering can be extended to cover filtering out information: a negative proactive trigger can *stop* information being delivered. Fieldwork tools [35], tourist guides [70, 71], office and meeting applications [51, 72] could be considered in this category.

2.5.3.2. Streamlining Interaction

This kind of applications can be thought together with natural interfaces [2]; the key is hugely simplifying the computer interface by taking advantage of the context: i.e. knowing what the user is likely to want to do, and what facilities are available nearby. The Satchel project [73] is an example application that is designed to achieve this.

2.5.3.3. Memory for Past Events

Memory prosthesis project at Xerox European Research Centre in Cambridge [74] was one of the most important works in this field. Forget-me-not prototype [75] was realized in the scope of this project. The aim of this prototype is to automatically capture the events in office environment at the time the event happens (e.g. the room, the other people present, the slides being displayed at a meeting, the current weather, the notes that the user made on their PDA). Context is then used for retrieval. Thus a user might say: "*Find me the notes I made at a meeting about a year ago in the Conference Room. (remember Bill and John were there, and it was raining outside*"[69] another application is the Remembrance Agent of Rhodes [76] 'a continuously running proactive memory aid that uses the physical context of a wearable computer to provide notes that might be relevant in that context'.

2.5.3.4. Reminders for Future Events

This kind of applications deals with the reminding/triggering process of previously recorded information and notes. User specifies a context and tags this context to a note; in the future, if this context is matched against the current context the note will be triggered. Cybre-Minder, a prototype context-aware tool that supports users in sending and receiving reminders that can be associated to richly described situations involving time, place and more sophisticated pieces of context. [77] In the context of this thesis work, we will also provide a prototype reminder agent called CAPRA – Context-Aware Personal Reminder Agent to show the applicability of our context matching algorithm.

2.5.3.5. Optimizing Patterns of Behavior

Applications where capture of context is an end in itself are considered in this category (e.g. fieldwork tools) [35]. Further this type of applications can make intelligent suggestions based on both current context and learnt user behaviors [78]. Example suggestions might be

- 'It is a good time to eat as restaurants are not crowded yet'.
- 'I suggest you go to exhibit 23 now, as it is close and uncrowded.' [69].

2.5.3.6. Sharing Experiences

These applications are based on the sharing the contexts concept between the entities of the system. The basic element needed to implement shared experience is exchange of context among the sharers [69]. The Matchmaker application lets a user rapidly identify an expert user with the knowledge to help solve a problem. An expert's suitability depends on many factors, such as technical expertise, friendliness, proximity, and availability [79].

2.6. Pervasive and Context-aware Computing Initiatives

During the past decade many research initiatives started ubiquitous computing projects;

- **Aura:** "distraction free ubiquitous computing" from Carnegie Mellon University, The project aims to design, implement, deploy, and evaluate a large scale computing system demonstrating a "personal information aura" that spans wearable, handheld, desktop, and infrastructure computers [27].

- **Endeavour:** The University of California at Berkeley's Endeavour project[1] is an academic effort that focuses on the specification, design, and prototype implementation of a planet scale. This smart environment is pervasive—everywhere and always there—with components that flow through the infrastructure, shapes themselves to adapt to their usage, and cooperate on tasks [27].

- **Portolano:** In its Portolano project,[2] the University of Washington seeks to create a test bed for investigating pervasive computing. The project emphasizes invisible, intent-based computing, which infers users' intentions via their actions in the environment and their interactions with everyday objects [27].

- **Sentient Computing:** AT&T Laboratories, Cambridge, UK, is collaborating with the Cambridge University Engineering Department on the Sentient Computing project.[3] The aim of the project is to create an accurate and robust sensor system which can detect the locations of objects in the real world; integrating, storing and distributing the model's sensor and telemetry information to applications so that they get an accurate and consistent view of the model; and finding suitable abstractions for representing location and resource information so that the model is usable by application programs and also comprehensible by people.

- **EasyLiving:** The EasyLiving project of Microsoft Research's Vision Group is developing architecture and related technologies for intelligent environments.[4]

Key system features:

- Computer vision for person-tracking and visual user interaction.

---

[1] endeavour.cs.berkeley.edu

[2] portolano.cs.washington.edu

[3] www.uk.research.att.com/spirit

[4] research.microsoft.com/easyliving

- Multiple sensor modalities combined.
- Use of a geometric model of the world to provide context.
- Automatic or semi-automatic sensor calibration and model building.
- Fine-grained events and adaptation of the user interface.
- Device-independent communication and data protocols.
- Ability to extend the system in many ways.

- **Cooltown:** In Cooltown, technology transforms human experience from consumer lifestyles to business processes by enabling mobility. Cooltown is infused with the energy of the online world, and web-based appliances and e-services give you what you need when and where you need it for work, play, life.[1]

In addition to these MIT Media Lab,[2] Georgia Institute of Technology Future Computing Environments,[3] Helsinki University of Technology's Computer Science Department,[4] and Mobile and Ubiquitous Computing Group[5] in Lancaster University are other important research centers for context-aware computing systems.

---

[1] www.cooltown.com

[2] www.media.mit.edu

[3] www.cc.gatech.edu/fce

[4] www.hut.fi/Units/CSE

[5] www.comp.lancs.ac.uk/research/mobileubiqcomp.html

CHAPTER 3

CONTEXT MATCHING

In this chapter, we will investigate matching of contextual information in more detail. This chapter consists of two main parts. The first part gives general information for context matching and related work. In this part, we will introduce a definition of what a context match is, and we will see what research efforts have been done so far. In the second one, essential characteristics for context matching operation are investigated and then our approach to some of these important context matching issues will be explained.

3.1. On Matching

Matching operations are needed in almost every field of research. Each study uses matching in some way in their processes. Pattern matching, Boolean match, best match, fuzzy and approximate match, string match and record field match are some of the used matching methods.

When there are two information to match or compare, the aim is to figure out how similar one information with the other information. In order to do this many similarity and distance metrics are proposed. Distance and similarity works in opposite. When the distance between two data increases, then the similarity between them decreases. These metrics are excessively used in clustering and classification studies. On the other hand matching and similarity might be considered differently. When we try to match something with another thing, we look for the exact one that we have, whereas in the case of similarity we look for no the exact copy but the similar ones. However, when we consider two approaches, we see that the similarity approach covers the matching approach since if two objects are 100% similar then they match perfectly. Euclidian distance, Minkowski metrics and Manhattan distance are some of mostly used distance metrics.

Similarity or dissimilarity metrics are chosen according to the context of use. There is no generic method applicable to all comparison operations. Different data type in different environment need different measures of similarity and comparison methods.

3.2. On Context Matching

Context matching is a matching process in context-aware computing systems. It matches two context data: provided context and desired context. Provided context information is coming from the sensors, other applications, and generally context providers. On the other hand, desired context information is the query of the context consumers in active or passive format. Context matching is needed when an explicit query is made by the user (active) or a previously recorded query waiting to be triggered (passive) [80]. It is very important to have a matching algorithm which can fit appropriately to the retrieval requirements of context-aware systems, such as precision and speed.

Matching operation on context data highly depends on the used context model and representation, and apparently the needs of context-aware applications. Use and selection of matching methods are directly related with the representation and defined ontology of context. Since the Context-aware computing is relatively a new field of research, there is no commonly accepted standard representations, models, and ontology for context information in these systems. Thus the proposed approaches becomes application specific.

3.3. Related Work

Although context matching plays a very critical role in Context-aware Computing systems, it is one of the least studied issues together with the utilization of context history in this field. So far, research on Context-aware Computing has been focused on new application areas for context-aware computing and provision of generic infrastructures for the development of context-aware systems. Although context matching occurred in all these research applications, it was in its most basic form - Boolean match without any ranking. Boolean match seems working in those systems since the application on this field are quite new and the aim is to just to show the basic capabilities of context-aware computing systems. However, the need for better retrieval and matching systems has been started to be realized. Most elaborative approaches to context matching in context-aware computing issue are done by Peter J. Brown [80, 81].

Brown claims that current search engines take no account of the individual user and their personal interests and their current context [81]. The development of personal networked mobile computing devices and environmental sensors means that personal and context information is potentially available for the retrieval process. He refers to this extension of established information retrieval as context-aware retrieval or CAR. The purpose of the usage of contextual information in retrieval process is to deliver information relevant to current context of user. He has concentrated on the matching operation with best match and context change. He proposed the context of interest related to upcoming context of user, context diary as a context history and a context-aware caching mechanism related to speed and reliability issues for CAR based on the idea of gradual and semi predictable change of context [80].

In a different manner, Azzopardi investigated dynamics and characteristics of the contextual information retrieval mostly in textual sense. He claims that "*if* *the retrieved documents are in a context that corresponds to the user's context, then these documents are far more likely to be relevant than documents which are not."* [30].

Azzopardi [30] propose a relationship between relevance, content and context with the RCC Model as seen in Figure 4.

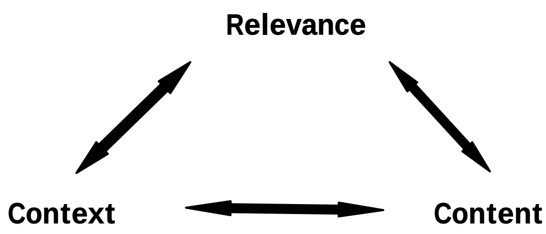


Figure 4 RCC Model

In order to reach a better understanding between relevant and non-relevant Azzopardi [30] gives the following example.

> *"When a user submits a query, say "jaguar" to an Information Retrieval system, because the term has multiple interpretations understanding the context in which it was submitted would remove this ambiguity. So was the user referring to "Jaguar cars", "jaguar cats", or "Jaguar Holiday Company"? We simply don't know unless we know the context such as the users task is "to find an exhaust system for a Jaguar XJ", in which case we can infer that the user wants documents referring to "Jaguar cars". These documents will be in context of the user's information need as opposed to documents on "jaguar cats", however, they may not necessarily be useful. In this example knowing the user's context would allow the system to only show documents about "Jaguar cars", limiting the amount of documents shown to the user. While this may not entirely satisfy the user's information needs, the information returned is more likely to because it is contextually related to the information need."*

Actually above example shows the contribution of context to the retrieval process and supports the idea of likely similarity of documents in similar context. Information Retrieval systems trying incorporate contextual information via some mechanism using models called content plus context models. A successful example of C+Cx model is Google's coupling of a Boolean model with the PageRank™ algorithm [82]. PageRank algorithm incorporates the semantic information that is available from the links between web pages and gives a score out of ten.

Rhodes used fuzzy matching techniques in wearable remembrance agents [91]. In this wearable Remembrance Agent, a continuously running proactive memory aids using the physical context of a wearable computer to provide notes that might be relevant in that context. Location, person, subject, and date context elements are used to tag the notes. When RA is in normal suggestion mode, all the related information in vector format is compared based on an information retrieval technique called Term Frequency Inverse Document Frequency – TFIDF assuming co-occurrence of terms in a document vector and the query vector indicates similarity. He used fuzzy similarity on date and time data. The closer two times or dates are to each other, the higher the relevance score is, following a logarithmic scale.

Makonen, Ahonen-Myka and Salmenkivi [83] present an approach that formalizes temporal expressions and augments spatial terms with ontological information and uses this data in the detection. Their study focuses on Topic Detection and Tracking which is an event-based information organization task where online news streams are monitored in order to spot new unreported events and link documents with previously detected events. For temporal similarity they try to cope with the different representations of temporal information by augmenting the temporal expressions like 'last Monday'. They used time in intervals and tried to find similarity according to the relations of intervals. The more the intervals overlap each other with respect to their lengths, the higher the similarity.

For the spatial similarity, Makonen et al. [83] introduced a geographical ontology enables measuring similarity of the spatial references on a fine scale than just binary decision mismatch. Their aim was to make the necessary connection between the spatial entities in hierarchical structure. They employ a

5-level hierarchy in our knowledge of world as depicted in Table 1. The levels involved depend on the type of the location.

