

NOTE TO USERS

Page(s) not included in the original manuscript are unavailable from the author or university. The manuscript was microfilmed as received

iv & v

This reproduction is the best copy available.

UMI[®]



uOttawa

L'Université canadienne
Canada's university

**FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTDOCTORALES**



**FACULTY OF GRADUATE AND
POSTDOCTORAL STUDIES**

Mohammad Anwar Hossain

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical and Computer Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Support of Quality of Context Aware Interaction in Pervasive Environments

TITRE DE LA THÈSE / TITLE OF THESIS

Abudlmotaleb El Saddik

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Dorina Petriu

Shervin Shirmohammadi

**Qi Tian (University of Texas at
Antonio)**

Jiying Zhao

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Support for Quality of Context Aware Interaction in Pervasive Environments

by

Mohammad Anwar Hossain

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Electrical and Computer Engineering

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

© Mohammad Anwar Hossain, Ottawa, Canada, 2010



Library and Archives
Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-69133-5
Our file *Notre référence*
ISBN: 978-0-494-69133-5

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Abstract

Interactions with the pervasive intelligence environment are increasingly becoming complex due to the presence of myriad sensors, artifacts and emerging services. Therefore, users need support to interact, control, and manage these services in their environment in an effective manner. One way to deal with this situation is to automatically identify their surrounding context and provide them context-aware services in order to reduce their cognitive load. Providing automatic support to users is known as implicit interaction, which minimizes the need for explicit command or interaction. However, as the sensor-based context identification often provides imprecise information, the automation based on such information may result in wrong actions or over automation by the intelligent environment. Therefore, the selection of inappropriate services while providing automatic support in the environment brings distrust and dissatisfaction to the people. In this thesis, we provide a framework for human-environment interactions that considers quality of context information to dynamically change the level of implicit interaction performed by the environment. Our goal is to provide a quality of context aware adaptive interaction mechanism in order to minimize the distrust and dissatisfaction of users on context-adaptive automated environment. Toward this, we first provide a model to dynamically assess the quality of context information. Based on such quality, our framework adjusts the level of output automation that the environment performs. Finally, according to the adjusted level of interaction, a set of services are either suggested or invoked for the users based on the gain in those services and their associated cost. Our experiments show that dynamically adjusting the level of interaction based on quality of context information increases the satisfaction and trust of the people in the implicit actions performed by the intelligent environment.

Acknowledgements

First and foremost, I would like to acknowledge my sincere gratitude and indebtedness to my Ph.D. supervisor, Prof. Abdulmotaleb El Saddik, for the guidance and support I have received from him during the last four years. His research vision, enthusiasm and confidence have significantly improved my research capability and kept me motivated toward the completion of this thesis. I sincerely thank him for the many discussions that helped me consolidate on my ideas. I also thank him for providing a supportive research environment that encourages everyone in MCRLab research group to aim for innovation and novelty. I hope I will continue getting moral and professional support from him in the years to come.

During my thesis, I had the opportunity to collaborate with Dr. Pradeep K. Atrey. I greatly acknowledge his help for his valuable comments and suggestions. I also sincerely acknowledge the help of Dr. Ali Nazari for his valuable inputs while writing my thesis. My sincere thanks also goes to Jorge Parra, a visiting researcher from IKERLAN, for his help in implementing a working prototype of the framework presented in this thesis.

This thesis would not have been completed without the constant support, inspiration and sacrifice of my lovely wife. I thank her for the understanding and patience. I thank my two sweet daughters for the time they were deprived of me. I sincerely thank my parents for their constant prayer. I am also grateful to all my friends with whom I discussed many aspects of this thesis. My special thanks go to Anis, Atif, Kafi, Mehedi, Mizan, and Shamim.

I thankfully acknowledge the financial assistance of the Natural Sciences and Engineering Research Council Canada (NSERC) and IBM Canada.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Existing Interaction Approaches and Limitations	2
1.2.1	Lack of Model to Assess Quality of Context Dynamically	3
1.2.2	Missing Ability to React to Changing Quality of Context	3
1.2.3	Lack of Cost-optimal Selection of Services	4
1.3	Research Goal	4
1.4	Proposed Approach: Quality of Context Aware Interaction Framework	5
1.5	Thesis Statement and Contribution	8
1.6	Publications	8
1.7	Organization of the Thesis	12
2	Related Work	14
2.1	Human-Environment Interaction	14
2.2	Quality of Sensor-driven Information	18
2.3	Service Provision in Ambient Environment	22
2.4	Summary	25
3	Modeling Quality of Context Information	28
3.1	Requirements for Quality of Context	28
3.2	QoI Problem Specification	31
3.3	Detecting Information Items	32
3.4	Modeling Quality Attributes	34
3.4.1	Modeling Certainty	34

3.4.2	Modeling Accuracy	37
3.4.3	Modeling Timeliness	40
3.4.4	Contextual Effect in QoI Modeling	41
3.4.5	Aggregating Quality Attributes	44
3.5	Experiments for QoI Assessment	46
3.5.1	System Setup	47
3.5.2	System Implementation	48
3.5.3	Information Item Processing	49
3.5.3.1	Sensor Media Processing	50
3.5.3.2	Sensor Observation Fusion	52
3.5.4	Dynamic QoI Assessment and Result Analysis	54
3.5.4.1	Certainty Assessment	55
3.5.4.2	Confidence Assessment	57
3.5.4.3	Timeliness Assessment	58
3.5.4.4	Context Effect Analysis	59
3.5.4.5	QoI Attribute Aggregation	59
3.5.4.6	Result Verification	60
3.5.5	Observations based on QoI assessment	64
3.6	Summary	65
4	QoI-based Dynamic Levels of Interaction	67
4.1	Different Levels of Interaction	67
4.1.1	Input Automation	69
4.1.2	Output Automation	70
4.2	QoI based Levels of Interaction	71
4.3	Summary	74
5	Context-aware Service Provisioning	75
5.1	Service Provisioning Problem Specification	75
5.2	Dynamic Gain Estimation	77
5.2.1	User's Context	77
5.2.2	User's Profile	79

5.2.3	Interaction History	81
5.2.4	Media Reputation	81
5.2.5	AMP Update Procedure	82
5.2.6	Gain Estimation	84
5.3	Media Service Selection	86
5.3.1	Gain Fusion	87
5.3.2	Cost Computation	87
5.3.3	Selection of a Single Service	88
5.3.4	Selection of Multiple Services	89
5.3.5	Overall Subset of Services	90
5.4	Experiments with Service Selection	90
5.4.1	Sample Implementation	91
5.4.2	Gain Computation	91
5.4.3	Gain-cost based Service Selection	93
5.5	Summary	96
6	Framework Integration	98
6.1	High-level Architecture	98
6.2	Component Communication for Interaction	101
6.2.1	Explicit Interaction Support	103
6.2.2	Implicit Interaction Support	103
6.2.3	Mixed-initiative Interaction Support	103
6.3	Implementation	106
6.3.1	Environment Setup	106
6.3.2	Development of Different Component Services	106
6.3.2.1	Context Identifier WS	109
6.3.2.2	AMP Manager WS	110
6.3.2.3	Media Source Manager WS	110
6.3.2.4	Interaction Manager WS	110
6.3.2.5	Gain Manager WS	110
6.3.2.6	Home Automation WS	111

6.3.2.7	Media Renderer WS	111
6.3.2.8	Ambient Controller WS	111
6.3.2.9	Ambient Media Selector WS	111
6.3.2.10	Ambient Media Selector GUI WS	114
6.4	Summary	115
7	Evaluation	117
7.1	Experiment Details	117
7.1.1	Environment Setup and Scenario	117
7.1.2	Test Procedure	119
7.1.3	Data Collection	121
7.2	Result Analysis	123
7.2.1	User Acceptance Measure	123
7.2.2	Trust Measure	127
7.3	Summary	128
8	Conclusion and Future Work	129
8.1	Conclusion	129
8.2	Future Work	130

List of Tables

2.1	A summary of QoI-related literature	19
2.2	Comparison between the existing and proposed interaction approaches . .	25
3.1	Example context-dependent rules that influence the computation of QoI .	43
3.2	Identification of information items based on different sensor media streams	51
3.3	List of sensors and corresponding processing tasks to determine I_1 and I_2	53
3.4	An example of the dynamic QoI assessment	55
3.5	Aggregation of quality attribute values	59
4.1	Mapping Rules between QoI and Level of Output	73
5.1	The different factors contributing to gain estimation	77
5.2	Data set used in the experiment showing different services with cost . . .	94
6.1	The devices/resources in the experimental smart environment.	107
7.1	Expected relationship between UC-AM variables	119
7.2	Demographic distribution of the test participants	121
7.3	Questionnaire used for feedback collection regarding acceptance	122
7.4	Questionnaire used for feedback collection regarding trust	123
7.5	User acceptance based on the overall evaluation	126

List of Figures

1.1	Four aspects of the proposed work	6
3.1	Context information identification in multi-sensor system	33
3.2	The sliding sampling window for measuring sensor observation	39
3.3	An example of environmental context effect in QoI	42
3.4	QoI aggregation at information item level and quality attribute level	45
3.5	Layout of the simulated smart home experimental environment.	47
3.6	Architecture of the prototype for QoI assessment.	48
3.7	UI snapshot of the running system.	50
3.8	The observation scores from the individual media processing tasks	53
3.9	Number of sensors and value of the QoI attributes with time-varying mean	57
3.10	The verification of dynamically evolved confidence value	61
4.1	Level of automation	68
4.2	QoI and Context adaptive interaction system	71
4.3	QoI-based dynamic level of automation	72
5.1	The different context parameters	78
5.2	The AMP structure of a user in a particular context	80
5.3	Example of movie service usage pattern by a user	82
5.4	Gain of a media service based on given AMP	85
5.5	Schematic view of the process of optimal service selection	89
5.6	Changes in AMP scores after movie service usage	92
5.7	Example gain evaluation of different services	93
5.8	The set of selected services with different cost constraints	95

5.9	Service provided via a smart mirror interface	96
6.1	Architecture of the proposed context-aware interaction framework	99
6.2	Sequence of interactions in explicit mode	102
6.3	Sequence of interactions in implicit mode	104
6.4	Sequence of interactions in mixed-initiative mode	105
6.5	The layout of the experimental environment.	107
6.6	The interconnection among the different components.	109
6.7	Code snippet of the four web methods in the lighting control web service.	112
6.8	Code snippet of the two web methods in the media renderer web service.	113
6.9	Example output interface when system runs in full automation mode.	114
6.10	The user interface of the prototype system on a real HTC phone.	115
7.1	User response percentage on the usage of System-A.	124
7.2	User response percentage on the usage of System-B.	125
7.3	Comparison of mean responses between age groups for System-A.	125
7.4	Comparison of mean responses between age groups for System-B.	126
7.5	Comparison of mean responses on System-A and System-B.	127
7.6	Trust measures of System-A and System-B.	128