Table 1 Hierarchical Representation for Location [83]

Location	Type	Level 1	Level 2	Level 3	Level 4	Level 5
Delft	city	Europe	W.Europe	Netherlands	Zuid-Holland	Delft
Europe	continent	Europe	–	–	–	–
Haag	city	Europe	W.Europe	Netherlands	Zuid-Holland	Haag
Main	river	Europe	W.Europe	Germany	Rhine	Main
Netherlands	country	Europe	W.Europe	Netherlands	–	–
North Sea	sea	Atlantic	North Sea	–	–	–
Rhine	river	Europe	W.Europe	Switzerland, Germany, France, Netherlands	North Sea	Rhine

A simplified taxonomy with a number of places is seen in Figure 5, as a tree where each node in the tree stands for a location. In order to measure the similarity (μ_s) of two locations in the tree, length of the common path to the sum of the lengths of l_1 and l_2 the paths to elements is compared. The formula could be seen in Equation 1.

$$\mu_s(l_1, l_2) = \frac{(level(l_1 \cap l_2))}{(level(l_1) + level(l_2))}$$

Equation 1

Comparing France and Germany would result in 1/(2+2)=0,25 since the length of common path (Europe) is 1 and length of path to both France and Germany equals to 2. Parallel to this, similarity between China and Paris would give 0/(2+3)=0, and Paris and France would give 2/(2+3)=0,4.

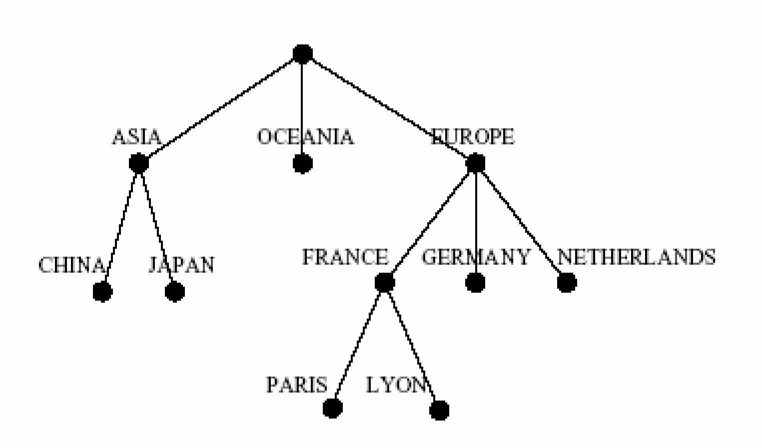


Figure 5 Tree Representation for Location [83]

3.4. Information Retrieval and Information Filtering

Delivery of information is done by two traditional ways, information retrieval - IR and information filtering - IF. Short-term information needs are satisfied by the information retrieval by indicating availability of data relevant to particular requests i.e. queries. On the other hand information filtering is related to long term information needs by maintaining profiles representing the desired characteristics of target documents [84]. Filtering also implies removal of data from an incoming stream rather than finding data in the stream; users see only the data that is extracted. Moreover IR works on static data storage during the query process but dynamic data streams are filtered in IF. The detailed comparison of these two approaches to information delivery is figured out in [85].

Main retrieval models are Boolean retrieval model [86], Vector Space Model [87] and Probabilistic Model [85, 88] Boolean Retrieval Model is based one the idea of “exact match” and other two are based on the concept of “best match”. Boolean retrieval is based on the concept of an exact match of a query specification with one or more text surrogates. The standard operators of Boolean logic are used during the operation. A major problem of exact match is that provided results are not ranked according to the relevance. Best match retrieval model tries to overcome the problems of exact-match retrieval. Vector space model is the widely known example of this type. Documents and queries are treated as vectors in multidimensional space and comparison is done by comparing the vectors with for example cosine correlation similarity. The fundamental assumption is that the more similar a vector representing a text is to a query vector, the more likely that the text is relevant to that query. Weights can be associated with query terms in this model [85]. Probability ranking principle is used in probabilistic information retrieval models. With the utilization of all the evidences, available IR system ranks the texts in database according to their probability of relevance to the query. Evidences might be the statistical distribution of terms in the database [85, 89]. Generally advantages of exact match are efficiency, predictability, simplicity, structured queries; and it works well when you really know what documents you want. However, query formulation is difficult for most users, difficulty increases with the collection size and acceptable precision might result in unacceptable recall.

3.5. Main Issues for Context Matching

There are some critical issues or requirements special for the context matching in context-aware computing systems due to the internal characteristics of these systems and due to being a very new field of research.

- *Fuzziness of context:* Notion and definition is context is not so clear. Its meaning and its usage differ from application to application, systems to systems. It is impossible to draw the borders of context at any point of time, and it is unknown that when a certain context ends and then another one starts. Fuzziness exists for both each of the context element's value and the number of context elements to be included to the scope of current context. Proposed matching methods and modeling efforts for contextual information should handle with this fuzziness issue. Although some applications may be satisfied with very limited dimensions of context information and just an exact matching of context information, more advanced applications need larger set of context elements and more sophisticated context matching methods sensitive to fuzzy characteristic of context information.

- *Scale and dimension of context*: Context may include almost everything, so it is very difficult to categorize and model all context information. Although contextual information needed by the applications may differ very much, it is necessary to provide some generic infrastructure and model of context covering a reasonable set of context elements. However, determining a common set of context information is not so easy. Parallel to this, providing a generic matching mechanism for an unknown set of context elements is also quite tricky, and even is not possible. As stated before a reasonable amount of context elements should be selected according to common characteristics and needs of applications, and for the rest of the elements some configurable and extensible model should be exploited.

- *Different provider profiles*: Most of the context information is obtained from the sensory mechanisms with different measurement types (meters, centimeters, centigrade, Fahrenheit etc.), different precisions and different error rates and reliability. For just one type of context element, there could be many providers with different profiles. Again models provide some mechanisms to handle this heterogeneity and some preprocessing phase might be included to the process of context gathering.

- *Different characteristics of context elements*: The characteristics and dynamics of context elements are very different. Time, location, temperature, activity are the examples of some context elements. Each of these elements has different validity times, precisions, usage types and specifications. Thus provision of generic mechanisms for modeling and matching are difficult but specific methods for each type seems not so feasible.

- *Different abstraction levels of context elements:* For many of context elements there could be many abstraction levels. For example, location element of a context record might be Turkey, Ankara, Çankaya, METU, MM-Building or room-410 or just outdoor/indoor. Similarly, time could be January 2, 14:50, first week of the January or just 2005. For the activity element of context, stationary, non stationary, meeting, working or talking is some of the possible abstractions. There could be some structures for abstraction levels like a hierarchical structure in order to set up some relation with different abstractions and also alternative representations [20, 29]. Prekop and Burnett proposed an activity centric modeling approach focusing on the context that surrounds the performance of an activity by an agent. The key components of the activity-centric view of context are agents and activities. An agent can be a single person, a group of people, or an intelligent (or semi intelligent) machine. The agent is any entity performing the activity. An activity is a description of something being done by the agent. They organize the context according to their scope with a cascading view. from the very broad to the very specific, with broad activities often containing more refined or specific activities [29]. A cascading view of context and activities are seen in Figure 6.

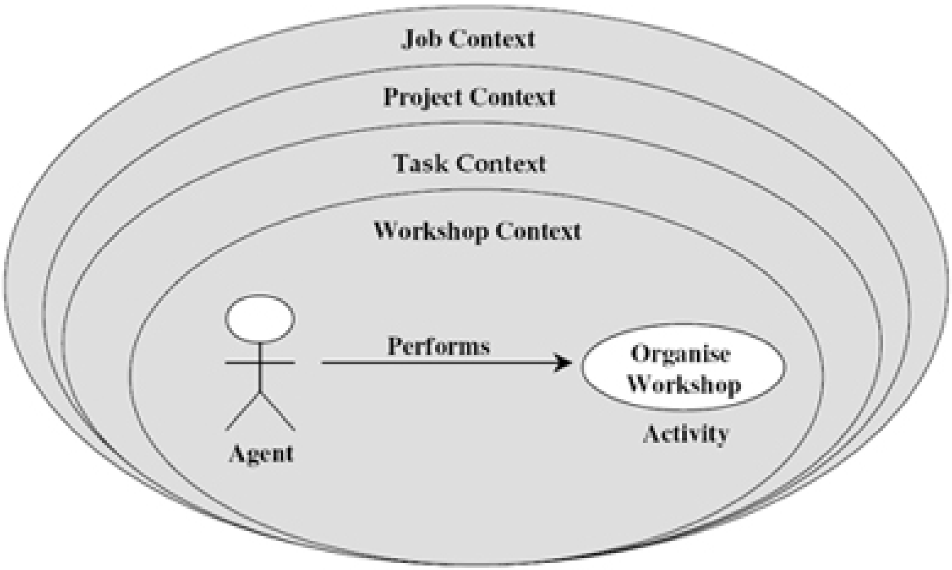


Figure 6 Cascading View of Activity Context [29]

A similar approach for temporal and spatial elements of context is proposed by Makonen et al. [83].

- *Uncertainty of contextual information:* Uncertainty is an issue in both gathering and processing and reasoning parts of context-aware systems. There some preliminary works dealing with the uncertainty in context-aware systems [89, 90].

- *Different representation formats:* Various formalisms explicitly represent context in system building including, logic, rule sets, conceptual graphs and semantic networks [34]. Use of an appropriate representation approach depends on different needs of applications.

Thus provision of a common representation satisfying all different needs seems not possible. In addition to formal approaches, different points of views are also possible for such as both of the location contexts represent the same place my office and room-410 or for the people contexts such as Ceren and my girlfriend.

- *Difficulty of describing necessary distance and similarity metrics:* A typical context record consists of many fields textual, numeric and symbolic. Although many methods available in IR literature for these, they are not applicable to most of the context elements. For example, how can we define a similarity metric between two activities such as studying and reading or walking and running? How can the spatial relations with the locations with symbolic or textual representations be established? When we consider two context records Mersin and Bolu with the query field Ankara, what will be the similarity score of those two records? If we use exact match, it is 0, but what about when we use best match? Should we consider the geographical distances as a criterion? These kinds of problems should be solved.

- *Evaluation methods:* Precision, the ratio of the number of relevant documents retrieved to the total number of documents retrieved, and recall, the ratio of the number of relevant documents retrieved to the total number of relevant documents, might be the main evaluation methods of effectiveness for the retrieval of context like the traditional IR. Precision is believed to be more important in context-aware systems [81]. Meaning or interpretation of relevant is directly related to current context. How well is the fit between the query context and document context? Evaluation is subject to point of views and understanding of context. Moreover, there is no commonly used or widely accepted evaluation methods for context-aware systems and its sub-domains.

3.6. Our Approach

In this section, our approach will be explained according to the issues stated in previous section. Since the focus of this study is on matching process, some of the issues discussed before may not be addressed. The aim is to enhance the current matching operations in context-aware systems and make matching of context more sensitive the user needs and queries.

3.6.1. Features Related to Context Matching

- *Fuzziness of context:* In order to make the matching operation more sensitive for the context queries, a weighted granular best-match algorithm is proposed. By the use of this algorithm each record has a matching score value between 0 and 1 where 0 represents the worst match 1 represents the perfect match. In addition to this, a fuzzy matching operation is done for the time dimension of contextual information. Although the other dimensions of context have fuzzy characteristics, fuzzy matching is applied to only the time dimension in order to show just the usefulness of this approach. For time dimension, time is considered as fuzzy ranges with five different patterns. These patterns are investigated in section 3.6.2.2. The other fuzziness issue related to inclusion of more context element was not addressed.

- *Scale and dimension:* It is not intended to create a generic context model representing a large set of context elements. The aim is to show the applicability and benefits of more advanced algorithms for context matching. Thus a four dimensional context model is used with the location, time, and activity and people context elements.

- *Different provider profiles:* heterogeneity of provided context information issue was not addressed and it is assumed that the provided context information is at the finest granularity with full reliability.

- *Different characteristics of context elements:* For each context element included in the model, there will be a specific matching algorithm proposed. By this way their internal characteristics are better satisfied. For the location and activity elements, granular matching methods were exploited. For the time field, a matching algorithm considering the time as ranges and making the evaluation according to the span of range and position of query context is proposed. For the people context, exact match is applied.

- *Different abstraction levels of context elements:* for the location there are six levels of abstraction hierarchically structured. For time and people context elements, there is one abstraction level and finally, two abstraction levels were used for the activity.

- *Uncertainty of contextual information:* Uncertainty issue is not addressed.

- *Different representation formats:* it is assumed that only one provider of context information exists, so there is one representation of context available.

- *Distance and similarity metrics:* For the location and activity fields a hierarchical tree model is exploited and matching scores are calculated according to the place of the location on the tree. Similarity of two time value is determined according to their span of range and positioning of time in range. It is a kind of fuzzy match between two time information. Exact match is used for the people field of context record.

- *Evaluation methods:* Since the lack of necessary evaluation methods, some typical scenarios are introduced and experimented. In these scenarios, the usage and benefits of the proposed matching algorithm are tried to be figured out.

3.6.2. Related to Matching Algorithms

As stated before there are individual matching algorithms for each dimension of context. After calculating the individual matching scores for each context element, an aggregation function - weighted mean, is used to obtain a single matching score. Location match and time match are the focuses of this study; activity match and people match are included to increase the contextual dimensions.

3.6.2.1. Location Matching

A hierarchical tree representation with multiple granularities and with six levels is set up for the location dimension of context. At the top, location is represented very coarse, and then when we go down in the tree the granularity becomes finer.

Symbolic representation is chosen for the representation. At the first level, location values consists of one digit, at the second level two digits, at the third level three, at the fourth four, at the fifth five and finally at the sixth six digits. This structure is depicted in Figure 7.

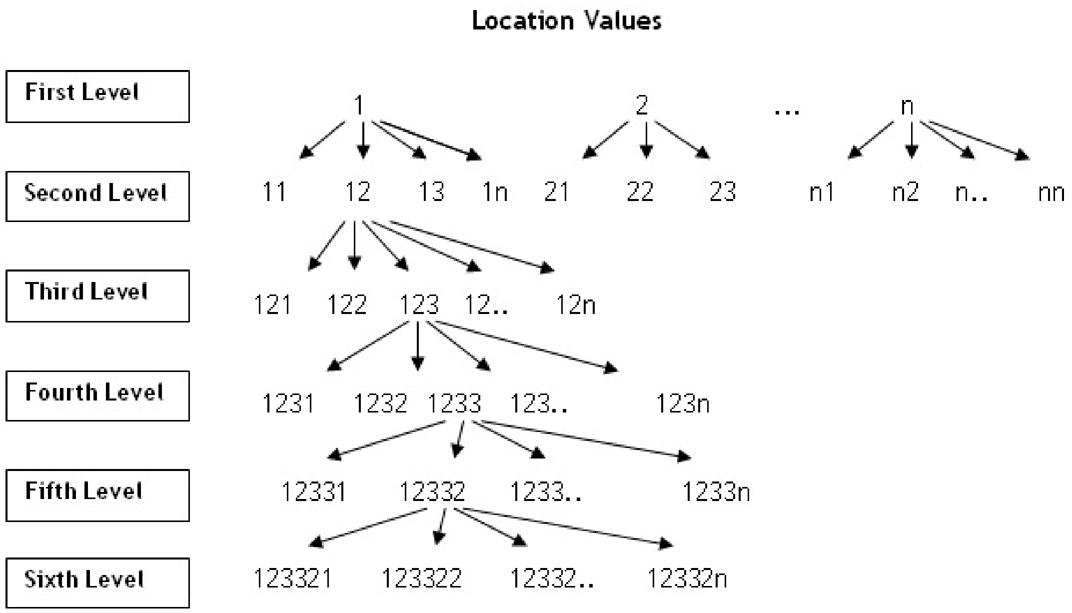


Figure 7 Granular Tree Representation for Location Context

The depth of tree may change depending on the need. Each level corresponds to a real life spatial representation. The rules are that places at the same level on the tree should be at the same granularity and the upper level element should be common for the lower ones and should cover them. Granularity should become finer from top to bottom.

Mapping of the granular levels to the real life spatial entities and corresponding average distances and similarities are depicted in Table 2

Table 2 Similarity and Distance Values According to Granularity Levels

Real Life Places	Levels	Average Distances (meters) for Levels	Similarity
District	Level 1	4000	0,2
Neighborhood	Level 2	750	0,5
Street	Level 3	150	0,7
Building	Level 4	40	0,85
Floor	Level 5	20	0,95
Room	Level 6	5	1

As stated before all elements are subject to change in this structure; for example average distance may start from 100 meters and end with centimeters. Average distances and corresponding similarity measures should be carefully chosen according to the needs.

Similarity of location values are expected to decrease when the distances between the locations increase. This situation can be better seen in Figure 8. The shape of the curve depends on the granular difference between levels.

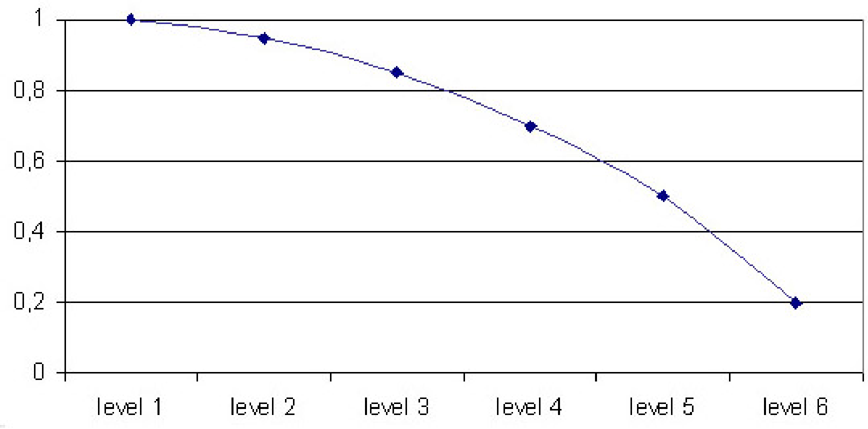


Figure 8 Similarity Scores according to Granularity Levels

Let's examine the following situations: first location field of our query context is coded as 112 and location field of target contexts are coded as follows 113161, 112112, 111311, 112532 and 212321; second the query code is 112322 and target codes are 211333, 112222, 112322, 112354 and 152231. The similarity of each pair is calculated as number of the common digits starting from the left side of the code. The similarity calculation of each pair can be seen in Table 3. Total satisfaction score is used when all the digits of the query context are satisfied by the target context independent from the actual similarity value according to number of common digits.

Table 3 Calculation of Similarity Scores for Location Context

Query Context	Target Context	No of Common Digits	Total Satisfaction	Similarity Score
112	113161	2	No	0,5
112	112113	3	Yes	1
112	111454	2	No	0,5
112	112532	3	Yes	1
112	341321	0	No	0
112322	211333	0	No	0
112322	112222	3	No	0,7
112322	112322	6	Yes	1
112322	112354	4	No	0,85
112322	152231	1	No	0,2
112322	161121	1	No	0,2

3.6.2.2. Time Matching

We consider the time with two sub-parts date and hour. Thus matching of time fields of two context information is done first considering the date and then hour data. Total time matching score is calculated by the weighted mean of date and hour scores. The formula is for the time matching could be seen in equation 2.

$$TTS = DS*w_{ds} + HS*w_{hs} \qquad \textit{Equation 2}$$

$$DS = DSS*w_{dss} + DPS*w_{dps} \qquad HS = HSS*w_{hss} + HPS*w_{hps}$$

where DSS is Date Spread Score, DPS is Date Positioning Score, HSS is Hour Spread Score, HPS is Hour Positioning Score, HS is hour score, DS is date score and finally TTS is total time score; and w_{ds}, w_{hs}, w_{dss}, w_{dps}, w_{hss}, w_{hps} stand for the weights for the scores for each parameter.

Spread scores for date and hour are calculated according to the span of the range of date. The rule can be defined as: larger the span lesser the score. However, the score should not go under a certain point, since it is a still a match according to the query criteria. Calculation of spread score is depicted in Figure 9, where a stands for the minimum score for a match, and n denotes the maximum range. After the point 'n' all the ranges are treated as the same, since they are still a match and scores lower than a certain point can not represent this situation. The function needs not to be linear, but it should be decreasing.

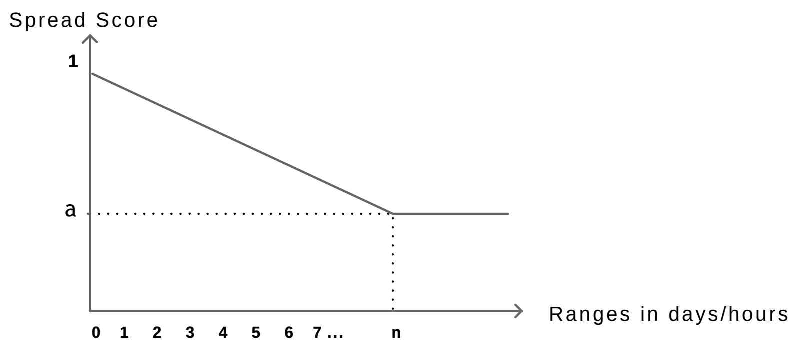


Figure 9 Range Spread Score Function

Determination of 'a' and 'n' values depend on the needs of the applications. These values can be derived from some sort of learning mechanisms like neural networks etc.

Positioning score is calculated according to the position of the query context date or hour in the range of target context. There are five options to be selected for the positioning scores, but there could be more if desired. For each option, the score for a certain position may be different. First option is the equally weighted position scoring, the second is the center weighted position scoring, the third is the beginning weighted position scoring, the fourth is the end weighted position scoring and finally the fifth is the edge weighted position scoring. The graphs for each option are shown in Figures 10-14, where r1 and r2 denote start and end of the target range respectively, and q1, q2 and q3 in Figures 10-14 stand for the quartiles for the range. Again quartiles need not to be used; one might use just the midpoint of the range as break points.

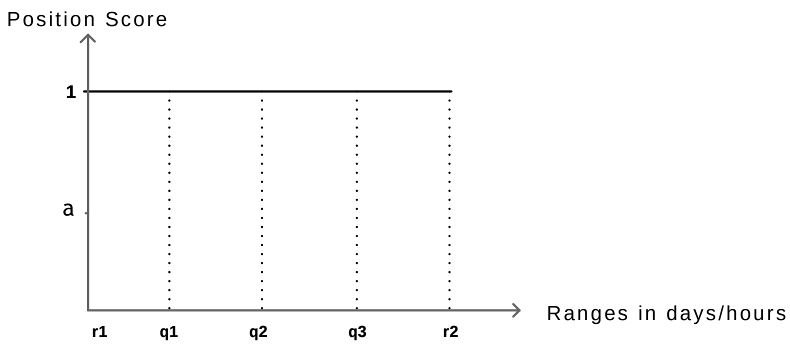


Figure 10 Equally Weighted Positioning Graph

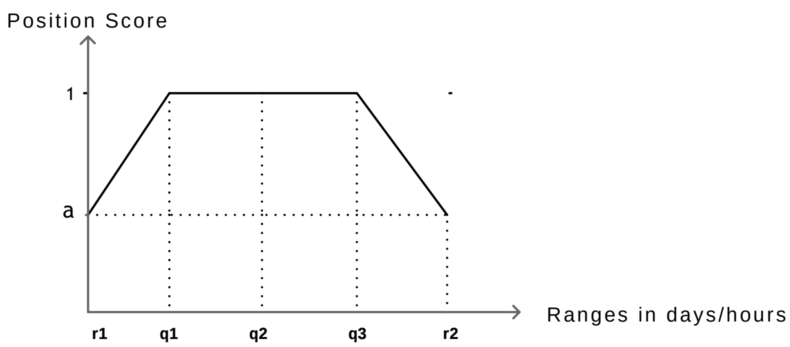


Figure 11 Center Weighted Positioning Graph

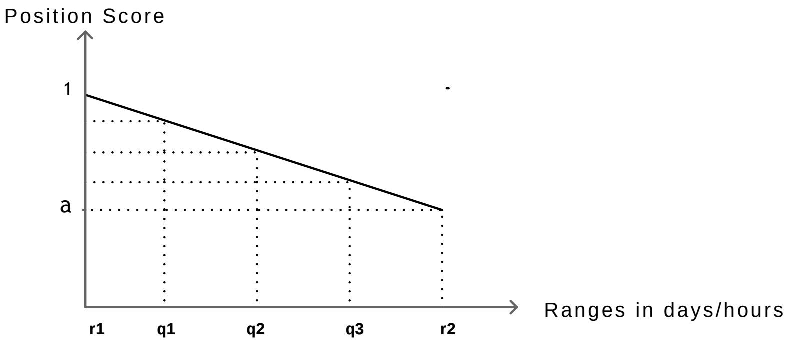


Figure 12 Beginning Weighted Positioning Graph

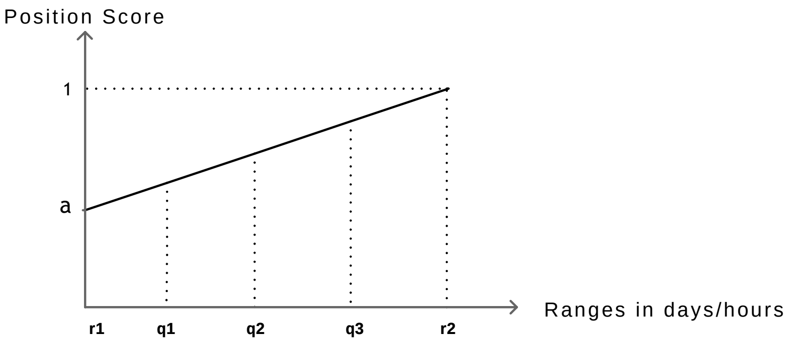


Figure 13 End Weighted Positioning Graph

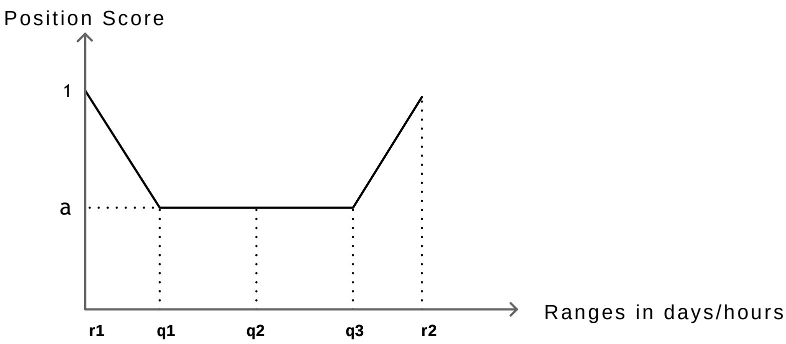


Figure 14 Edge Weighted Positioning Graph

Position score is determined according to the selected option and the position of the query context date and hour values in that range. For example, if a query context date value is between q1 and q3, its position score equals to '1' for the equally weighted and center weighted option and equals to 'a' in edge weighted option.