Chapter 1

Introduction

1.1 Motivation

The paradigm of Pervasive Computing, Ubiquitous Communication and Ambient Intelligence all aim to realize a future where humans are surrounded by sensors, actuators and emerging services [5, 6, 38]. Such visions are being realized by many projects, for example, AwareHome [65], GatorTech Smart Home [48], EasyLiving [28], and MavHome [33]. The overall goal is to make the environment aware of its inhabitants and support people to interact with the services of interest in a user-friendly and seamless manner. However, people are increasingly facing challenges in the way they interact, control and manage services in their environment. This is due to the huge volume of the new and existing services, and the complexity in accessing them. One way to address this challenge is to provide context-aware support without requiring them to constantly interact with the environment through explicit means. In such case, the environment can automatically perform relevant actions based on the current situation of people. For example, automatically turning on a light when someone enters a dark room; generating an alarm as soon as the environment detects a fall of an elderly; activating the security system when nobody is at home; etc. This type of automatic actions or pro-activity is commonly known as implicit interaction [95], which helps people to reduce their cognitive load and enable them to focus more on their objective tasks rather than on the technology and settings. However, despite the benefit context-aware interaction offers, there are issues

like imprecision and ambiguity related to the acquisition of context information that greatly influences such interaction. As a result the environment often performs wrong actions or unnecessary actions that lead to users' dissatisfaction and distrust [13, 51]. Although there have been several attempts to deal with imprecision and ambiguity of context information through different feedback and control mechanisms [36, 53, 100], the dynamic level of automated interaction based on such imprecision and its effect on user satisfaction and trust is hardly addressed. Therefore, we need a mechanism to minimize users' distrust and dissatisfaction in order to leverage the true benefit of context-aware interaction. We aim to address this issue in this thesis, which we believe is a significant research problem.

1.2 Existing Interaction Approaches and Limitations

There are many existing and ongoing research efforts that aim to ease the human-environment interaction for providing better quality of experience to the people in sensor-rich aware environments. Some researchers have suggested to use explicit and intuitive interaction and totally avoid implicit interaction. But with no support of implicit interaction or no automation at all, users will be confronted with the whole complexity of the technology and hence will be overwhelmed and loose focus [83]. To overcome this problem and to reduce some burden from the user, other researchers have advocated the use of context-aware interaction or implicit interaction [95]. In the implicit interaction mode, a computer system attempts to automate some tasks that the user would otherwise have performed manually, such as automatically determining user's location and intent (context information) and selecting the right services (implicitly) for him/her. Although, implicit interactions help to reduce the number of explicit user interactions that are usually needed to access the desired services, it can give the users a feeling of loss of control [21, 103]. This motivates some researchers to use mixed-initiative approach [11, 36] for effective human-computer and human-environment interaction. In this mode, the users and the environment dynamically collaborate with each other. Users provide explicit commands when necessary and the environment responds by automatically (implicitly) performing the desired tasks depending upon the context. Unlike explicit interaction, the

implicit and mixed-initiative interaction approaches provide automation support to the user. However, existing approaches in providing this support suffer from the following problems:

1.2.1 Lack of Model to Assess Quality of Context Dynamically

A context-aware system usually collects context information in an automated manner, which primarily involves gathering and processing of heterogeneous sensor data. However, due to the imprecision of sensing and media processing [80, 86], the context processor often produces ambiguous and vague context information. While there have been attempts to model quality of context information such as [50, 51, 72, 117], the need for dynamic assessment of such information in a multi-sensor environment and its application to human-environment interaction has largely been ignored.

1.2.2 Missing Ability to React to Changing Quality of Context

Researchers have utilized quality of context information in different scenarios. Sheikh et al. [98] used quality of context parameters for better context management and to enforce privacy of a user. In another work, Huebscher et al. [61] used similar parameters as part of a middleware for context-aware smart home applications and performing tasks such as selecting appropriate context provider. In [71], the authors presented quality of context based conflict resolution policies to be applied at different layers of a context management system. As can be seen, the work in [61, 71, 98] is not specifically concerned with the interaction issues in a smart environment. In this respect, Dey et al. [36] proposed a mediation technique that involves user to resolve the ambiguity of context when interacting with a context-aware system. But as the changes of context occur at a rapid pace, the mediation process may put extra burden on the user and can affect system's performance, which may reduce the satisfaction of the user. Instead, quality of context may be an indicator for supporting multiple interaction level such as recommending services, automatically invoking services, providing useful information, or performing no actions. However, existing systems mostly provide a single-level of assistance to the user, where the implicit support level or automation is usually pre-determined with the assumption

of accurate context identification. Therefore, they fail to leverage the true benefit of multiple level of assistance according to the changing quality of context information.

1.2.3 Lack of Cost-optimal Selection of Services

In context-aware interaction approach, the environment aims to automatically provide required services to people. Therefore, what type of services should be provided in different contexts remains an important research question. Existing researches have addressed this issue from many directions. For example, [49] suggests to select the best services, which have the highest utility factor. However, in the emerging ambient intelligent era, the environment is equipped with numerous sensing devices and other emerging services. Different users have different interests in accessing or using these services. Furthermore, the services itself may incur varying cost for invocation while the user may have different cost constraints. There are hardly any works that took all these issues into consideration.

1.3 Research Goal

In order to support efficient context-aware interaction in an aware environment perspective, our research goal is to address the above problems with the following agenda:

- **Work on a model to dynamically assess the quality of context:** Automation support in the ambient intelligence environment depends on the current context of the user. Context determination is based on the processing of sensory data that is often imprecise and vague in nature. Therefore, there is a need to devise a model to determine the quality of context information that would have an effect on the level of automation. Furthermore, as context changes over time, the quality of context also needs to be determined dynamically.
- **Incorporate quality of context to adjust the level of implicit interaction:** An intelligent environment should not always act by assuming higher quality of context information. In fact, quality can vary dynamically. Therefore, we will

analyze the relationship between the quality of context information and the levels of automation to be supported by the environment.

- **Define a model to select relevant services:** In context-aware systems, the environment implicitly selects various services based on the context. However, what services to be selected in different context depends on the changing user's preferences. Hence, we will work on a mechanism to learn user's preferences over time. In addition to user preference, the selection of services will also depend on several cost criteria that includes service-related cost (e.g. energy consumption) as well as user-specific cost constraints (e.g. user's budget). Therefore, the goal of our service selection model is to make use of user-service preferences and cost issues.
- **Design and development of a framework for service provisioning in aware environment:** We will work on a service provisioning framework that can integrate the different models such as quality of context model, the multi-level assistance mechanism, and the service selection model into a coherent architecture. However, it is desirable to have loosely-coupled functional units of the architecture so that they can be tuned for different applications scenarios.

1.4 Proposed Approach: Quality of Context Aware Interaction Framework

The cornerstone of the approach proposed in this thesis is to dynamically determine the different level of assistance (implicit actions) an intelligent environment may provide based on the quality of context information and also show how the user's are involved in this process. We develop a framework for service provisioning in order to realize this view. Figure 1.1 shows the different aspects of the work presented in this thesis. These are 1) model and assessment of the quality of context information, 2) context-aware dynamic adjustment of the levels of assistance in the environment, 3) provisioning of relevant services in the environment and 4) design and development of the overall framework. In the following, we briefly highlight our solution to these aspects.

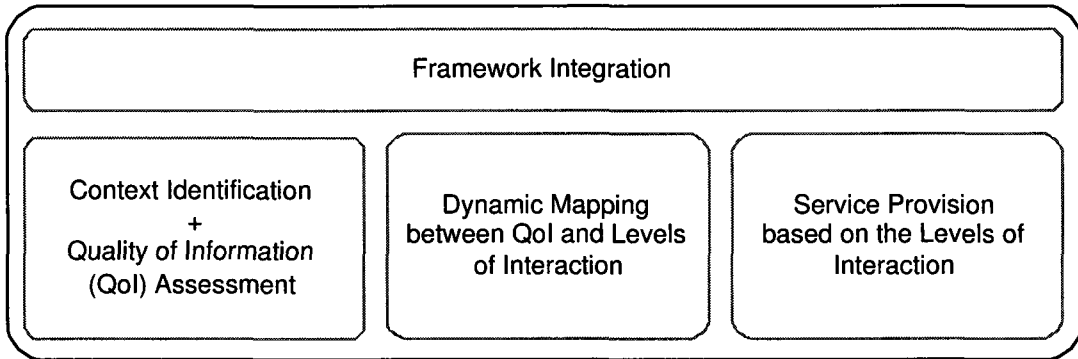


Figure 1.1: Four aspects of the proposed work

Determination of quality of context information: In this thesis, we model Quality of Context Information or simply Quality of Information (QoI) in a multi-sensor environment where the environment employs numerous sensors to assist users with services. Our proposed QoI model defines several key attributes to represent QoI [59]. The consideration of multiple QoI attributes instead of just a single attribute is due to the fact that a single attribute such as accuracy just shows one aspect of the context information quality. However, several other factors such as certainty and timeliness may influence how and when the services are to be provided to the user. For example, an environment should consider the timeliness of an information item to determine whether or not to provide a service at a particular time after an activity occurs. The value of the different QoI attributes we defined will influence the level of implicit interactions the environment will perform. Further details of QoI modeling and assessment are provided in Chapter 3.

Dynamic adjustment of levels of interaction: Context-aware interaction in an intelligent environment could have different levels: providing suggestions, selecting and invoking services, providing information, and not performing any actions whatsoever. Earlier, these sort of levels has been proposed and adopted with success by the automation community [84]. However, in existing context-aware systems, the selection of these assistance levels usually has been done in a static and pre-determined manner, which limits the users to only receive a single level of assistance. For example, if a system only provides suggestions based on the determined context, it is up to the user to do further

interactions with the system to invoke the services. On the other hand, if a system is developed to only invoke services based on the context, the user might be dissatisfied with the wrong selection of services due to the ambiguous determination of context. Therefore, we propose to use quality of context information to dynamically adjust these levels of assistance. Unlike ours, current work utilizes the ambiguity of context information to only disambiguate the identified context by involving users [36]. Our approach not only maximizes the levels of implicit assistance an environment can provide, but also minimizes the wrong actions performed by the environment. We have defined a rule-based approach to dynamically select and adopt an interaction level based on which current human-environment interaction will be directed. In Chapter 4, we present the mapping between the *QoI* and the dynamic levels of implicit interactions.