3.6.2.3. Activity Matching

Although more sophisticated cognitive modeling efforts needed for better definition and matching of activities, here, we will use a simple method towards that direction. The model is very similar to location context and structured as a tree. This time, only two levels exist for the simplicity. More levels could be added to the tree if desired. Rules, constraints and calculations are the same with the location context matching. It is important to repeat the main characteristic of the tree that higher levels must be coarser than the lower ones and should cover them. For example, if higher level activity is 'busy', then the lower one should be something like 'busy-studying' or 'busy-meeting' activities.

However, it is very difficult to define average distance kind of measures and parallel to this similarity measures for two activity context. Thus they are set by the user based on personal views.

3.6.2.4. People Matching

Matching of people context is done by exact match method. It is assumed that there is a number of people available for current context. Context matching for people field is satisfied if and only if specified people exist or does not exist for a certain context. In brief, exact matching is executed between query and target context fields.

In addition to Boolean match a compulsory option is included to provide a mechanism for guarantying non-existence of specified people. An example situation is depicted at Table 4. In that situation an exact match occurs if Ceren and Abdu exist in that context and Selda does not, the state of Mert is not important for the matching operation. If the state of Mert is desired to be true, it becomes a compulsory field.

Table 4 Example Query Structure for People Context

People	Ceren	Abdu	Selda	Mert
State	true	true	false	false
Compulsory	true	true	true	false

People and activity dimensions are included to increase the dimension of contextual information for context matching operation. Thus, the used algorithms for these dimensions are kept simple; and their needs for the more complicated matching algorithms are not addressed.

3.6.2.5. Aggregate Context Matching

In order to obtain an overall matching score for the target context record, weighted mean of the scores for all context dimensions is calculated. If the overall score is greater than a specified threshold, this means that a context matching is reached. Selection of the threshold value depends on the needs of application. If it is selected too low, large set of target context is triggered, on the other hand if it is selected too high, very small set or none of the target context is triggered.

The formula of the total matching score for a certain context pair could be seen in Equation 3:

$$TMS = LS*w_{ls} + TS*w_{ts} + AS*w_{as} + PS*w_{ps} \qquad \textit{Equation 3}$$

Where TMS stands for total matching score, LS location score, TS time score, AS activity score, PS people score. w_{ls} , w_{ts} , w_{as} and w_{ps} denote the weights for each dimension and where $\sum w=1$.

In addition to overall threshold, individual field thresholds can also be defined. In this case, if the one of the individual field threshold is not satisfied, the context matching does not occur; even if total matching score exceeds the overall threshold.

CHAPTER 4

A CASE: CAPRA – CONTEXT-AWARE PERSONAL REMINDER AGENT

In this section, CAPRA- Context-Aware Personal Reminder Agent prototype is introduced. The purpose of this prototype is to show the applicability of proposed context matching algorithm to one of the application areas of context-aware computing, reminder tools. In addition to this, an enhancement to the current reminder systems is provided. Firstly, the current reminder tools are figured out, than we will examine the implementation details of CAPRA.

4.1. Reminders

Reminders are used for informing and remembering some events or to-dos which will be done in the future. They are created by us before the targeted conditions are realized. They have an integral part of our lives.

Signal and description are two main features for the reminders. Signal is responsible from doing something to take our attention and indicating something to be remembered. The description is some written text related to things to be remembered. Current reminder tools use at least one of them in order to remember us what we need to do in the future [77].

Current reminders can be categorized as time based reminders, location/time based reminders and activity/location/time based reminders [90].

Time based reminders are the simplest type of the reminders just using the time dimension of context. Alarm clock is a typical example of this type. When the appointed time arrives, a general audible or vibration alert is produced to attract the user's attention or to get up in the case of bed-side alarm clock which has a signal and implicit description.

Location/time based reminders use two dimensions of context location and time, thus they have more complex mechanisms to gather and process location information. CybreMinder [77] is an example of this type, but it supports the inclusion of different kinds of context information also like door status, activity level and stock amounts.

Finally, the third type of reminders use three dimension of context activity, location and time. Recognizing the current activity of a person from a variety of sensors or one vision based system is quite difficult and error prone. The Memory Glasses project is to produce an activity/location/time based proactive reminder system that is powerful enough to recognize a wide variety of user activities and environmental conditions. It is not a static application that can be trained to recognize new user activities and conditions easily and with a minimum of user intervention [90].

4.2. Desired Characteristics of Reminders

There are a number items desired to be in the current context-aware reminder systems. Dey and Abowd [77] summarizes these as follows:

- the use of rich context for specifying reminders, beyond simple time and location and for proactively determining when to deliver them;
- the ability for users and third parties to submit reminders;
- the ability to create reminders using a variety of input devices;

- the ability to receive reminders using a variety of devices, appropriate to the user's situation;
- the use of reminders that include both a signal indicating something to be remembered and a full description of what is to be remembered; and allowing users to view a list of all active reminders

4.3. CAPRA

CAPRA/ Context-aware Personal Reminder Agent is a prototype application with a mission of showing the applicability of new context-matching algorithm and proposing enhancements to current reminder systems.

CAPRA would be a standalone system to show the applicability of new Context matching algorithm. CAPRA is designed for personal usage. It has the abilities to search previous context with related information and abilities to make future appointments and add reminders based on the contextual elements Location, Time, Activity and nearby People. These context elements are tagged to the reminder texts and triggered if the desired amount of matching between the simulated current context and tagged query context occurs. The system will use an advanced matching algorithm which supports granular/multi-level and best match operations together with relative importance of the fields.

4.3.1. Aim and Scope of Case

CAPRA aims to propose a more advanced reminder system using more contextual elements and a more elaborate matching algorithm.

The objectives of the project are:

- To show the usage of more contextual elements to tag reminders
- To enhance the capability of current reminder tools
- To show the applicability new granular and weighted context-matching algorithm

The case is not intended to propose a comprehensible context model or architecture, and not intended to cover HCI issues of reminder tools. The focus is on the realization of more advanced and sensitive matching operations on context-aware computing systems, and on the utilization of more of the context elements on current reminder tools.

4.3.2. High Level Requirements

High level requirements of CAPRA are:

- System should make use of four dimensions of context - location, time, and activity and nearby people. All four dimensions of context should be tagged to the created reminders.
- System should enable user to select proposed best match or standard Boolean match algorithm.
- System should store a history of context, and enable user to query this context history by using all of the four context elements.
- System should enable a user to configure the matching algorithm.
- System should have a mechanism to pretend current context.
- System should display the outputs sorted according to the relevancy or similarity scores.
- System should enable a user to specify desired preferences and settings: matching defaults and delivery preferences.

4.3.3. System Architecture

In CAPRA, a user centric approach is used. It is a standalone system. User, context simulator and context monitor are the interacting entities of system. Interactions of these entities with the system are depicted in Figure 15. CAPRA takes reminder information, description text and context tag, and query parameters, preference and settings and pretended context information and simulated context information as inputs. After processing of inputs, a matching report, triggered reminders and the signal is produced.

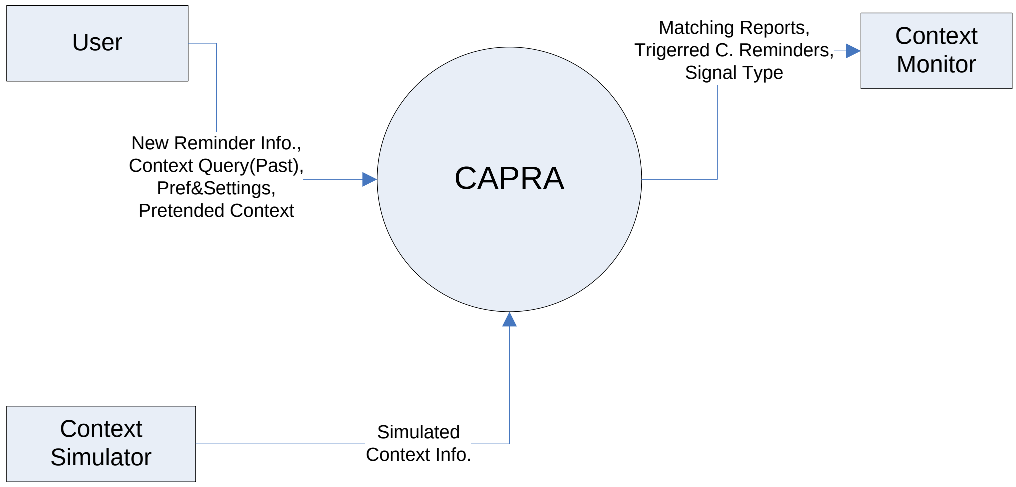


Figure 15 Level - 0 DFD for CAPRA

CAPRA is composed of six sub modules a Context Gathering module, a Context Database module, a Query Manager module, a Matching Engine module, a User Interface module and an Output Generator module. The structure of the system is depicted in Level-0 DFD and Level-1 DFD, see Figures 15 and 16.

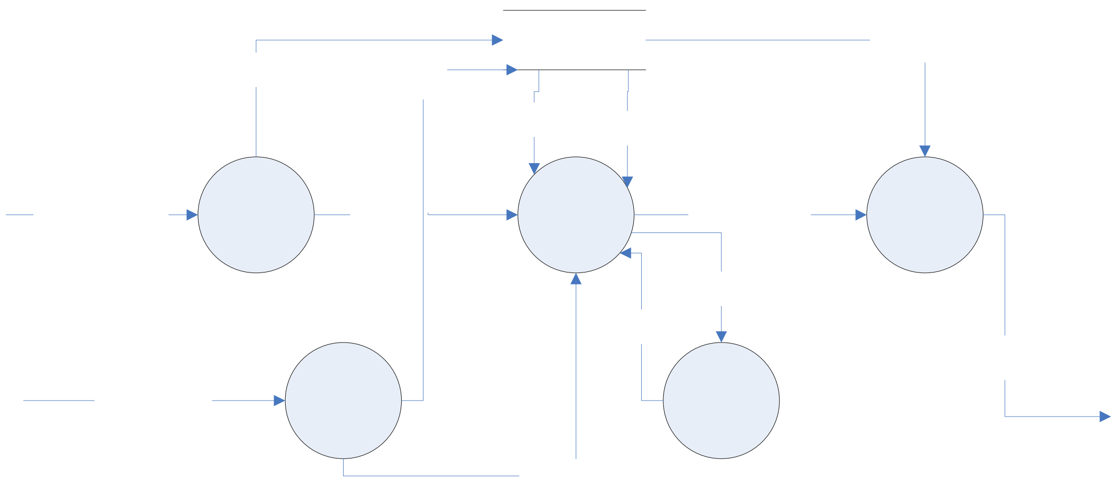


Figure 16 Level - 1 DFD for CAPRA

User Interaction Module is responsible from the provision of following operations: creation of new contextual reminders, creation and update of user profile, pretension of current context, displaying matching reports, triggered contextual reminders and querying past contexts. New reminder information with reminder description and context tag, query and pretended context elements, and user preference and settings are taken as inputs and conveyed to the Context Storage Module and Query Coordinator Module.

Simulated context information is obtained from Context Simulator by the Context Aggregator Module and conveyed to the Query Coordinator module. In addition to this current simulated context is saved to the Context Database Module by Context Aggregator Module.