Service provisioning: After the determination of the level of implicit interaction in the environment, the challenge is to select or suggest the right set of services in a particular context. Users in a context have varying level of interest to a service. Also each service may or may not incur some cost while invoking them. In this thesis we define a mechanism to determine the gain of a service [57, 60], which refers to the extent a service is relevant and useful to a user. We also show how the cost of the service influence the selection process. The cost issue has been introduced in order to enable user to specify their cost limits. Therefore, we make a trade-offs between gain and cost in order to select or suggest the best set of services to the user in a context given the selected level of implicit interaction. The service selection mechanism is described in Chapter 5 in detail.

The design and development of the framework: We have developed a framework to integrate the above aspects in a logical manner and showed its realization with respect to ambient media service provisioning in an aware environment scenario. In this framework, all the modules are designed and developed as loosely coupled so that it can be extended and used for other application. We adopted a service-oriented architecture to glue the different loosely coupled modules. We describe the overall architecture in Chapter 6.

1.5 Thesis Statement and Contribution

The central idea of this thesis is:

Considering the quality of context information to dynamically adjust the different levels of implicit interaction performed by the environment is an effective way to maximize user's satisfaction and avoid distrust in the automated system.

To this end, this thesis makes the following contributions:

- Analyze, model and assess the Quality of Context Information (QoI) in a dynamic multi-sensor environment which would influence the levels of interaction and the selection of services in the environment.
- Define a mechanism to support the mapping between QoI and the different levels of implicit interaction such that this level can be dynamically determined based on the current QoI.
- Design and development of service provisioning algorithm that is based on the user's gain and the service cost.
- Design and development of a framework that incorporates the QoI model, the mapping model and the service provisioning model, which has been realized using a distributed service-oriented architecture.

1.6 Publications

The following peer-reviewed journal and conference papers as well as one book chapter were published during the course of this thesis:

Journals:

1. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Modeling and Assessing Quality of Information in Multi-sensor Multimedia Monitoring

- Systems. *ACM Transactions on Multimedia Computing, Communications and Applications*, 7(1), 2011, (to appear).
2. Pradeep K. Atrey, **M. Anwar Hossain**, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. Multimodal Fusion for Multimedia Analysis: A Review. *Springer Multimedia Systems Journal*, (to appear).
 3. **M. Anwar Hossain**, Jorge Parra, Pradeep K. Atrey, and Abdulmotaleb El Saddik. A framework for human-centered provisioning of ambient media services. *Springer Journal of Multimedia Tools and Applications*, 44(3):407–431, Sept. 2009.
 4. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Learning multisensor confidence using a reward-and-punishment mechanism. *IEEE Transactions on Instrumentation and Measurement*, 58(5):1525–1534, May 2009.
 5. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Gain-based selection of ambient media services in pervasive environments. *Springer Journal of Mobile Networks and Applications*, 13(6):599–613, Dec. 2008.
 6. Jorge Parra, **M. Anwar Hossain**, Aitor Uribarren, Eduardo Jacob, and Abdulmotaleb El Saddik. Flexible smart home architecture using device profile for web services: a peer-to-peer approach. *International Journal of Smart Home*, 3(2), 2009.
 7. Pradeep K. Atrey, **M. Anwar Hossain**, and Abdulmotaleb El Saddik. Association-based dynamic computation of reputation in web services. *Inderscience International Journal on Web and Grid Services*, 4(2):169–188, 2008.
 8. ASM. Mahfujur Rahman, **M. Anwar Hossain**, and Abdulmotaleb El Saddik. Suitability of Searching and Representing Multimedia Learning Resources in a 3D Virtual Gaming Environment. *IEEE Transactions on Instrumentation and Measurement*, 57(9):1830–1839, Sept. 2008.

Conference/workshop papers:

1. Bassam Barake, Pradeep K. Atrey, **M. Anwar Hossain**, and Abdulmotaleb El Saddik. Human-coordinated Camera control for Monitoring Public Spaces. In *International Conference on Information Sciences, Signal Processing and their Applications*, Malaysia, May 2010.
2. Ali Karime, **M. Anwar Hossain**, Abdulmotaleb El Saddik, and Wail Gueaieb. Magic Stick: A Tangible Interface for the Edutainment of Young Children. In *IEEE International conference on Multimedia and Expo*, Cancun, Mexico, June 2009.
3. ASM Mahfujur Rahman, **M. Anwar Hossain**, Jorge Parra, and Abdulmotaleb El Saddik. Motion-path based gesture interaction with smart home services. In *Proceedings of the seventeen ACM international conference on Multimedia*, pages 761-764, Beijing, China, Oct. 2009.
4. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Learning multi-sensor confidence using difference of opinions. In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, pages 809–813, May 2008.
5. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Context-aware QoI computation in multi-sensor systems. In *First IEEE Workshop on Quality of Information (QoI) for Sensor Networks (QoISN 2008)*, pages 736–741, Atlanta, Georgia, USA., Sep.-Oct. 2008.
6. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Management of ambient media preferences in distributed environments for service personalization. In *Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks*, pages 204–209, May 2008.
7. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Dynamic gain estimation in ambient media services. In *1st ACM Multimedia Workshop on*

- Semantic ambient media experiences*, pages 49–56, Vancouver, British Columbia, Canada, Oct. 2008.
8. **M. Anwar Hossain** and Abdulmotaleb El Saddik. Quality-driven human-centered approach for service provisioning in ambient environment. In *3rd ACM International workshop on Human-centered computing*, pages 43–48, Vancouver, British Columbia, Canada, Oct. 2008.
 9. Pradeep K. Atrey, **M. Anwar Hossain**, and Abdulmotaleb El Saddik. Automatic scheduling of CCTV camera views using a human-centric approach. In *IEEE International Conference on Multimedia and Expo*, pages 325–328, Hannover, Germany, June 2008.
 10. Pradeep K. Atrey, **M. Anwar Hossain**, and Abdulmotaleb El Saddik. A method for computing the reputation of multimedia services through selection and composition. In *Fifteenth Annual IS&T/SPIE Conference on Multimedia Computing and Networking*, volume 6818, pages H1–H8, Jan. 2008.
 11. Ali Karime, **M. Anwar Hossain**, Abdulmotaleb El Saddik, and Wail Gueaieb. A multimedia-driven ambient edutainment system for the young children. In *2nd ACM International workshop on Story representation, mechanism and context*, pages 57–64, Vancouver, British Columbia, Canada, Oct. 2008.
 12. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Smart mirror for ambient home environment. In *3rd IET International Conference on Intelligent Environments*, pages 589–596, Ulm, Germany, September 2007.
 13. Pradeep K. Atrey, **M. Anwar Hossain**, and Abdulmotaleb El Saddik. A Multimedia-based System for Monitoring Sleepwalkers. In *IEEE International Conference on Signal Processing and Communications (ICSPC 2007)*, Dubai, UAE, November 2007.
 14. M. Shamim Hossain, **M. Anwar Hossain**, and Abdulmotaleb El Saddik. Multimedia Content Repurposing in Ambient Intelligent Environments. In *IEEE ICDE*

Workshop on Ambient Intelligence, Media, and Sensing (AIMS), pages 11–18, Istanbul, Turkey, April 2007.

15. **M. Anwar Hossain**, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Modeling quality of information in multi-sensor surveillance systems. In *IEEE ICDE Workshop on Ambient Intelligence, Media, and Sensing (AIMS)*, pages 11–18, Istanbul, Turkey, April 2007.
16. **M. Anwar Hossain**, Md. Anisur Rahman, Iluju Kiringa, and Abdulmotaleb El Saddik. Metamaf: Metadata tagging and mapping framework for managing multimedia content. In *IEEE International Symposium on Multimedia*, San Diego, CA, USA, Dec. 2006.
17. **M. Anwar Hossain**, Md. Anisur Rahman, Iluju Kiringa, and Abdulmotaleb El Saddik. Ambient Intelligence Multimedia Management Framework. In *Proceedings of the 5th IEEE SMC Chapter Conference on Advances in Cybernetic Systems (AICS 2006)*, Sheffield, UK, September 2006.

Book Chapter:

1. **M. Anwar Hossain** and Abdulmotaleb El Saddik. LORNAV: Virtual Reality Tool for Navigation of Distributed Learning Objects Repositories. In Borko Furht, editor, *Encyclopedia of Wireless and Mobile Communications*. April 2008.
2. Abdulmotaleb El Saddik and **M. Anwar Hossain**. Nomadic users: Context-aware information retrieval. Samuel Pierre , editor, *E-Learning Networked Environments and Architectures: A Knowledge Processing Perspective*, 2006.

1.7 Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 presents related work with a focus to human-environment interaction, quality of context information and service provisioning.

An approach to model and assess the quality of sensor-driven context information is described in Chapter 3. Here we provide a multi-dimensional metric to evaluate the QoI in a multi-sensor system.

Based on the QoI model, we provide a mechanism to dynamically determine the level of human-environment interaction in Chapter 4. This is followed by Chapter 5 where we describe how to select a set of context-aware services. In this chapter, we also show how to make a trade-off between the gain and the cost of services with respect to user's cost constraints.

The consideration of quality of context information along with the dynamic adjustment of the level of interaction and the particular approach of context-aware service provisioning led to the design of our proposed quality-driven context-aware interaction framework. In Chapter 6 we describe the detail design of this framework that includes high-level diagram, component diagram, and activity diagram. At the end of this chapter we also provide the implementation details of the framework.

We present our evaluation result in Chapter 7. Finally, in Chapter 8, we present a summary of this research and state the future research directions.

Chapter 2

Related Work

This chapter presents previous work that is broadly related to this thesis. As the goal of this thesis is to devise a holistic model of interaction in pervasive environment to provide context-aware services, there are three main areas pertinent to this topic. The primary area of relevance is the human-environment interaction in sensor-rich ambient environment. The second area of relevance is the quality of sensor-driven information that greatly effects the interaction. The third area of relevance is the service provisioning approaches with the fact that human-environment interaction occurs to support people with relevant services. We first comment on existing literature related to these areas. A summary of the existing works with respect to the overall goal of this thesis has been given at the end of this chapter.

2.1 Human-Environment Interaction

In this section, we first present a brief overview of common interaction techniques and comment on some literature that study these techniques in the context of pervasive environment.