Query Coordinator Module works as a central management unit. It is responsible from the management of context queries and from the handling of matching process of context query couples. Contextual elements from explicit context queries of user (active context-awareness) are taken and coupled with previously saved context records from Context Storage Module. In addition to this, contextual elements from Contextual Reminders retrieved from Context Storage Module (passive context-awareness) are taken and coupled with the simulated context elements generated from Context Simulator. After this coupling operation, coupled context fields are sent to the Context Matching Engine. The matching report or the alarm signal is released if the matching results coming from the Context Matching Engine are satisfactory.

Context Matching Engine is responsible from the application of proposed context matching algorithm to coupled context packets from the Query Coordinator Modules. It processes them and returns the matching score for each packet to the Query Coordinator Module.

Output Generator Module release the reminders whose matching scores exceeds the overall matching threshold and priority settings appropriate to the user preferences previously defined. It also generates a matching report including the sorted list of successful matches and sends them the context monitor.

Context Storage Module is responsible from the maintenance of user preference and settings, context history and contextual reminders for later retrievals of them.

Main user interface of the program consists of three output screens first for the monitoring current context, the second for the displaying of contextual reminders and the other for the displaying of matching reports. Main user interface screen is depicted in Figure 17.

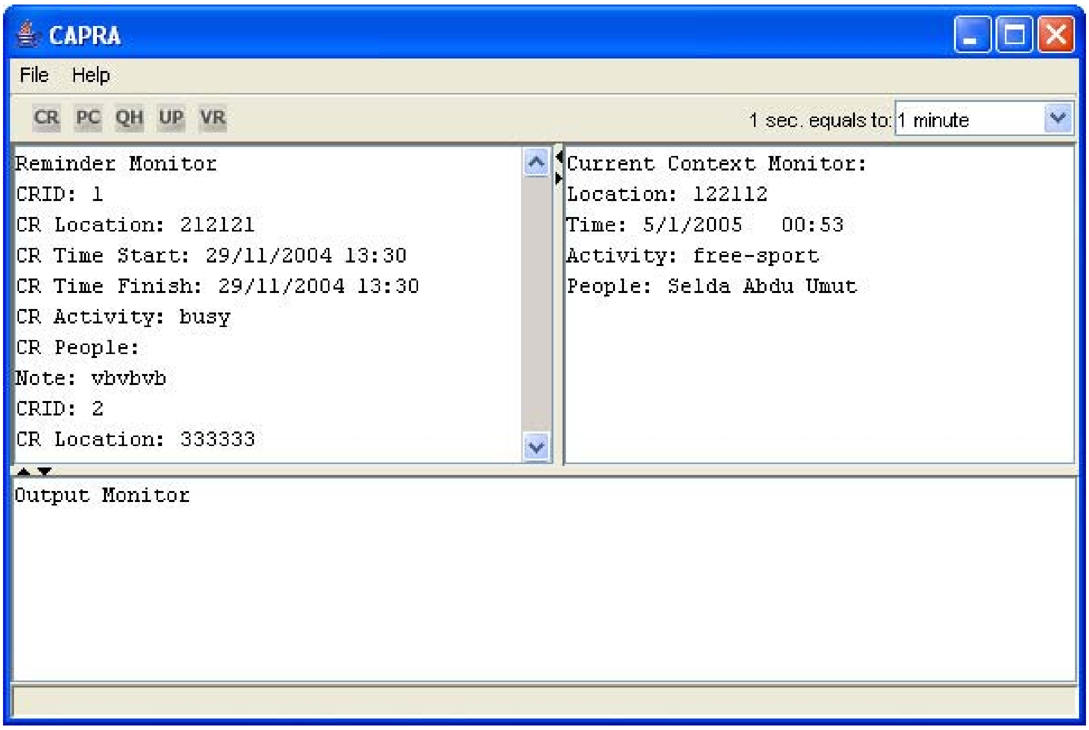


Figure 17 CAPRA Main User Interface

Creation of reminders, pretend context and querying context history operations are realized by the wizard method. Screens of them can be viewed in Figures 18, 19 and 20.

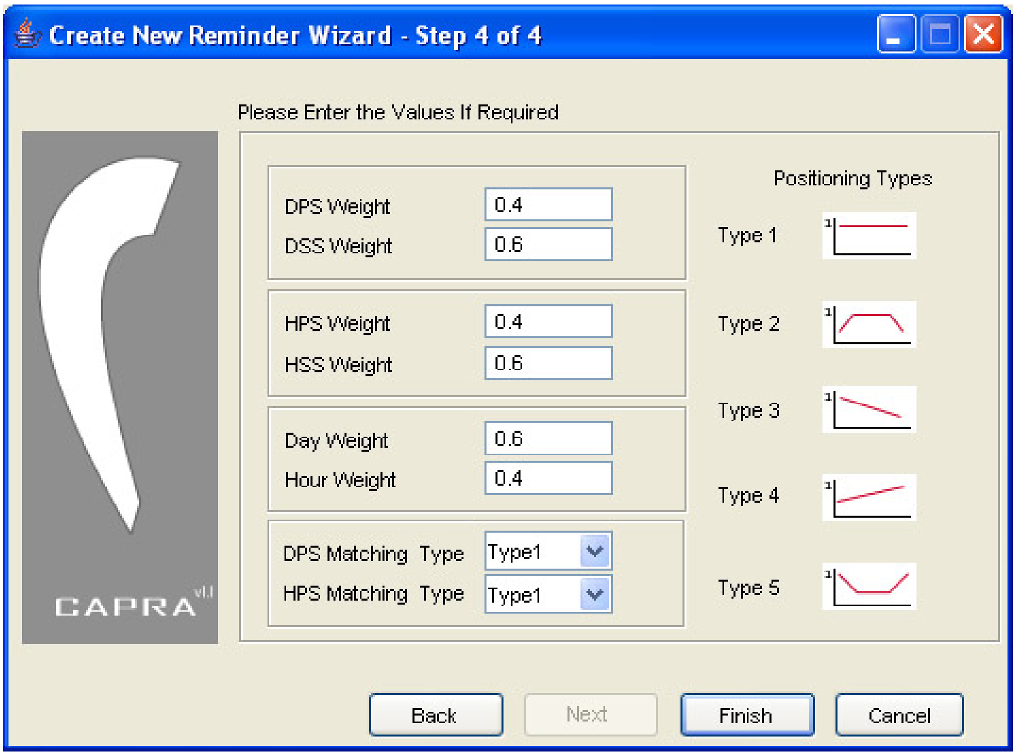


Figure 18 CAPRA Create New Reminder Wizard Screen

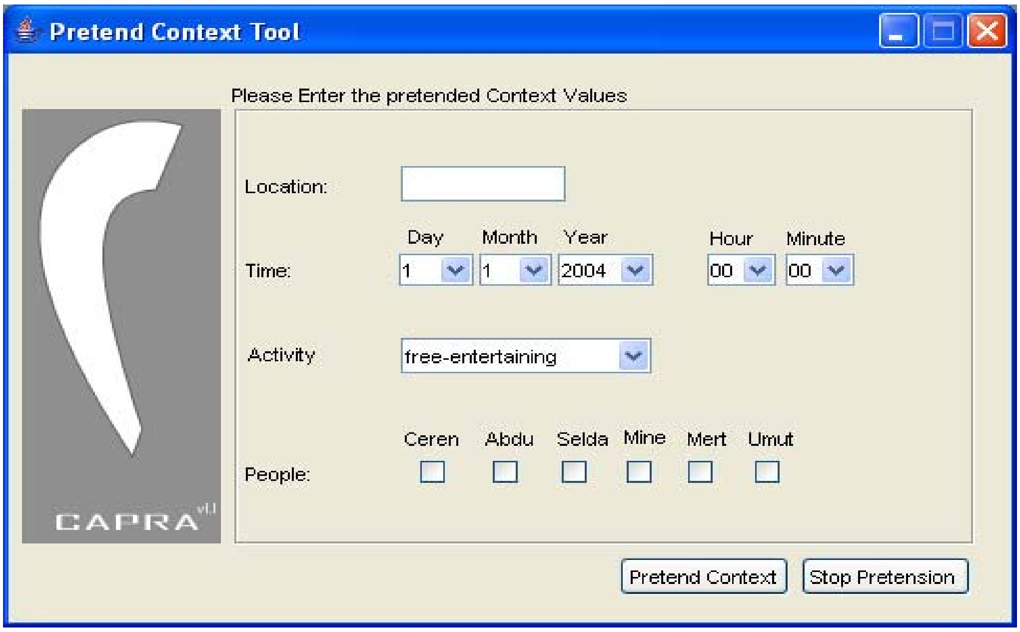


Figure 19 CAPRA Pretend Context Tool Screen

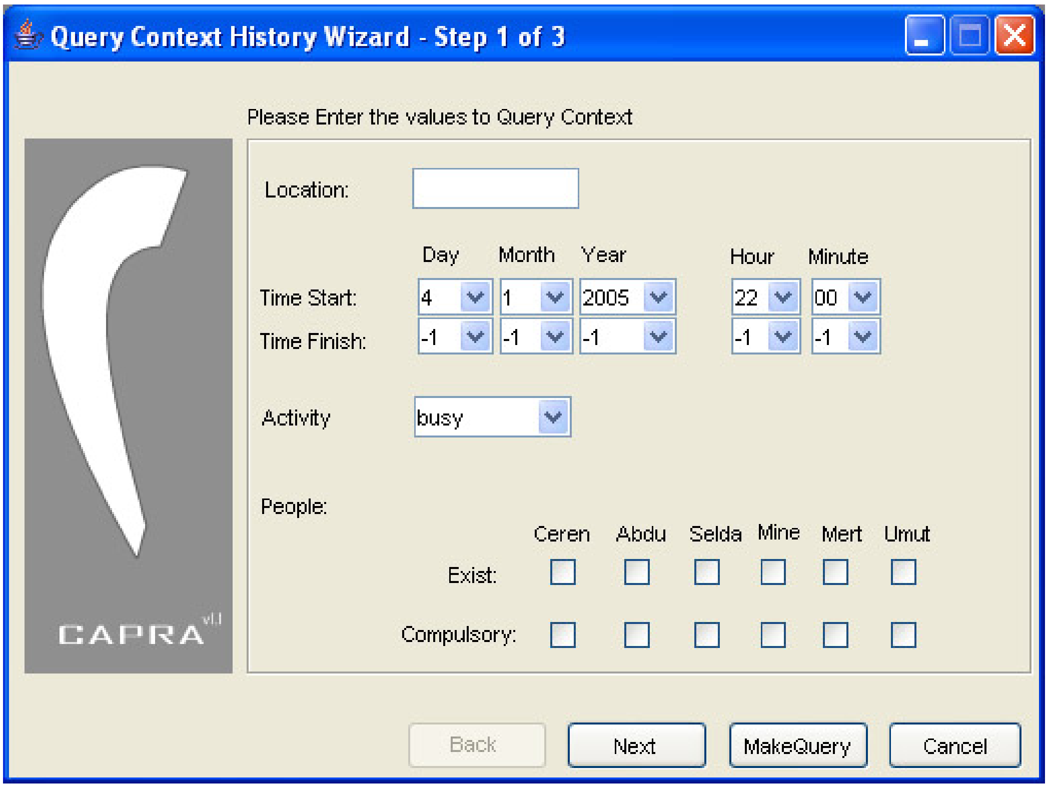


Figure 20 CAPRA Query Context History Wizard

4.4. Typical Scenarios

In this section, typical scenarios for the CAPRA will be explained. The aim of introducing these scenarios is to show the benefits and working dynamics of the proposed algorithm. First scenario is related to context history search, and the second and the last one is related to creation of future reminders with CAPRA.

Scenario 1: Context history search

Suppose that you are looking for a record of conversation in the past. You remember a few things about the context of that conversation.

Natural Language Representation:

"It was 2-3 weeks ago after the lunch, I was in METU with my friends Selda and Mert. We had some free time together."

CAPRA Representation:

Location: METU.

- It corresponds to Neighborhood Level granularity- level 2. It can be seen in Table 2 Similarity and Distance Values According to Granularity Levels.
- Suppose that It maps the symbolic representation in CAPRA as "12".
- Weight for location field could be defined as "0.3" which represents higher than the average importance for four dimensional contexts.
- Threshold for location field could be defined as "0.5" which represents an average matching score. A record with location matching score over this threshold value satisfies the matching criteria for this dimension. If we want to obtain more similar matches for location dimension, we should increase the threshold for location matching score.

Time: 2-3 weeks ago, after the lunch.

- Time dimension in CAPRA is represented as fuzzy or non-fuzzy ranges, and composed of two parts: date and hour (time of day).
- For day, it is defined as dd/mm/yyyy.

  Start of range: 21.10.2004

  End of range: 27.10.2004

  A range of seven days is chosen not to miss the desired record. Defining too small a range may retrieve no results, but defining too large a range may retrieve so many results.

- For hour, it is defines as hh:mm.

  Start of range: 12:00

  End of range: 18:00

  A range of six hours is chosen to meet the query statement - *after lunch*.
- If we chose best match algorithm and fuzzy range option, we have to define DPS weight, DSS weight, HDS weight, HSS weight, DS weight and finally, HS weight; or used the default weights we defined before. However, exact match works well for this scenario for time dimension. Thus, we choose exact match algorithm, and define the weight for overall time dimension which can be "0.3".
- Threshold for time field must be 1.0 since we choose the exact match option.

People: Selda and Mert.

- Simple boolean match is applied for people dimension.
- In order to be a match, a record should include both of the people Selda and Mert. Since no extra information is given for the unwanted people, if there are more people in that context, there will be no negative effect of them for calculating matching score.
- An additional feature for this dimension is to retrieve according to non-existence of people. For example, you might say that Mert should not be in that context record.
- Weight can be defined as 0.3. However, if Selda and Mert rarely come together, the weight of this dimension could be higher than the other context elements. Because, it is the most selective element in that context.
- Threshold must be 1.0, since the exact match is used.

Activity: Free.

- It corresponds to level-1 granularity in the activity tree.
- It maps the symbolic representation in CAPRA as "1".

- Weight for activity field could be defined as “0.1” which represents lower than the average importance for four dimensional contexts. Because it is the least selective element for this scenario.
- Threshold for activity field could be defined as “0.5” which represents an average matching score.

The parameters for the matching process used by CAPRA for scenario 1 can be seen in Table 5.

Table 5 Parameters for Context History Search

Context Dimension	Value	Matching Type	Weight	Threshold
Location	METU->12	Best Match	0.3	0.5
Time	21.10.2004-27.10.2004, 12:00 – 18:00	Boolean Match	0.3	1.0
People	Selda, Mert	Boolean	0.3	1.0
Activity	Free->1	Best match	0.1	0.7
Aggregate	-	Best Match	-	0.7

Calculations for sample situations might be:

Context record #1:

Location: 121222 vs. 12

First two digits – ‘12’ are the same for query and target fields. This corresponds to level-2 similarity. However, since there totally two digits in query field, total satisfaction is reached. Thus it gets full score, i.e 1.0.

It exceeds the specified threshold for location, 0.5.

Time: 22.10.2004, 13:30 vs. 21-27.10.2004, 12:00 – 18:00

Since the simple boolean match is used, the matching score for time equals to 1.0.

It satisfies the specified threshold for time, 1.0.

People: Selda, Abdu, Mert

Selda and Mert exist at the target context, so the matching criterion is fully satisfied and it gets 1.0.

It satisfies the specified threshold for people, 1.0.

Activity: 13 vs. 1

First digits – '1' are the same for query and target fields. This corresponds to level-1 similarity. However, since there totally two digits in query field, total satisfaction is reached. Thus it gets full score, i.e 1.0.

It exceeds the specified threshold for activity, 0.7.

Aggregate Matching: $TMS = LS*w_{ls} + TS*w_{ts} + AS*w_{as} + PS*w_{ps}$

$= 1.0*0.3 + 1.0*0.3 + 1.0*0.3 + 1.0*0.1$

$= 1.0$

It exceeds the specified threshold for overall matching threshold 0.7, and the record will be retrieved.

Info: Images, sounds, documents etc.

Context record #2:

Location: 141222 vs. 12

First one digit – '1' is the same for query and target fields. This corresponds to level-1 similarity. Thus it gets a score of 0.2.

It does not exceed the specified threshold for location, 0.5, so the criterion for location dimension is not satisfied and record will not be retrieved.

Time: Same with the Context record #1.

People: Same with the Context record #1.

Activity: Same with the Context record #1.

Aggregate Matching: $TMS = LS*w_{ls} + TS*w_{ts} + AS*w_{as} + PS*w_{ps}$

$= 0.2*0.3 + 1.0*0.3 + 1.0*0.3 + 1.0*0.1$

$= 0.76$

It exceeds the specified threshold for overall matching threshold 0.7. However, the record will not be retrieved since the individual threshold for location is not satisfied. On the other hand, if we define the threshold of location field as “0.2”, in that case, the record is retrieved.

Info: Images, sounds, documents etc.

Context record #3:

Location: 221132 vs. 12
First digits are not the same for the compared fields. So level-0 similarity score is given as “0.0”.
It does not exceed the specified threshold for location, 0.5, so the criterion for location dimension is not satisfied and record will not be triggered.

Time: Same with the Context record #1.
People: Same with the Context record #1.
Activity: Same with the Context record #1.
Aggregate Matching: $TMS = LS*w_{ls} + TS*w_{ts} + AS*w_{as} + PS*w_{ps}$
$= 0.0*0.3+ 1.0*0.3+1.0*0.3+1.0*0.1$
$= 0.7$

It satisfies the specified threshold for overall matching threshold 0.7. However, the record will not be retrieved since the individual threshold for location is not satisfied.

Scenario 2: Reminder Creation

Suppose that one of your close friends named Ceren has a birthday on 27, January. You do not want to miss it, and want to buy a gift for her birthday.

Natural Language Representation:

"I would like to buy a gift for Ceren to be given at her birthday till 27, January. Inform me after the work when I am not busy, and when I am close to shopping center."

CAPRA Representation:

<u>Location</u>: Shopping Center.

- In order to cover shopping center and the locations near to it, we go up one level in the tree, and It corresponds to Street Level granularity- level 3. It can be seen in Table 2 Similarity and Distance Values According to Granularity Levels.
- Suppose that It maps the symbolic representation in CAPRA as "311".
- Weight for location field could be defined as "0.2" which represents slightly lower than the average importance for four dimensional contexts.
- Threshold for location field could be defined as "0.7" which represents Street Level similarity score. A record with location matching score over this threshold value satisfies the matching criteria for this dimension. If we want to obtain more similar matches for location dimension, we should increase the threshold for location matching score or make the granulation of query location field finer.

<u>Time</u>: Till 27, January, after the work.

- For day:

  Start of range: 5.12.2004

  End of range: 27.12.2004

  A range of 22 days is defined according to the criterion.

- For hour:

  Start of range: 18:00

  End of range: 22:00

  A range of four hours is defined to meet the query statement - *after work*.

- In this case, we choose best match algorithm and fuzzy range option and we have to define DPS and HPS type, and DPS weight, DSS weight, HDS weight, HSS weight, DS weight and finally, HS weight; or used the default weights we defined before. DSS weight is defined as 0.2, since the spread size is not important for this one, but DPS weight is defined as 0.8, which means that when the current date comes closer to January 27, the time field of reminder will get a higher matching score. Parallel to this, HSS weight is defined as 0.2, and HPS as 0.8, but this does not mean that these two always acts similar.

  For DPS type, we choose “end weighted positioning function” which gives higher scores for the query field values closer to the end of the range.

  For HPS type, we choose “equally weighted positioning function”, since the all hours in the range is equally important for us.

  DSS value equals to 0.4 according to the range spread function for date, since the range is greater than 7 days.

  HSS value equals to 0.79 according to the range spread function for hour, since the range of hour is in between three to four hours.

  Finally, HS weight and DS weight is defined as 0.5 representing equal importance and weight for overall TS is defined as 0.4 representing a very high level importance for four dimensional context.

- Threshold for time field might be 0.7 which corresponds to a match over average.

People: Not Ceren.

- Simple boolean match is applied for people dimension.
- In order to be a match, a record should not include Ceren.
- Weight can be defined as 0.3 representing slightly higher than average level of importance for four dimensional contexts.
- Threshold must be 1.0, since the exact match is used.

Activity: Free.

- It corresponds to level-1 granularity in the activity tree.
- It maps the symbolic representation in CAPRA as “1”.
- Weight for activity field could be defined as “0.1” which represents lower than the average importance for four dimensional contexts. Because it is the least selective element for this scenario.
- Threshold for activity field could be defined as “0.5” which represents an average matching score.

The parameters for the matching process used by CAPRA for scenario 2 can be seen in Table 6.

Table 6 Parameters for Context Reminder Query

Context Dimension	Value	Matching Type	Weight	Threshold
Location	Near the shopping center->311	Best Match	0.2	0.7
Time	5.12.2004- 27.12.2004, 18:00 – 22:00	Best Match	W_{TS}:0.4 W_{DS}:0.5 W_{HS}:0.5 W_{DSS}:0.2 W_{DPS}:0.8 W_{HSS}:0.2 W_{HPS}:0.8	0.7
People	Not Ceren	Boolean	0.3	1.0
Activity	Free->1	Best Match	0.1	0.5
Aggregate	-	Best Match	-	0.7

Calculations for sample situations might be:

Pretended Current Context #1:

Location: 313223 vs. 311

First two digits – '12' are the same for query and target fields. This corresponds to level-2 similarity and a score of 0.5.

It does not exceed the specified threshold for location, 0.7.

Thus, the note will not be triggered.

Time: 12.12.2004, 19:30 vs. 5-27.12.2004, 18:00 – 22:00

DPS equals to 1.0 and HPS equals to 1.0 according to fuzzy range function. DSS and HSS were calculated before, so we can find the final time score.

$$TTS = DS*w_{ds} + HS*w_{hs}$$

$$DS = DSS*w_{dss} + DPS*w_{dps} \qquad HS = HSS*w_{hss} + HPS*w_{hps}$$

DS = 0.4*0.2+1.0*0.8 = 0.88

HS = 0.79*0.2+1.0*0.8 = 0.958

TTS = 0.88*0.5+0.958*0.5 = 0.92, which denotes a good match for time.

People: Selda, Umut vs. not Ceren.

Ceren does not exist at the target context, so the matching criterion is fully satisfied and it gets 1.0.

It satisfies the specified threshold for time, 1.0.

Activity: 11 vs. 1

First digits – '1' are the same for query and target fields. This corresponds to level-1 similarity. However, since there totally two digits in query field, total satisfaction is reached. Thus it gets full score, i.e 1.0.

It exceeds the specified threshold for activity, 0.7.

Aggregate Matching: TMS = $LS*w_{ls} + TS*w_{ts} + AS*w_{as} + PS*w_{ps}$

= 0.5*0.2+ 0.92*0.4+1.0*0.3+1.0*0.1

= 0.868

It exceeds the specified threshold for overall matching threshold 0.7. However, the reminder will not be triggered since the individual threshold for location is not satisfied.

Info: Reminder Note: Buy a gift for Ceren.

Pretended Current Context #2:

Location: 311123 vs. 311

First three digits – '311' are the same for query and target fields. This corresponds to level-3 similarity and a score of 0.7.

It satisfies the specified threshold for location, 0.7.

Time: 6.12.2004, 20:30 vs. 5-27.12.2004, 18:00 – 22:00

DPS equals to 0.52 and HPS equals to 1.0 according to fuzzy range function. DSS and HSS were calculated before, so we can find the final time score.

DS = 0.4*0.2+0.52*0.8 = 0.48

HS = 0.79*0.2+1.0*0.8 = 0.958

TTS = 0.48*0.5+0.958*0.5 = 0.719, which exceeds the threshold value for time, but score denotes that the first record has quite better match for time.

People: Same with the Pretended Context #1.

Activity: Same with the Pretended Context #1.

Aggregate Matching: TMS = $LS*w_{ls} + TS*w_{ts} + AS*w_{as} + PS*w_{ps}$

= 0.7*0.2+ 0.719*0.4+1.0*0.3+1.0*0.1

= 0.927

It exceeds the specified threshold for overall matching threshold 0.7. The reminder will be triggered.

Info: Reminder Note: Buy a gift for Ceren.

Pretended Current Context #3:

Location: Same with the Pretended Context #2.

Time: Same with the Pretended Context #2.

People: Same with the Pretended Context #2.

Activity: 21 vs. 1

First digits are not the same for the compared activity fields. So level-0 similarity score is given as “0.0”.

It does not exceed the specified threshold for activity, 0.5, so the criterion for activity dimension is not satisfied and reminder will not be triggered. If we decrease the threshold to the 0, the reminder will be triggered but activity dimension is made inactive or ineffective in the matching process.

Aggregate Matching: $TMS = LS*w_{ls} + TS*w_{ts} + AS*w_{as} + PS*w_{ps}$

$= 0.7*0.2+ 0.719*0.4+1.0*0.3+0.0*0.1$

$= 0.827$

It exceeds the specified threshold for overall matching threshold 0.7. However, the reminder will not be triggered, if the threshold value for activity remains 0.5.