Interaction with the computer systems or with the smart environment can be divided into explicit, implicit and mixed-initiative. Explicit interaction is mostly used on GUI interfaces in traditional HCI paradigm. It gives people more control and freedom as to what to perform and when to perform an action. Unlike traditional HCI, interaction

with the ambient intelligence spaces requires much more attention due to the presence of plethora of devices and services. Therefore, adopting only explicit interaction to access a device or to invoke a service in such environment would overwhelm people due to the complexity of interfaces and the increased number of explicit interactions needed [106]. Also, a user would miss the benefit of automation if only explicit interactions are support by the system.

The contrast to explicit interaction is implicit interaction, which involve actions performed by the environment that are not intent-specific toward the computer system, however, the system considers user's actions and context as input to support people [95]. Implicit interactions can thus be considered for pervasive environment to sufficiently reduce the potential number of explicit interactions and to proactively provide relevant information. However, there are few issues that hinder the use of implicit interaction. First, the loss of control and privacy [103] that arise due to fact that computer systems senses user's environment and predicts their needs in a proactive manner. Second, the ambiguity associated with context processing [36, 50, 51, 117, 72] that can impair the implicit behavior of a system. This thesis contributes to improve the implicit interaction approach in presence of such ambiguity of context.

To avoid the pitfalls of implicit-only interaction, researchers have advocated the use of mixed-initiative approaches [52] where the user and the computer systems collaborate in performing an action. In other words, mixed-initiative is the mixture of explicit and implicit interaction. In [52], Horvitz identified some principles of such interaction design, which is widely acknowledged by the community. He also puts concern on uncertainty in determining user's intention and advocates for effective dialog between human and computer to resolve it. The study done by Horvitz is more focused on traditional GUI-based system, however, it can be adapted to other systems. For example, the authors in [36] advocated the process of mediation in the presence of ambiguity of context, where the users are asked to revolve such ambiguity through additional interactions. Unlike these works, instead of asking users to resolve the imprecision of context whenever they occur, we dynamically adjust the level of implicit actions and also allow the user to change the level if not appropriate, which lead to mixed-initiative mode.

The mixed-initiative interaction mode has also been promoted by Villar et al. [106] in

developing a system called Pendle, which supports both explicit and implicit interaction with the environment. The Pendle as a wearable device enables access to multimedia services in an intelligent environment. However, as like others, it does not specify the influence of ambiguous context information.

The different modes of interactions have been adopted by many researchers in the context of smart and ubiquitous mobile environment. The proposal made by Dey and Mankoff [36] has been incorporated in their earlier work on Context Toolkit [35]. It is worth mentioning that authors in [36] emphasize more on mediating ambiguous context information that is used as input rather than on determining the systems's response. However, it is not clear how the quality of context information is captured in their model and how the ambiguity can further be leveraged dynamically for adjusting the level of implicit actions by an aware environment.

In a recent work [83], the authors proposed a mixed initiative approach where the implicit actions performed by the environment can be undone by the user through explicit actions. This work integrates user's media and physical environment in the context of ambient intelligence meeting room. A 3D GUI has been developed which links the physical objects in the virtual environment so that the user can interact with the physical objects through their virtual counterpart. The conflicts arising from the implicit selection of physical objects that do not satisfy a user are handled explicitly by the user via the GUI interface after the automatic selection has been made. However, the authors have not provided any indication whether to include the quality of context information and how to deal with that in terms of interaction with the environment.

A goal-based technique to control the ambient intelligence environment was proposed in [47, 40]. In this scheme, a user does not need to learn all the functionalities of available devices, rather the user will be interacting with the environment by specifying his/her goal. The authors have emphasized on self-organizing device ensembles such that user's goal can be translated into strategy planning to actually perform the desired action without conflict. The authors specified that the goals can be inferred via intention analysis, which would be based on user interaction and context situation. If it is to be done implicitly, there will be issue of uncertainty from the sensor-based context processing. However, the authors have not addressed this issue in their research.

In [88] the authors proposed a framework where services are discovered automatically and their list is provided on the user's mobile device. The actual interaction with the services occurs explicitly by the user when needed. The list of services are selected by finding their highest ratings based on user profile and context similarity measure. Here, the authors have not considered quality of contextual information, rather the processed context has been trusted fully. Authors in [110] also adopted explicit mode of interaction to pull the suggested information in a pervasive display environment. Like many others, they also ignored the issue of handling quality of context while generating system's output.

A location-aware service selection mechanism is presented in [70]. In this approach, the geographical area of a target environment is divided among several service domains, where a set of services can be bounded with a service domain, such as specific library services while within a library. Users in a service domain interact with the system through their handheld device and the system automatically discovers the domain services and displays them as a list, which can be invoked for further access. Like many others, this system also ignores the quality issues to adjust the level of interaction.

To interact with the pervasive environment and provide services to the user, we also adopted an explicit mode of interaction in our ambient media framework (FAMe) [60]. Here, the user first interacts with the environment through a mobile device by asking a particular media type. The system in response automatically provides relevant media that also includes setting up the environment (e.g. change the light condition) based on user's current context and learned preferences. The user is allowed to change the automatic selection that is done by the system, for example change the selected media. However, the presented framework does not consider the ambiguity of context information and therefore unable to deal with such ambiguous information. The FAMe framework has been extended in this thesis to support dynamic level of interaction based on quality of context information.

Based on the above review, we observe that researchers have utilized explicit, implicit and mixed-initiative approaches in various scenarios. However, only few of them actually dealt with the uncertainty or ambiguity of context information [52, 36]. Ambiguity in these works appeared with respect to input and is resolved by providing some kind of

mediation approach such as asking users in case of ambiguity. Nevertheless, it was hardly the focus of these works to dynamically adjust implicit output interactions based on the input uncertainties. Unlike other approaches, we define a QoI model and utilize this model to dynamically determine what would be an appropriate level of implicit actions to be performed by the environment. On a side note from the point of view of conflict handling in ambient environment, existing works mostly adopt the conflict resolution approach once a conflict is determined, while we adopt the conflict avoidance principle.

The following section summarizes the work related to quality of information, which we consider to dynamically adjust the level of interaction in a pervasive environment.

2.2 Quality of Sensor-driven Information

There is a growing interest in modeling and assessing quality of sensor-driven information, which also represents quality of context information. This has an effect on the interaction and context-aware service provision in ambient environment. In this section, we review the literature that put emphasis on sensor-driven information quality measurement. Table 2.1 provides a short summary of the QoI-related works from different perspectives, which are: the QoI attributes used, use of single/multiple media, dynamic/static measurement, and the usage domain of quality metric.

In general, QoI measurement in a sensor-based system may be considered as performance evaluation of different event detection tasks including motion detection, object tracking, location estimation, trajectory observation, person detection, person recognition, activity detection and so on. We observe that the performance of these tasks are measured mostly in terms of accuracy, as presented in Table 2.1. Authors in [94] evaluate the motion detection and tracking performance to identify the operational range of a video-based surveillance system. For motion detection, they measure the spatial deviations from the ground truth (GT) data to the semi-synthetic testing data in terms of several error metrics, such as missing foreground pixels, added foreground pixels and the rate of misclassification. Also for object tracking, they use hit rate, miss rate and false attempt. These metrics are intended to measure the accuracy of motion detection and tracking and hence may contribute partially in measuring the QoI in a multi-sensor

Table 2.1: A summary of QoI-related literature

Reference work	QoI attributes used	Single/multiple media	Dynamic/static	Quality metric used for
[94]	Accuracy related, e.g. rate of misclassification	Single	Static	Motion detection and tracking
[68]	Accuracy related e.g. precision and recall	Single	Static	Motion detection
[73]	Accuracy related e.g. recall and precision	Single	Static	Object detection
[79]	Accuracy related e.g. false positive/negative	Single	Static	Moving object detection
[81]	Accuracy related e.g. false alarms	Single	Static	Object detection
[116]	Accuracy related e.g. false positive/negative	Multiple, but mono-modal	Static	Event detection
[66]	Accuracy, completeness	Single	Dynamic	Data quality measure
[111]	Accuracy, timeliness	Single	Dynamic	Data quality measure
[23]	Timeliness, confidence/ reliability	Single	Dynamic	Event detection
[54]	Certainty, accuracy, timeliness, integrity	Multiple	Static	High-level information e.g. events
<i>Our QoI approach</i>	Certainty, accuracy/confidence, timeliness	Multiple and multi-modal	Dynamic	High-level Information e.g. event in smart home

system. Furthermore, due to the dependencies on GT, these metrics are considered as static quality measure.

Other research, such as [68], use the F-measure method for the optimization of quality evaluation parameters with respect to object-based motion detection tasks. The F-measure optimization may be explained as the weighted harmonic mean of the *Precision* and *Recall* [20] metric, which are often used in evaluating the accuracy of the information retrieval process. Authors in [73] propose several metrics for evaluating the quality of object detection and recognition algorithm. The objects they attempt to detect are vehicles, trees and people in the video data streams. They also use accuracy-related parameters such as precision and recall to compare the output of the object detection

algorithms with the GT. The other quality metric such as timeliness is not considered in their evaluation.

The trend in using accuracy metric for quality modeling and assessment is also visible in other literature. For example, [79] use the false positive track rate, the false negative track rate, the average position error, the average area error, and other accuracy-related metrics for performance evaluation of an object detection mechanism in a real-time video surveillance system. Unlike [79], who measure the performance of a particular object detection mechanism, [81] evaluate several object detection methods to compare the output of the video detector with the manually edited GT. The authors use detection failures, false alarms, and split/merges as the objective metrics to measure the accuracy of those methods.

Notably, the different quality metrics presented in the above literature are mostly applied to evaluate the accuracy of specific tasks (e.g. motion detection). Hence, they are not suitable to uniformly represent the QoI of various types of events as a whole from the application perspective. In order to address this issue, [116] present an evaluation method for video-based event detection solutions that are suitable for detecting high-level events in the context of public subway settings. Example of these events are dropping of an object, walking on rails, and crossing the rails. These events are categorized into three groups such as warning, alarm and critical alarm events. The authors use some generic quality attributes including correct detection, false positive and false negative to quantify the event detection solutions. Unlike the previous approaches, which individually use single camera data for event analysis, [116] utilize multiple camera image sequences to identify the events of interest. Nevertheless, all these approaches measure the quality based on video content analysis. They hardly utilize other important factors such as agreement/disagreement among the sensors and the context of the environment in the overall quality model.

In a recent study, [66] provide a general data quality management framework for the sensor-generated streaming data. In particular, they propose a preliminary DBMS structure for storing the values of different data quality attributes, such as accuracy and completeness. Sensor data quality is also the topic of study in [111]. Here, the authors define quality in terms of accuracy and timeliness, and study the cost versus quality

trade-off for data caching policy to support user's query. The approaches presented in [66] and [111] are specific to sensor data quality and hence may not be suitable for higher level context quality.

In another work [23], QoI has been studied in terms of timeliness and confidence, where the author has analyzed the impact of signal and system parameters such as the sampling rate on these quality attributes. In our earlier work [54], we initially proposed the use of certainty, accuracy, timeliness and integrity quality attributes and modeled them statically. In this thesis, we explore the dynamic modeling approach of the various quality attributes that also appears in [59].

One of the notable aspects of the existing quality evaluation research is that they consider some kind of reference sources or GT against which the processed information are verified when computing accuracy. This process is only good for static evaluation of accuracy in a constrained environment. Quality estimated in such a scenario will not hold in a real application scenario due to the changes in environment context, sensor placement and other issues. In addition, in a dynamic sensor environment the events and activities occur randomly. In that case, it is not possible to dynamically collect and prepare reference data sources to be used as GT. Therefore, we propose to use the confidence measure instead of accuracy and show how it can be measured dynamically on a running system depending on the context. Also in our approach, we model additional quality attributes (e.g. certainty, timeliness) that are important to measure on a dynamic basis in the context of an application. It is worth mentioning that existing research have hardly utilized the multi-sensory evidences obtained on-the-fly in modeling the quality attributes. Although there are many literature that use multi-sensor fusion to identify events in many monitoring scenarios [32], [18], they do not address the dynamic quality of information requirements that we examine here.

Overall, as summarized in Table 2.1, we note that most of the existing works only consider accuracy-related parameters for assessing the quality of specific media processing tasks. Although [66], [111], [23], [54] use additional metrics, [66] and [111] only address data quality issues rather than information quality. Also, the works that attempt to measure information quality do not utilize the dynamically obtained multi-sensor evidences to formulate the quality model. Unlike the above approaches, our proposed QoI

model uses three QoI attributes to dynamically model the quality of detected high-level events/activities using multiple and multimodal sensors [59]. As mentioned earlier, we use this QoI measure in our ambient interaction model.

2.3 Service Provision in Ambient Environment

In this section, we now review literature that study the different service selection and provisioning mechanisms.

The authors in [115], [110] aim to provide context-aware information in the pervasive display devices. Different types of information has been represented as image anchor in [115]. In this work, a user is first identified based on the proximity sensors and RFID tags and later a set of information sources are automatically displayed according to the user's preference parameters. The users are allowed to perform further navigation with the displayed content. It is, however, not clear how the display of information changes based on the changing context. With a similar objective, authors in [110] adopted a machine learning technique in order to analyze the usage log for determining co-occurrence of information items. This co-occurrence is then used to obtain a content utility for selecting appropriate media content. Although content selection based on preference matching and co-occurrence analysis is adopted by these researchers, there are issues such as cost and dynamic changing of preferences that are often ignored in many existing literature in this domain.

A mechanism to select services based on user profile and context similarity measure is proposed in [88]. The objective is to find the highest ratings of the services to be selected and displayed on a mobile device. In a recent work [14], the authors presented a semantic-based ambient media selection framework for intelligent home media entertainment. Like ours, this work uses the different dimensions such as the user's profile, interaction history, and context to personalize media, however, it does not provide the details of how the preference scores are evolved and does not address the mobility issue of the user.

Specific to service selection, we bring the resemblance of our work to the rating estimation problem in typical recommendation systems. For example, in [112] the authors presented the CoMeR platform that supports media recommendations for smart phones.

It consists of a content-based scheme for determining similarity of a media item to the user's preference context, a Naive Bayes classifier for determining the relevance of an item to the situation context, and a rule-based scheme for checking the presentation suitability of a media against device-capability context. The selected list of media is recommended on the user's smart phone. The work in [8] considers multiple dimensions to recommend items to users based on the time of the day, the user and the media item. Unlike the recommendation system referenced here, which usually recommends a particular item to a user based on different criteria, our focus in this thesis is to recommend or invoke media by dynamically determining the gain of a service and make a trade off with the service cost. The media recommendation or invocation is done depending on the level of implicitness that has been determined based on the quality of context information, but can be modified by the user.

The authors in [76] adopts a semantic model for context sensitive message delivery. They model task, domain, location and devices using semantic language. The use of semantic based language gives their system inferencing capability, which is useful to understand the user's task context in a logical manner. The authors have only explored the utility of their method in limited context, for example, in a message delivery scenario.

Research in user modeling attempts to build preference model to capture user's interest and accordingly relates the preference to available content by defining a set of rules. For example, a rule-based approach is adopted by McBurney et al. [75] to define users' preferences for service selection, call redirection, and network/device selection. The approach is applied in the Daidalos project [10] that aims to develop a pervasive service platform and provide flexibility in service selection. However, in this work there is no indication on how the defined rules can address the user's changing need for services in the pervasive environments over a period of time.

Authors in [24] present a platform for capturing the location information of the different static/mobile objects and devices based on the abstract space model of the environment. The platform also computes the location and proximity of the user to the available services. Subsequently, a short-list of accessible services are provided to the user's personal handheld device in the form of vector-based maps. The active museum [30] approach describes a large scale interactive multi-user environment, where the vis-

itors are provided with information about the exhibits. Based on the user's position, resource availability and context information, a varying set of components coordinate the delivery of a specific service to the user. In the process, the user modeling agent analyzes user interaction histories and builds their profile that is used for future customization of information delivery. The service providing agents in the system act as a team and dynamically negotiate the process of service provisioning in the environment with the changing user, their locations and the context. Although the authors have mentioned to include the user's profile and interaction history data, there is no clear methodology that demonstrates how those were integrated into the service selection loop.

A natural language based service matching strategy is presented in [85]. The authors argue that the traditional service selection methods, which often provide a hierarchical list of services, are not sufficient for the user to select the right services from the long list of available services. Such a situation can be avoided if some context information such as location is included in the service discovery mechanism [114]. However, even the context integration with the service discovery may result in many duplicate or similar types of services [105]. To reduce such ambiguity, [85], [69] allows users to specify their requirements to a *ServiceMatcher* agent using a natural language query and the system finds the best matching agents/services for him/her from the large list of agents. This approach uses ontology for describing agent relations semantically. One operational constraint is that the user, if unable to construct correct natural query syntax, will not find the appropriate services even if they are available. Although, natural language based interaction is an intuitive means of service selection, the accuracy of natural languages still remains an issue. Besides, through natural language, services can only be selected via explicit interaction, whereas in a pervasive environment automatic service provisioning based on the current context is often desirable.

In this thesis, we define a multi-criteria service selection approach. More specifically, we dynamically compute a service gain based on user's contextual preference, interaction history and reputation, and makes a trade off between gain and cost, if cost of the services are available [57].

In the following, we provide a summary of the existing literature that we have discussed in above sections.

Table 2.2: Comparison of significant issues between the existing approaches and the proposed interaction mechanism

Significant features	Feature properties	Horvitz [52]	Dey et al. [36]	Shirehjini [83]	Villar et al. [106]	Qin et al. [88]	Xu et al. [110]	Wai Loke et al. [70]	This thesis
Modes of interaction	Explicit					x	x		
	Implicit							x	
	Mixed-initiative	x	x	x	x				x
Consider context ambiguity/ quality of information	Yes	x	x						x
	No			x	x	x	x	x	
Interaction adjustment based on quality of information	Input	x	x						x
	Output								x
	Ignored			x	x	x	x	x	
Conflict management between user actions and automation	Resolution	x	x	x					x
	Avoidance			x					x
	Unknown				x	x	x	x	
Types of services provided	Media/Information	x	x	x	x	x	x	x	x
	Devices/Artifacts			x	x				x
Service/information selection strategy	Context	x	x	x	x	x	x	x	x
	Profile/Preference				x	x	x	x	x
	Interaction history		x				x		x
	Reputation								x
	Utility/Rating					x			
	Gain								x
	Cost								x
Service delivery options	Automatic invocation				x				x
	Recommendation	x	x	x		x	x	x	x
	Filtering							x	
	Information								x
Support for mobility	Yes		x	x	x	x		x	x
	No	x					x		

2.4 Summary

Based on the discussions of existing literature in the domain of human-environment interaction, quality of context information and service provisioning, we now highlight the similarities and differences between our proposed context-aware interaction model and

the existing approaches. Table 2.2 shows the list of several significant features against which we make such comparison. An elaboration of these features are given below.

Modes of interaction: It refers to the three types of interaction that includes explicit, implicit and mixed-initiative from the point of view of taking initiative in a human-environment interaction scenario. We observe that most of the works adopted a mixed-initiative approach due to the obvious reason of giving some control to users.

Consider context ambiguity/ quality of information: With respect to this feature there are only few works that considered context uncertainty in the interaction. In our approach, we propose a quality of information model to represent such uncertainty while interacting with the environment.

Interaction adjustment based on quality of information: In the context-aware interaction scenario, quality of context information may be utilized to resolve the ambiguity of context identification by involving users (input to the system) as well as to adjust what kind of actions to perform in response to inputs (output to the user). We witness that many of the works do not use the context uncertainty to dynamically adjust the level of interaction, except Horvitz [52] and Dey et al. [36]. However, they only adjust input interaction in terms of mediation as a result of uncertainty. In contrast, we consider quality of information to adjust system's response.

Conflict management between user actions and automation: In the ambient intelligence environment users will likely be receiving some sort of automatic interaction support for accessing context-aware services. This may sometimes introduce conflict between explicit interaction of the user and implicit support by the environment. With this respect, Horvitz [52] and Dey et al. [36] adopt a conflict resolution approach, that is they try to resolve conflict once it occurs. In contrast, Shirehjini [83] adopts a conflict resolution and conflict avoidance mechanism like the one presented in this thesis. However, for conflict avoidance Shirehjini [83] manually restricts the functionality of a set of services that may contribute to a conflict, thereby limiting the availability of environment services.

Types of services: Services in the ambient environment are heterogeneous in nature that includes various media/information services and devices/artifacts. Our work considers all these types of services while selecting them for the user.

Service/information selection strategy: In terms of service/information selection, existing research have considered several factors including context, preference, interaction history, reputation, and utility/rating. Unlike many others, we adopted a multi-criteria approach to compute gain in a service and make a trade-off between gain and cost of services whenever cost is available.