Info: Reminder Note: Buy a gift for Ceren.

Final scores show the similarities of context information pairs, higher scores denote greater similarity. In the case of Scenario 1, the first record has the maximum similarity and the second record has next best similarity and the third record has the least similarity among three records. When we consider the Scenario 2, the first reminder has the maximum similarity and the third reminder has next best similarity and the second reminder has the least similarity among three reminders, if we do not consider the thresholds. Because threshold values are important for triggering the reminders, they do not play a role to determine the similarity. It is a kind of filtering mechanism.

CHAPTER 5

CONCLUSION

5.1. Summary

In this thesis, a research on context-aware computing systems is presented with a special emphasis on context matching. Context-aware computing is examined under the Ubiquitous computing systems.

Notion of context, characteristic of context information and context-aware computing are figured out and architectural and modeling approaches are examined. After giving the details of the context matching operation and context matching issues, our approach to these issues is presented. Then our prototype application CAPRA is introduced.

The problem is the inadequate capabilities of exact match method in context-aware systems. Since context information has some unique characteristics like high dimensionality, high subjectivity and multi granularity; it is needed to have more advanced matching algorithms to satisfy the requirements for context matching.

A weighted granular best matching algorithm is proposed for context matching operations on context-aware computing systems. Matching scores of each context dimension is calculated by the specific algorithms for those contexts and total matching score is obtained from the weighted average of values for all dimensions of context used in comparison. Multi granularity, subjectivity, high dimensionality and fuzziness are the supported characteristics of context information by the proposed matching algorithm.

Finally the applicability of the proposed context matching algorithm is experimented with a case: CAPRA – Context-Aware Personal Reminder Agent which utilizes four dimensions of contexts, location, time, activity and nearby people to tag and query contextual information.

5.2. Discussion

Context matching is used in context-aware computing systems to compare and find the similarity between context pairs. In this thesis study, a weighted granular best matching algorithm is proposed for the context matching. The proposed algorithms in this thesis work are taken from different fields of research independent from the context-aware research and adopted to our study. This algorithm takes into account some important aspects of contextual information in context-aware computing systems by using a granular information structure, weighting mechanism and fuzzy matching methods.

- *High Dimensionality*: Context information might have a multitude of dimensions. Context matching algorithm should properly cope with this high dimensional structure of context information. Our algorithm is capable of managing four-dimensional context information; location, time, activity and people. For each dimension, there are individual matching algorithms with field weights to show the importance of that dimension and field thresholds to define minimum similarity score.

- *Subjectivity*: Characteristics and importance of contextual elements changes dramatically according to the contexts of use, time and according to user. This feature of context information is addressed with the integration of a weighting mechanism (weighted mean) to the aggregation algorithm. By this way, changing importance of context elements by the time could be handled. Higher weights indicate higher importance in matching process. In addition to this, specific algorithms are developed for the location and time dimensions of context. This also satisfies the need for different treatment of each context elements with unique characteristics.

- *Fuzziness*: In order to draw borders or to define the frame of reference for context information is quite difficult. Actually, it seems impossible to know when a certain context starts and when it ends and then another one starts. However, a flexible mechanism might be provided to define and detect contexts. The proposed context matching algorithm realizes this for the time dimension of context by treating the time intervals as fuzzy ranges. Similarity in these fuzzy ranges is calculated by two ways. First score is related to the spread of query context's range of time, and the second one is related to the place of target context's time in the query context's fuzzy range in which different points corresponds different similarity scores. Although the other dimensions of context have fuzzy characteristics, fuzzy matching is applied to only the time dimension in order to show just the usefulness of this approach.

- *Multi-Granularity*: Many of the context elements have a granular structure such as time, activity and location. Location context has six granularity/abstraction levels in our case. Granularity of location information varies from coarser to finer granularity and from district level to room level. Location matching algorithm can compare location information at different granularities according to their average distances for that granularity and their places on the granularity tree.

Our observations show that algorithm used by CAPRA satisfies these requirements, and produces more sensitive results according to the user needs and intention than the simple exact match. Our algorithm has many advantages over the currently used exact match method in context-aware systems. Firstly, it treats each context element differently, by this way different characteristics of the context elements are taken into account and better matching for each context dimension is realized. Secondly, changing importance of context elements are reflected to the matching algorithm by using a weighting mechanism. Higher weights indicate higher importance in context matching. In addition to these, a granular information structure is provided for location and activity dimensions. This granular structure makes it possible to match information at different levels of abstraction. In the case of exact match method, it is not possible to match that kind of information. Fuzzy ranges are used for time dimension and it is possible to assign different importance levels to the points in the range. Exact match method treats the ranges as equally weighted and there is no way to make them. For example, in fuzzy ranges, the importance of a time context might increase when the time comes closer to the end of the range. However, due to the large number of parameters, management of the matching operation might become difficult if the preset values are not used frequently. Thus, an automatic adaptation mechanism that learns from the user and that dynamically adjust parameters is required.

Actually it is not clear that whether each context element should be treated differently or the same. Although it is certain that many of the context element has its unique characteristics and behavior, applying special algorithms for each may not be so feasible when dimensions of context increases over a certain point. Four dimensional contexts are used in our case, and each context dimension was treated differently. However, if we use more than ten or twenty elements of context it will be very difficult to provide specific algorithms for each of them. One solution might be to integrate plug-in algorithms for each dimension. Every context consumer might create its own profile in which they define their desired context elements and corresponding matching algorithms for them. If no definition is provided by the consumers, the default ones might be used. Another solution might be that one multi-dimensional context space model with flexible and comprehensive context ontology could be developed which permits the usage of some exceptional cases for different treatment of desired context elements.

5.3. Future work

Context-aware computing is a relatively very new field of research. So far, research has been mainly application driven and could not achieve introducing standard mechanisms for development and deployment of context-ware systems.

So far Boolean match has been the mostly used matching method in context-aware systems to compare context information. However it is certain that more elaborate and query sensitive algorithms are needed to match contexts. The new proposed algorithm introduces some of the important mechanisms for context matching. However, it is a very initial step and specific to certain representation and there are still many rooms for development.

For location dimension, determination of average distances for each granularity level and parallel to this derivation of granular similarity scores need to be more generic. Efficient data structures and representations for location information should be investigated. Symbolic representation for location is preferred and desired in context-aware computing systems thus research should concentrate more on this direction.

For time dimension, it is realized point vs. range comparison; however in some cases there might be two ranges to compare. Positioning graphs for fuzzy match in our case are static; some evaluation and development studies are needed for these graphs to find optimum fuzzy functions and to find some sort of adaptive behavior.

For activity dimension, it is observed that hierarchical structuring for modeling activity context is useful. However activity context has very challenging characteristics and more cognitive approaches is needed. The most difficult part is to establish the interrelations and dependencies between different activities at the same and different levels of abstraction. These are important to define the required similarity metrics between activity contexts.

For nearby people dimension, exact match seems working well. By the use of compulsory option more of the possible situations are covered. However context sharing between entities and necessary exchange protocols should be defined and privacy concerns should be addressed.

For aggregate matching of context, in addition to the proposed weighted mean operator, applicability and performance of other available aggregation operators should be investigated. According to Brown [82], aggregation operator should calculate the overall matching score according to the specific combination of elements.

In order to declare that a context match has occurred, the overall matching score of the related context should exceed the specified threshold. It is very important to decide the right value for matching threshold. Too low or too high thresholds will probably not be useful for many cases. Some learning mechanism considering the user feedbacks might work.

Although HCI issues are not addressed in this study, they have in central importance for context-aware systems. In order to achieve the goal of effective and unobtrusive use of computing capabilities, HCI issues should be studied with the support of context-awareness.

Finally, necessary evaluation methodologies should be developed for context-aware systems and its sub-fields.

REFERENCES

1. Weiser, M. “The computer for the 21st century”, Scientific American 265, 3, 94–104, 1991

2. Abowd G.D., Mynatt E., “Charting Past, Present, and Future Research in Ubiquitous Computing”, ACM Transactions on Computer-Human Interaction, Vol. 7, No. 1, Pages 29–58, 2000

3. Dey A., Abowd G. D., “Towards a Better Understanding of Context and Context-Awareness”, GVU Technical Report GIT-GVU-00-18, Graphics, Visualization and Usability Center, Georgia Institute of Technology, 2000

4. Want, R., Hopper, A., Falcao, V., Gibbons, J., Goldberg D., Ellis J. R., and Weiser M., “The Active Badge Location System”, ACM Transactions on Information Systems 10(1), pp. 91-102, 1992

5. Want, R., Schilit B. N., Adams N. I., Gold R., Petersen K. , “An Overview of the PARCTAB Ubiquitous Computing Environment”, IEEE Personal Communications, Vol 2, no 6, pp. 28-43, 1995

6. Mitchell K., “Supporting the Development of Mobile Context-Aware Systems”, PhD Thesis, Computing Department, Lancaster University, U.K., 1999.

7. Winograd T., “Architectures for Context”, Human Computer Interaction, Vol. 16, pp401-419, 2001

8. Hong J., Landy A., "An Infrastructure Approach to Context-Aware Computing", Human Computer Interaction, Vol. 16, 2001.

9. Dey A.K., Salber D., Abowd G. D., " A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Computing", Human Computer Interaction (HCI) Journal, Vol. 16, pp.97-166, 2001.

10. Petersen A., Aamodt A., "Case Based Situation Assessment in a Mobile Context-aware System". AmbieSense Project, Department of Computer and Information Science, Norway, 2003

11. Henricksen K., Indulska J., Rakotonirainy A. (2002). Modeling Context Information in Pervasive Computing Systems. Proceedings of Pervasive, Pages: 167 – 180, 2002.

12. Schmidt, A., Beigl, M., Gellersen, H.-W., "There is more to context than location", Computer & Graphics 23(6), pp. 893-901, December 1999.

13. Korpipää P., Mäntyjärvi J., Kela J., Keränen H., Malm E. J., "Managing Context Information in Mobile Devices", IEEE Pervasive computing, Vol. 2, No. 3, pp. 42-51, 2003.

14. Doğac, A., Laleci, G., Kabak, Y., "Context Frameworks for Ambient Intelligence", eChallenges 2003, Bologna, Italy, 2003.

15. Ranganathan A., Campbell R. H., "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments", In Proc. of International Middleware Conference, 2003.

16. Schmidt, A., "Ubiquitous Computing – Computing in Context", PhD Thesis, Computing Department, Lancaster University, U.K., 2002

17. M. Satyanarayanan, "Pervasive computing: Vision and challenges", IEEE Personal Communications, vol. 8, pp. 10–17, 2001.

18. Abowd G. Iachello G., "Security Requirements for Environmental Sensing Technology", GVU Technical Report, Graphics, Visualization and Usability Center, Georgia Institute of Technology, 2002.

19. Van Laerhoven K., Aidoo K., Lowette S., "Real-time Analysis of Data from Many Sensors with Neural Networks", Proc. of the International Symposium on Wearable Computing, Zurich, Switzerland, 2001.

20. Van Laerhoven, K., "Online Adaptive Context Awareness", Licentiate thesis at the University of Brussels (VUB), Brussels, 1999.

21. Mayrhofer R., Radi H., Ferscha A., "Recognizing and Predicting Context", Institut für Praktische Informatik, Johannes Kepler University Linz Austria, 2004.

22. Flanagan A. J., Himberg J., Mäntyjärvi J., "A Hierarchical Approach to Learning Context and Facilitating User Interaction in Mobile Devices" Artificial Intelligence in Mobile System, 2003.

23. Wu H., Siegel M., Stiefelhagen R., Yang J., "Sensor Fusion Using Dempster-Shafer Theory", IEEE International Measurement Technology Conference (IMTC) 2002.

24. Chen G., Kotz D., "A Survey of Context-aware Mobile Computing Research", Technical Report TR2000-381, Department of Computer Science Dartmouth College, 2000.

25. Weiser M., Brown J.S., "The Coming age of Calm Technology", Technical Report, 1996.

26. Weiser M., Gold R., Brown J. S., "The origins of ubiquitous computing research at PARC in the late 1980s", IBM Systems Journal, Pervasive Computing, Volume 38, Number 4, 1999

27. Saha D., Mukherjee A., "Pervasive Computing: A Paradigm for the 21st Century", The IEEE Computer Society, 2003.

28. Satyanarayanan, M., "Fundamental challenges in mobile computing", In Fifteenth ACM Symposium on Principles of Distributed Computing, Philadelphia, Pennsylvania, 1996.