Service delivery options: While providing output, the system might automatically invoke a set of services, provide recommendation to the user, apply filter to reduce the list of available services, or present information to guide the user. We observe that existing related approaches mostly leverage the recommendation options and do not take benefit of other alternatives.

Support for mobility: It is important to consider users mobility in the ambient environment. Our approach, like many other systems, considers user's mobility in order to provide context-aware interaction support in distributed environment.

Chapter 3

Modeling Quality of Context Information

As mentioned earlier, the proposed interaction model is based on quality of context information and hence, in this chapter we first provide the details of how we model and assess the quality of context information in a multi-sensor environment. Note, we use the term quality of context information and quality of information interchangeably.

3.1 Requirements for Quality of Context

Context-awareness has gained much attention since the last decade. Its influence on the interaction with the computer systems has been acknowledged by many researchers [95, 97, 19]. In particular, it will likely change the way people interact with the ambient intelligence environment [5]. In order to support context-aware interaction, a system need to define several context-specific attributes that would be useful for the target application. Basically, context is characterized in terms of user's location, time of presence, current activity, and physiological state of the user as highlighted in [96], [7], [27], [34].

An ambient intelligence environment, which contains numerous sensors and devices, adopts several mechanisms to determine the current context or obtain high-level information of interest by processing the elementary data provided by the individual sensors. The information obtained in this manner is then used for interaction support, safety,

and monitoring purposes. For example, in an ambient assisted living space, multiple sensors (e.g. audio, video, RFID, proximity) are deployed to identify the current context and activity of an elderly to ensure his safety and provide him with timely assistance [32]. Despite the importance of context, a system can often obtain imprecise context information as a result of noisy and imprecise sensor measurement and processing [80], [31], [86]. Therefore, a context-aware interaction system should not only use the context but also the quality of context information. The quality of context information (QoI) may be expressed as the measure of “goodness” of the high-level context information, which would allow an automated system or a decision maker to take an informed decision about a situation.

In previous approaches, several attributes including timeliness, accuracy, confidence, throughput and cost have been proposed for estimating the quality of sensory data [25]. Some of these attributes have also been used for assessing the quality of high-level information based on the processing of low level sensory data [23, 116], however in a limited scope and focus. In the domain of traditional information systems, attributes such as accuracy, timeliness, precision, completeness, usability and many other factors have been advocated for information quality measure [39, 108, 77]. Relevant to context-aware research, the authors in [29] have proposed multiple context quality dimensions that gives an indication of the importance of several quality attributes. While these approaches indicate the use of several quality attributes, they have not presented any mechanism on how to compute the different quality attributes based on multiple sensory media.

Based on existing research and motivation, we define a three-attribute QoI metric consisting of certainty, accuracy/confidence and timeliness attributes, which are important to advertise the quality of a particular piece of context information. Here, we have ignored throughput from the QoI attribute due to its being more relevant to sensory data delivery rate than to high-level information. The cost attribute is not of primary importance when computing QoI rather it is related to data collection, processing, and delivery. The precision attribute indicates the level of detail in the sensed data and is inherently related to accuracy [23]. Completeness is not a primary requirement for QoI as any partial information is also of importance to the user. For example, if the processing of sensory media is able to identify the existence of a person, this evidence is of some

interest to the user or the system even if the identification of the person is not known. The usability issue is mostly related to the formatting and readability aspects and hence it is omitted from the QoI metric. In the following, we provide our motivation for the different QoI attributes we considered.

- An ambient environment system that utilizes multiple sensors may consider the individual or group decisions from the sensory media streams in obtaining the information of interest. The different media streams may provide similar or contradictory evidences for the target information. For example, a camera sensor may provide the evidence of a *human presence* while the audio sensor may not. However, given the constraints of the environment (e.g. lighting condition), the detection of human presence may not be possible with full certainty even with the use of camera. On the other hand, if there are redundant sensors in the environment whose observations can be utilized as evidence, the detection certainty may increase significantly [102]. Hence, there is a need to consider this difference of opinions of the participating sensors to model and measure the “certainty” of information.
- Information obtained from the sensor-based observation should accurately reflect the exact state of the events or activities that occur in the monitored environment. Accuracy has traditionally been computed by comparing the output of the system with the Ground Truth (GT) to determine whether the processed information conforms to the reality, where GT refers to the known fact or reality. However, comparing the output with the GT is not feasible when the accuracy is to be computed dynamically. This is due to the unavailability of the GT data or the overhead involved in performing the physical investigation in real-time. Therefore, we propose to use “confidence” [101, 15] as a measure complementary to accuracy, which can be computed dynamically over a period of time based on the observations of the sensors. The use of confidence instead of accuracy may also be justified with the fact that accuracy refers to measure of correctness and that implicitly denotes confidence. For example, if a sensor provides accurate observations, we would have higher confidence in it and vice versa. We will explore this justification later in this chapter.

- In a real-time scenario, timely identification of contextual event may help in safeguarding an individual and preventing an undesired consequence. For example, in a smart home environment, the time to detect an intrusion may be crucial to protect its occupants. In general, timeliness of a sensor-based system is influenced by the computational efficiency of the media processing algorithms and the assimilation of the distributed sources of data. In a particular scenario, the timely determination of information may not be of utmost importance, while in other scenario, the information may become useless to the target application if not delivered in time. Consequently, “timeliness” remains an important QoI attribute.

In this chapter, we show how the above three quality attribute is modeled and assessed in a multi-sensor environment. A preliminary discussion on these quality metrics has been presented earlier in [54], where we showed a static modeling and evaluation of QoI. The dynamic modeling of these metrics, as in this chapter, will appear in [59].

3.2 QoI Problem Specification

In the following, we specify the problem of computing QoI in a multi-sensor ambient environment through notations.

- Let \mathbf{S} be a multi-sensor system deployed in a smart environment for identifying a set of information items $\mathbb{I}_r = \{I_1, I_2, \dots, I_r\}$, r being the total number of information items. Here, by information item we refer to a piece of information that contributes to the identification of context parameters and are obtained from the processing of sensory streams. Examples of some information items are person detection and person identification.
- Let the system \mathbf{S} utilize a set $\mathbb{M}_n = \{M_1, M_2, \dots, M_n\}$ of $n \geq 1$ number of media streams obtained from heterogeneous sensors in order to detect r number of information items.
- Let $q_{b,j} \in [0, 1]$, ($1 \leq b \leq k$, $1 \leq j \leq r$) be the b^{th} quality attribute for the j^{th} information item (I_j) in a particular context, k being the total number of quality

attributes. In our model $k = 3$; the rationale behind considering these quality attributes has been explained before.

- Let w_{I_j} ($1 \leq j \leq r$) and w_{q_b} ($1 \leq b \leq k$) be the weights assigned to r information items and k quality attributes, respectively. These weights will be determined based on the user requirements and context.

Our objective in this approach is to:

1. Provide a dynamic computation model for each of the quality attributes based on the multi-sensor observation. This model will be used to compute the value of b^{th} quality attributes $q_{b,j} \in [0, 1]$ for the individual information items (I_j).
2. Measure the quality of individual information items, QoI_j , $1 \leq j \leq r$. Precisely, this can be expressed as,

$$QoI_j = f(q_{b,j}, w_{q_b}) \quad (3.1)$$

3. Measure the quality of the system in terms of individual quality attribute for all the information items, which can be expressed as,

$$QoI_{q_b} = g(q_{b,j}, w_{I_j}) \quad (3.2)$$

4. Measure the quality of the overall system, QoI .

Note that, f and g are two aggregation functions, which will be described in subsequent sections.

3.3 Detecting Information Items

In this section, we first provide an overview of the processing of information items that lead to context determination in a multi-sensor setting. The modeling of the three QoI metrics, certainty, confidence/accuracy, timeliness, are based on this general architecture, as will be described in the next section.

Figure 3.1 shows a general view of the information item processing framework. In this figure, one of the main tasks of the event detectors is to fuse the different sensory data

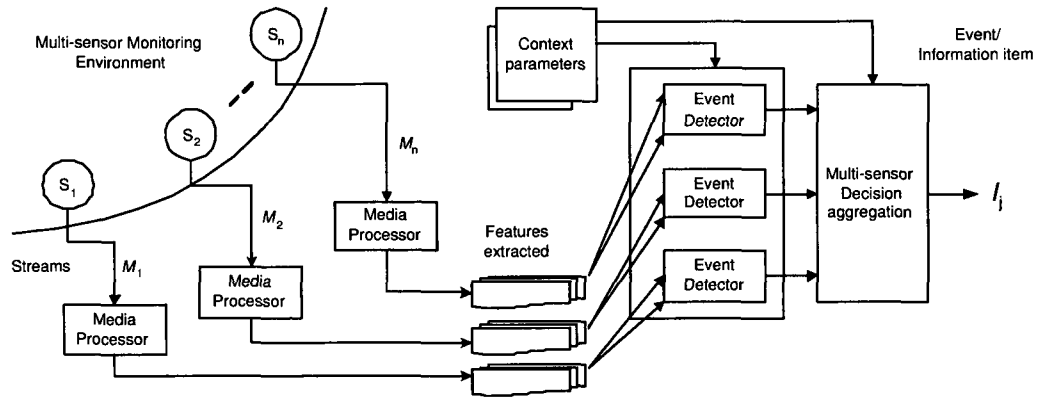


Figure 3.1: Schematic view of a generic event detection framework based on multi-sensor streams.

for making a decision about the occurrence of an event. This is done either by adopting an early fusion (feature-level) or a late fusion (decision-level) approach [44]. In an early fusion approach, the features extracted from the sensor streams are first combined and then sent to an analysis unit (usually a classifier) that provides decision about the task. On the other hand, in a late fusion approach, the different analysis units first provide local decisions (usually in probability score) based on individual media features and then combine these local decisions to make a global decision about the task. In this case, we adopt a late-fusion based event detection mechanism, where the decisions from multiple sensors are combined at the semantic level. We choose this fusion process as it allows us to use the most appropriate methods for analyzing each single modality or media streams [17]. As such, in Figure 3.1, the media processors process the individual media streams to extract various features (e.g. facial features) from the heterogeneous sensors. To determine the occurrence of an event, the corresponding event detector uses the features from the media stream. All the other event detectors follow the same process. Due to the uncertainty in individual decisions, the scores from the individual event detectors are fused to obtain a final score about the occurrence of that event. We assume the fusion process also utilize the past knowledge about the different sensors while performing fusion. Note that several context parameters such as sensor location, lighting conditions etc. also influence the overall computation process. In its simplistic

form, context parameters can be used to define some rules of processing, such as under poor lighting conditions the evidences from the video cameras will have less influence than the audio or motion sensor data.