29. Prekop P., Burnett M., "Activities, Context and Ubiquitous Computing", Computer Communications, Special Issue on Ubiquitous Computing, 2002.

30. Azzopardi L., "Finding Order in the Chaos of Context: Towards a Principled approach to Contextual Information Retrieval", Technical Report, Department of Computer Science, University of Glasgow, 2002.

31. Brézillon P., "Context in Artificial Intelligence: II. Key elements of contexts", Computer & Artificial Intelligence 18(5): 425-446, 1999.

32. Brézillon P., "Context in Artificial Intelligence: I. A survey of the literature", Computer & Artificial Intelligence, 18(4): 321-340, 1999.

33. Schilit, B., Adams, N. Want, R., "Context-Aware Computing Applications" 1st International Workshop on Mobile Computing Systems and Applications, 85-90, 1994.

34. Akman V, Surav M., "Steps toward Formalizing Context", AI Magazine 17, 55-72, 1996.

35. Pascoe, J., Ryan, N., Morse, D., "Issues in Developing Context-Aware Computing", HUC'99: Springer-Verlag, 1999.

36. Korkea-aho M., "Context-Aware Applications Survey", Department of Computer Science Helsinki University of Technology, 2000. http://www.hut.fi/~mkorkeaa/doc/context-aware.html (Last visited on 25.5.2005)

37. Pascoe J., "Context-aware Software", PhD Thesis, Computer Science Department, The University of Kent at Canterbury, 2001.

38. Held, A., Buchholz, S., Schill, A., "Modeling of Context Information for Pervasive Computing Applications", Proc. of the 6th World Multi Conference on Systemics, Cybernetics and Informatics (SC12002), Orlando, FL, USA, 14-18, 2002.

39. Dey, A., Salber, D., Abowd, G., "A context-based infrastructure for smart environments", In 1st International Workshop on Managing Interactions in Smart Environments, 1999.

40. Schmidt, A., Aidoo K. A., Takaluoma A., Tuomela U., Laerhoven K. V., Velde W. V., "Advanced Interaction In Context", In 1st International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, 1999.

41. H.-W. Gellersen, A. Schmidt, M. Beigl., "Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts, Mobile Networks and Applications", MONET, 341-351, Kluwer, 2002.

42. Salber, D., Dey A., Abowd G. D., "The Context Toolkit: Aiding the Development of Context-Enabled Applications", In Proceedings of CHI'99, pp. 434-441. 1999.

43. Schmidt A., Aidoo K. A., Takaluoma A., Tuomela U., Laerhoven K. V., Velde W., "Advanced Interaction In Context", In Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99, pages 89-101, Karlsruhe, Germany, 1999.

44. Dey, A., Abowd D., Wood A., "CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services", IUI'98,47-54, 1998.

45. Schmidt, A., Winterhalter, C., "User Context Aware Delivery of E-Learning Material: Approach and Architecture (Extended Version)", In: Journal of

Universal Computer Science (JUCS), Special Issue on Human Issues and Personalization in E-Learning, 2004.

46. Jansson M., "Context Shadow: An Infrastructure for Context AwareComputing", Proceedings of Artificial Intelligence in Mobile System, Lyon, France, 2002. http://www.dsv.su.se/feel/DSV/ContextShadow.pdf (Last visited on 25.5.2005)

47. Malm E. J., Kaartinen J., Vildjiounaite E., Alahuhta P., "Smart-It Context Architecture", Technical Report E1SU00075, Interconnected Embedded Technology for Smart Artefacts with Collective Awareness Project, 2003.

48. Brown, P.J., "The stick-e Document: a Framework for Creating Context-aware Applications", Electronic Publishing '96, pp. 259-272, 1996.

49. Roman M., Campbell H. R., "A User-Centric, Resource-Aware, Context-Sensitive, Multi-Device Application Framework for Ubiquitous Computing Environments", Technical Report UIUCDCS-R-2002-2284 UILU-ENG-2002-172, 2002.

50. Covington. J. M. et al., Zhan Z., Ahamad M., "A Context-Aware Security Architecture for Emerging Applications", In Proceedings of 18th Annual Computer Security Applications Conference, 2002.

51. Want, R., Hopper, A., Falcao, V., Gibbons, J., "The Active Badge Location System", ACM Transactions on Information Systems 10(1) pp 91-102, 1992.

52. Farringdon, J., Moore, A., Tilbury, N., Church, J., Biemond, P., "Wearable sensor badge & sensor jacket for context awareness", In Proc. Third International Symposium on Wearable Computers (ISWC), 107--11, 1999.

53. Nakanishi, Y., Tsuji, T., Ohyama, M., Hakozaki, K., "Context Aware Messaging Service: a Dynamical Messaging Delivery using Location

Information and Schedule Information", Journal of Personal Technologies, Vol.4, No.4, pp.221-224, 2000.

54. Davies N., *Cheverst K., Mitchell K., Efrat A.*, "Using and Determining Location in a Context-Sensitive Tour Guide", IEEE Computer Society Press, Volume 34 , Issue 8, 2001.

55. Hull R., Neaves P., and Roberts J.E., "Towards Situated Computing", IEEE Computer Society Press, In Proceedings of the First International Symposium on Wearable Computers, Cambridge, Massachusetts, 1997.

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

57. Want R., Schilit B. N., Adams N. I., Gold R., Petersen K., Goldberg D., R. J. Ellis, and Weiser M., "The ParcTab Ubiquitous Computing Experiment", In Tomasz Imielinski and Henry F. Korth, Mobile Computing, ch. 2, 1996.

58. Ward A., Jones A., and Hopper A., "A New Location Technique for the Active Office", IEEE Personal Communications, 4(5):42-47, 1997.

59. Addlesee M. D., Jones A., Livesey F., and Samaria F., "The ORL Active Floor", IEEE Personal Communications, 4(5):35-41, 1997.

60. Orr J.R. and Abowd G. D., "The Smart Floor: A Mechanism for Natural User Identification and Tracking", ACM Press, In Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), The Hague, Netherlands, 2000.

61. Stillman S., Tanawongsuwan R., and Essa I., "A System for Tracking and Recognizing Multiple People with Multiple Cameras", In Proceedings of Second International Conference on Audio-Vision based Person Authentication, Washington, DC, 1999.

62. Harter A., Hopper A., Steggles P., Ward A., Webster P., "The anatomy of a context-aware application", In Mobile Computing and Networking, pages 59--68, 1999.

63. Castro P. and Muntz R., "Managing Context Data for Smart Spaces", IEEE Personal Communications, 7(5):44-46, 2000.

64. Byun H. E., Cheverst K., "Harnessing Context to Support Proactive Behaviors", Technical Report, Department of Computing Lancaster University, 2002. http://www.comp.lancs.ac.uk/computing/research/mpg/projects/cortex/archive/dmrg%20publications/Byun-AIMS2002.pdf. (Last visited on 25.1.2005)

65. Laerhoven K. Van and Cakmakci O., "What Shall We Teach Ur Pants?", In Proceedings of the fourth International Symposium on Wearable Computers, Atlanta, IEEE Press, pp.77-83, 2000.

66. Costa P. D., "Towards a Services Platform for Context-Aware Applications", MSc. Thesis, Telematics from the University of Twente, The Netherlands, 2003.

67. Schilit W. N., "A System Architecture for Context-Aware Mobile Computing", PhD. Thesis, the Graduate School of Arts and Sciences, Columbia University, 1995.

68. Pascoe, J., "Adding Generic Contextual Capabilities to Wearable Computers", 2nd International Symposium on Wearable Computers, 92-99, 1998.

69. Brown P., Burleson W., Lamming M., Rahlff O., Romano G., Scholtz J., and Snowdon D., "Context-awareness: Some Compelling Applications", 2nd International Symposium on Handheld and Ubiquitous Computing, 2000.

70. Davies, N., Mitchell, K., Cheverest, K., Blair, G., "Developing a Context Sensitive Tourist Guide", First Workshop on Human Computer Interaction with Mobile Devices, GIST Technical Report G98-1, 1998.

71. Yang J., Yang W., Denecke M. and Waibel A., "Smart Sight: A Tourist Assistant System", 3rd International Symposium on Wearable Computers, San Francisco, California, pp. 73-78, 1999.

72. Dey, A.K., Salber, D., Abowd, G.D., Futakawa, M., "The Conference Assistant: Combining Context-Awareness with Wearable Computing", 3rd International Symposium on Wearable Computers, San Francisco, California, pp. 21-28, 1999.

73. Hull, R., Neaves P. and Bedford-Roberts J., "Towards Situated Computing", 1st International Symposium on Wearable Computers, Cambridge, Massachusetts, pp. 146-153, 1997.

74. M. G. Lamming, Brown P. J., Carter K., Eldridge M., Flynn M., Robinson P. and Sellen A., "The design of a human memory prosthesis", Computer Journal 37 (3), pp. 153-163, 1994.

75. M. G. Lamming, Flynn M., "Forget-me-not: Intimate Computing in Support of Human Memory", Technical Report EPC-1994-10, Proceedings of FRIEND21, '94 International Symposium on Next Generation Human Interface, Meguro Gajoen, Japan, 1994.

76. Rhodes B. J., "Remembrance Agent: A Continuously Running Automated Information Retrieval System", In The Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology, pp. 487-495, 1996.

77. Dey A.K. and Abowd G.D., "CybreMinder: A Context-Aware System for Supporting Reminders", In Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, Springer Verlag, Bristol, UK, 2000.

78. G.G. Abowd, C.G. Atkeson, J. Hong, S. Long, R. Kooper and M. Pinkerton, “Cyberguide: A Mobile Context-Aware Tour Guide”, ACM Wireless Networks, 3, pp. 421-433, 1997.

79. Smailagic A., Siewiorek D.P., Anhalt J., Gemperle F., “Towards Context Aware Computing: Experiences and Lessons”, IEEE Journal on Intelligent Systems, Vol. 16, No. 3, pp 38-46, 2001

80. Brown P.J., and Jones G.J.F., “Exploiting Contextual Change in Context-Aware Retrieval”, Proceedings of the 17th ACM Symposium on Applied Computing, 2002.

81. Jones, GJF., Brown, PJ., “Context-Aware Retrieval for Pervasive Computing Environments”, Proc First International Conference on Pervasive Computing, Zurich, Switzerland, pp. 10—27, 2002.

82. Page L., Brin S., Motwani R., Winograd T., “The PageRank Citation Ranking: Bringing Order to the Web”, Stanford Digital Library Technologies Project, 1998.

83. Makkonen J., Ahonen-Myka H., and Salmenkivi, M., “Topic Detection and Tracking with Spatio-Temporal Evidence”, In Proceedings of the 25th European Conference on IR Research (ECIR 2003), Pisa, Italy, pp. 251-265, 2003.

84. Tauritz D. R., “Adaptive Information Filtering: concepts and algorithms”, PhD. Thesis, Leiden Institute of Advanced Computer Science, 1973.

85. Belkin N. J. and Croft W. B., "Information Filtering and Information Retrieval: Two Sides of the Same Coin?", Communications of the ACM, 35(12):29–38, 1992.

86. Rijsbergen C.J. van, "Information Retrieval", Butterworths, second edition, London, 1979.

87. Salton G., "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer", Addison-Wesley, Pennsylvania 1989.

88. Cooper W., "The Formalism of Probability Theory in Ir: A Foundation Or An Encumbrance?", In Proceedings Of The Seventeenth Annual International Acm-Sigir Conference On Research and Development In Information Retrieval, Pages 242–247, New York, Ny, Usa, 1994.

89. Robertson, S.E. "The probability ranking principle in IR". Journal of Documentation 33, 4, 294-304. 1977.

90. DeVaul R. W., Clarkson B., Pentland S., "The Memory Glasses: Towards a Wearable, Context Aware, Situation-appropriate Reminder System", Technical Report, MIT Media Laboratory, 2000. http://web.media.mit.edu/~rich/memory/SIUC00/Main.html (Last visited 15.1.2005)

91. Rhodes, B.J., "The wearable remembrance agent: a system for augmented memory", Personal Technologies, 1(1):218-224, 1997.

92. Goslar K., Buchholz S. and Schill A., "A Multidimensional Approach to Context-Awareness", Technical Report, Department of Computer Science, Dresden University of Technology, Germany, 2003

APPENDICES

APPENDIX A

SOFTWARE REQUIREMENTS SPECIFICATION

The Software Requirements Specification document is available on the CD (Appendix_A_SRS.pdf).

APPENDIX B

SOFTWARE DESIGN DESCRIPTION

The Software Design Description document is available on the CD (Appendix_B_SDD.pdf).