3.4 Modeling Quality Attributes

We now describe how each of the three quality attributes $q_{b,j}$ ($1 \leq b \leq k$, $k = 3$, $1 \leq j \leq r$) related to an information item are modeled in this thesis. Furthermore, we show the contextual effect such as sensor placement, orientation, lighting condition etc. in computing QoI. Two approaches of QoI aggregation is also presented in this section.

3.4.1 Modeling Certainty

In sensor-based multimedia systems, evidence (decision about events or activities) obtained from the processing of sensory data might be imprecise, ambiguous or incorrect [31, 86, 32]. Therefore, it is important for the user or the system to know how certain is the obtained evidence. The certainty of evidence can be computed in terms of probability of its occurrence. There are various classification methods (e.g. Bayesian classifier, Support Vector Machine) that are usually adopted to obtain the probability of an observation belonging to a particular class.

The various sensory media streams in a multi-sensor system provide different probability scores for different detection tasks. Assuming that the media streams are independently used for an observation but are correlated in providing evidence, the individual probability scores of the streams can be fused together in order to obtain the overall probability of the occurrences of an event. As such, let a system use a set $M'_n = \{M_1, M_2, \dots, M_{n'}\}$ of $n' \geq 2$ media streams from n' different sensors for a particular observation task at a particular time instance. Let $P(I_j|M_1)$ (also denoted as P_1^j) be the probability score based on the media stream M_1 that refers to its local decision for the j^{th} information item (I_j). In fact, for multiple sensors the local decisions obtained from the sensory streams may be similar or contradictory to each other with regard to an information item. To take into considerations this similarity and contradiction among the local decisions, they are grouped into two subsets (ϕ_1 and ϕ_2) by considering 0.50 as the

grouping threshold where the former subset is in support of the occurrence of an event and the latter subset is in support of the non-occurrence of that event. Intuitively, it implies that in a probabilistic framework, sensors with an observation score higher than 0.50 supports the true hypothesis for event occurrence and vice versa. As an example to illustrate this grouping, let us consider 0.60, 0.30, and 0.80 as the observation scores from three sensors. Obviously, 0.60 and 0.80 would mean a support for the target event and belong to group ϕ_1 , while 0.30 would mean the support for non-occurrence of that event. Therefore, the score 0.70 ($=1-0.30$) will belong to group ϕ_2 .

The local decisions of each of the groups are individually fused to determine which of the groups has higher aggregate score. We adopt a Bayesian mechanism for this fusion process as presented in [18]. Note that the fusion process uses the past confidence and the agreement/disagreement among the sensors. In this approach, any two scores in support of evidence I_j are fused using the following equation:

$$P_{i,m}^j = \frac{(P_i^j)^{f'_i} \cdot (P_m^j)^{f'_m} \cdot e^{\gamma_{i,m}^j}}{(P_i^j)^{f'_i} \cdot (P_m^j)^{f'_m} \cdot e^{\gamma_{i,m}^j} + (1 - P_i^j)^{f'_i} \cdot (1 - P_m^j)^{f'_m} \cdot e^{-\gamma_{i,m}^j}} \quad (3.3)$$

Where, $P_{i,m}^j$ is the aggregate probability score based on two sensor media M_i and M_m at time t . Note, the timing parameter t is associated with the left and right hand side of eq. (3.3), however, it is not shown in this equation for clarity of presentation. The denominator in eq. (3.3) is used as a normalization factor to limit the value in $[0,1]$. The terms $f'_i = f_i/(f_i + f_m)$ and $f'_m = f_m/(f_m + f_i)$ are the two weight factors computed from the past confidence of the two participating sensors, where $f'_i + f'_m = 1$. Here, f_i and f_m are the confidences of the sensors or media streams M_i and M_m at time $t - 1$, respectively. We will show in Section 3.4.2 how the confidence attribute is modeled.

In eq. (3.3), $\gamma_{i,m}^j \in [-1, 1]$ refers to the agreement/disagreement (aka. agreement coefficient) [62, 18] between two sensors. This is used as a growth factor while fusing the probability to give certain weight to the observations of the participating sensors. The value of $\gamma_{i,m}^j$ at time t between two sensors can be computed by their current agreement/disagreement with the past value at time $t - 1$ as,

$$\gamma_{i,m}^j(t) = \beta [1 - 2 \times |P_i^j(t) - P_m^j(t)|] + (1 - \beta) [\gamma_{i,m}^j(t - 1)] \quad (3.4)$$

Where, $1 - 2 \times |P_i^j(t) - P_m^j(t)|$ is the current agreement/disagreement between two sensors. Note, this value is 1 when the two sensors are in full-agreement, while it is -1 when the sensors are in full-disagreement. The weighting factors β and $1 - \beta$ are assigned to the current and past agreement coefficient, respectively. While computing the agreement coefficient between a group of sensors and a single sensor, the pair-wise value of the agreement coefficient is averaged using average-link clustering [63]. For example, the agreement coefficient between a group (M_i, M_m) and a single sensor media M_s is computed as $\gamma_{i,m,s}^j = (\gamma_{i,s}^j + \gamma_{m,s}^j)/2$.

Eq. (3.3) and (3.4) are iteratively used to combine the current observation scores based on media streams in ϕ_1 resulting in an aggregate score of $P_{\phi_1}^j(t)$. Similarly, for the non-occurrence of events we obtain the group score of ϕ_2 which is $\widehat{P}_{\phi_2}^j(t)$. Finally, if $P_{\phi_1}^j(t) \geq \widehat{P}_{\phi_2}^j(t)$, the decision based on the ϕ_1 group is considered as winning, which is in support for the occurrence of j^{th} information item. On the other hand, the decision based on the ϕ_2 group is considered as loosing, which is not in support of the occurrence of that event.

Intuitively, we model the *certainty* of an information item $(q_{1,j})$ based on the above observation. The model uses the highest fused score between the two groups (ϕ_1 or ϕ_2) as the certainty of an information item. If, however, the observations of all the participating sensors belong to only one group, the fused score of that group will be used as the certainty value of an information item. Thus, the computation of the certainty attribute takes the following form:

$$q_{1,j}(t) = \begin{cases} P_{\phi_1}^j(t), & \text{if } P_{\phi_1}^j \geq \widehat{P}_{\phi_2}^j \\ \widehat{P}_{\phi_2}^j(t), & \text{otherwise} \end{cases} \quad (3.5)$$

Where, $q_{1,j}(t)$ is the certainty quality attribute at time t for the j^{th} information item. Note, the value of this certainty is referred to as a scalar-valued quality assessment, which can also be represented by its mean, $\mu_{q_{1,j}}(t)$ and variance, $\sigma_{q_{1,j}}^2(t)$ (type-2 quality assessments in [87]). This representation will show the variations of the quality attribute values as a statistical distribution, which is more general than the scalar-only value. In our method, we use a time-varying mean and variance, which are updated as new measurement data are available. By adopting the formalisms stated in [67, 109], we

update the mean and variance of the certainty attribute value as follows.

$$\mu_{q_{1,j}}(t) = \mu_{q_{1,j}}(t-1) + \frac{1}{W} [q_{1,j}(t) - \mu_{q_{1,j}}(t-1)] \quad (3.6)$$

$$\sigma_{q_{1,j}}^2(t) = \frac{1}{W} [D_\mu(t-1) + [q_{1,j}(t) - \mu_{q_{1,j}}(t-1)] [q_{1,j}(t) - \mu_{q_{1,j}}(t)]] \quad (3.7)$$

where, $D_\mu(t) = [D_\mu(t-1) + [q_{1,j}(t) - \mu_{q_{1,j}}(t-1)] [q_{1,j}(t) - \mu_{q_{1,j}}(t)]]$ represents the sum of squares of differences from the current mean of the certainty attribute value. $D_\mu(t-1)$ is 0 at system startup. W is the length of a window within which we maintain the mean and variance value of certainty.

3.4.2 Modeling Accuracy

Accuracy is the most investigated attribute in measuring quality. It refers to the degree of how the observed information conforms to the reality. Therefore, the accuracy of event detection in a monitoring scenario can be computed as the ratio of the number of correctly detected events to the total number of events that occurred in the environment. It is only possible to know the number of correctly detected events through physical investigation or by comparing it with a reference media stream (ground truth or GT) where the events are pre-annotated. The accuracy of a sensor media M_i in detecting information item I_j can be computed using four possible parameters including true positive (TP_i^j), false positive (FP_i^j), false negative (FN_i^j) and true negative (TN_i^j) as follows.

$$Accuracy_i^j(t) = \frac{TP_i^j(t) + TN_i^j(t)}{TP_i^j(t) + FP_i^j(t) + FN_i^j(t) + TN_i^j(t)}, 1 \leq j \leq r \quad (3.8)$$

The accuracy computation using eq. (3.8) is not practically feasible in a real-time system due to the tedious process of performing physical investigation and the unavailability of GT. In such a case, we propose to use the measure of confidence [101, 62, 15] as an alternative to accuracy measurement. Confidence may be interpreted as the level of trust or assurance on the decisions provided by the sensor system. Using confidence not only eliminates our dependencies on GT to evaluate the correctness of information but also enables us to compute it dynamically over the period. Nevertheless, we use eq. (3.8) for initializing the confidence value that is used at the beginning of testing session (see

Section 3.5.3.2) and to verify the dynamically computed confidence with the GT-based accuracy (see Section 3.5.4.6).

In the following, we describe the proposed confidence computation method, which is derived in two steps: first, we obtain the confidence in individual sensor modality that can be used to obtain an information item and second, we compute the confidence in an information item by aggregating the individual confidence values of the sensors that support the given information item. We describe these two steps in the following:

Step1: We model the individual sensor confidence $f_i^j(t)$ at time t based on the fused probability scores $P_{\phi_1}^j(t)$ and $\widehat{P}_{\phi_2}^j(t)$ that we obtained by iteratively using eq. (3.3). As mentioned, if $P_{\phi_1}^j(t) \geq \widehat{P}_{\phi_2}^j(t)$, we consider the occurrence of an event based on the group ϕ_1 , otherwise we consider the non-occurrence of the event based on group ϕ_2 . Therefore, intuitively we adopt a reward and punishment mechanism and accordingly increase the confidence of the streams or sensors in group ϕ_1 if the final decision is based on its outcome. At the same time we decrease the confidence of the streams in group ϕ_2 , and vice versa. This is precisely stated in the following general model,

$$f_i^j(t) = f_i^j(t-1) \pm Rd_i(t)|Pt_i(t) \quad (3.9)$$

where, $f_i^j(t-1)$ is the past confidence of the i^{th} sensor for j^{th} information item. Rd_i is the reward to be assigned to sensor media M_i if it belongs to winning group. In contrast, the punishment Pt_i will be assigned to it if it belongs to the losing group. In the following we derive the model for the reward and punishment factor.

We seek an analogy between the reward and punishment factor to the increase and decrease of accuracy value that can be computed using eq. (3.8). It is obvious from eq. (3.8) that an increase in the numerator will increase the value of accuracy, whereas an increase in the denominator will decrease the value of accuracy. In other words, TP and TN will cause an accuracy increase, while an increase in FP and FN will cause an accuracy decrease. We observe these changes of accuracy value, that is the rate of increase and decrease, and use it in our model. Recall that in our case, we determine an information item based on ϕ_1 or ϕ_2 group depending on whichever group has the highest fused score. Therefore, if ϕ_1 wins, the observation of a sensor in ϕ_1 would have an impact similar to the increase in TP , while at the same time the observation of a

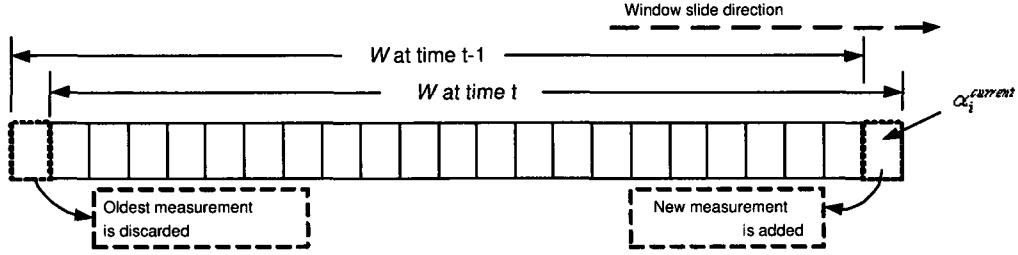


Figure 3.2: The sliding sampling window, where each cell contains either 1 or 0 depending on whether the current observation of sensor i conforms to the winning or losing group, respectively.

sensor in ϕ_2 would have an impact similar to the increase in FN . On the other hand, if ϕ_2 wins, the observation of a sensor in ϕ_2 would have the impact of increased TN , while the observation of a sensor in ϕ_1 would have the impact of increased FP . Based on the above analogy, we obtain the reward/punishment factor for a sensor media M_i over a sample window of length W as,

$$Rd_i|Pt_i(t) = \left| \frac{\alpha_i^{W-1}(t-1)}{W-1} - \frac{\alpha_i^{W-1}(t-1) + \alpha_i^{current}(t)}{W} \right| \quad (3.10)$$

where, each cell of the sample window W (see Figure 3.2), contains either 0 or 1 depending on whether it is a punishment or a reward based on $P_{\phi_1}^j(t)$ and $\hat{P}_{\phi_2}^j(t)$. Although, we propose to maintain a fixed-length window where its size would be determined by the user or the application, it will slide over time to incorporate the decision of the new sampled instance while discarding the decision of the oldest instance. Note, the actual size of this window will depend on how much effect we want to see on the dynamically computed confidence. In fact, focusing on a narrower window would enable the target application to better tune their responsiveness to the changing system performance over time. Also, α_i^{W-1} in eq. (3.10) refers to the number of times sensor i was rewarded within a window of length $W-1$. The term $\alpha_i^{current} \in \{0, 1\}$ contains the latest punishment or reward decision.

Step2: Once we have the individual sensor's confidence, we aggregate these value for the sensors in the winning group to obtain the confidence of the system with respect

to I_j . Therefore, given p number of sensor streams are used in obtaining a decision, the confidence attribute $q_{2,j}(t)$ of the system for the j^{th} information item at time instant t is precisely computed as:

$$q_{2,j}(t) = \frac{1}{p} \sum_{y=1}^p f_y^j(t) \quad (3.11)$$

Similar to the certainty attribute, the confidence quality attribute is also expressed in terms of its mean, $\mu_{q_{2,j}}(t)$ and variance, $\sigma_{q_{2,j}}^2(t)$ to show a statistical distribution over a period of time. This is done using eq. (3.6) and (3.7), respectively by substituting the term $q_{1,j}$ with $q_{2,j}$.

3.4.3 Modeling Timeliness

In a real-time multi-sensor monitoring system, it is crucial to know the timeliness of a detection task in order to provide timely assistance to a person and generate an alarm if necessary. For example, in an assisted living environment if a person falls on the floor, the system should be capable of reporting that event within an acceptable time delay. However, it is very difficult to provide such real-time guarantees in all situations especially in the dynamic sensory environment, as mentioned in many real-time literature [22, 92]. Therefore, we need to measure the timeliness of such a system to know its efficiency.

Looking back at Figure 3.1, we see that the time required to obtain an information item may include the time to capture the media, transmit the media, process the media and fuse the observations for multiple sensors. Further prediction of such time can be made by analyzing the system operations at a more granular level, which a system designer can perform if needed. However, from the point of view of modeling and assessing the timeliness as a QoI attribute, we focus on the application layer timeliness. That is, we try to model the timeliness to determine whether the information of interest (e.g. detection of an unauthorized person) is obtained and propagated to the target application within an allowed range.

To model timeliness, we consider that a system is expected to detect the j^{th} information item (event) within T_j units of time of its occurrence. Here, the value of T_j for an information item is experimentally determined. Note, however, due to changes in the operating environment and other issues involved in a multisensory setting, the

actual units of time for detecting the target event would be $T_j \pm \delta_j$, where δ_j is the jitter associated with the detection task for I_j . In general, the jitter associated with a task varies based on the type of implementation. A task carried out at the hardware device would have lower jitter than a task carried out at the software level [92]. Based on the above specification, we can define the current timeliness of an information item as,

$$q_{3,j}(t) = \begin{cases} 1 & \text{if } \delta_j \leq Th_j, \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

where, $T_j \pm \delta_j$ is the actual time taken to obtain information item I_j and we assume that a system is timely if the value of δ_j is less than or equal to a threshold time value, Th_j . That means, Th_j is the allowed range within which the system's response can vary from the pre-computed value of T_j to be considered as timely. On the other hand, we would consider a system not timely at a particular instance in obtaining the information item I_j if δ_j exceeds its allowed limit. The value of Th_j is usually more in the case of a soft real-time system than in the hard real-time system where $Th_j \rightarrow 0$.

The timeliness quality assessment is then represented by its mean, $\mu_{q_{3,j}}(t)$ and variance, $\sigma_{q_{3,j}}^2(t)$ over a period of time, similar to using eq. (3.6) and (3.7), respectively.

3.4.4 Contextual Effect in QoI Modeling

Environment context information such as geometry, sensor placement, orientation, time, etc. has influence on the computation of QoI [56]. In a multi-sensor system, multiple sensors are placed in strategic locations in order to detect various events or information items of interest. However, at a particular moment we need to determine the sensors whose observation will be considered for the processing of information items, which would eventually influence the computation of QoI. Recall that in Section 3.4 we have stated how QoI can be computed based on multiple sensor observations, where the sensors might have an agreement/disagreement among them even when they are placed in the same observation area. In Figure 3.3 [56], we show various context situations that can play a role in this regard.

In Figure 3.3(a), we show the case when two sensors are placed in a way that they cover the same observation area from two different directions. In such cases, both sensors

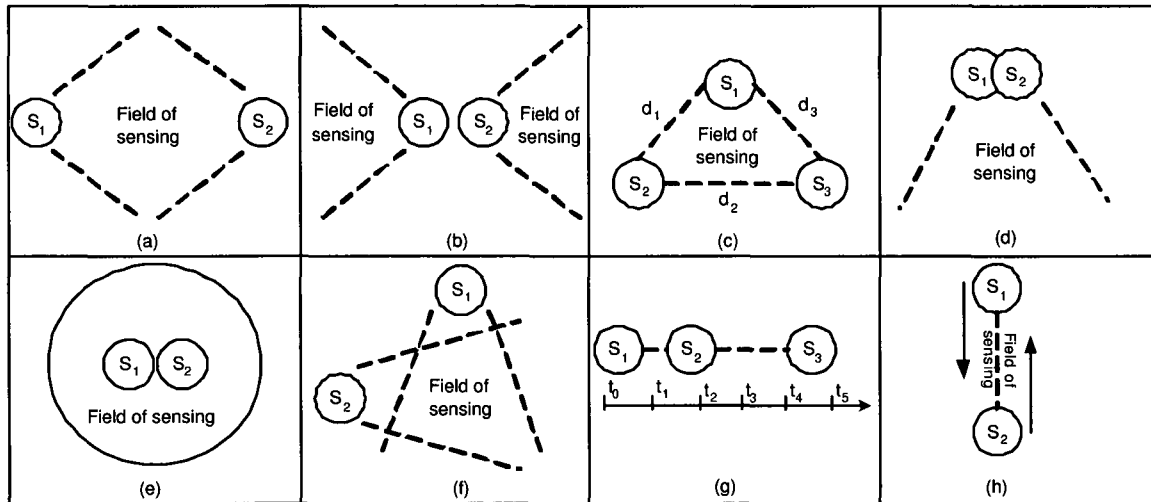


Figure 3.3: Example of some spatial and temporal context that would influence the computation of QoI in the case of the existence of sensors S_1 , S_2 , and S_3 .

can capture an event occurring in their common field of sensing/view. For example, the media streams obtained from two cameras facing each other can be processed at the same time for finding blobs to detect the presence of a human in the monitored environment. In Figure 3.3(b), the two sensors have the opposite field of sensing. Therefore, the observation from S_1 may not be combined with the observation of S_2 at the same instant. However, assume an object has moved from S_1 's field of sensing to S_2 's field of sensing, in such a case the observation of S_1 and S_2 can be combined for collaborative decision-making. Figure 3.3(c) shows the case when three sensors have the same field of sensing and hence the detection of an event may be determined by combining the evidences provided by them. The distance among these sensors may also be considered to determine whether they are neighbors and whether their observations should be aggregated for determining the occurrence of an event. However, this will also depend on the type of sensors and the information item of interest. For example, a camera and two motion sensors are placed in close proximity. Therefore, for some information items (e.g. movement) the observation of these sensors can be considered. However, for other information items (e.g. face detection) only the camera data would make more sense to process. In Figure 3.3(d), two sensors are placed in close proximity and the field of

Table 3.1: Example of some context-dependent rules that influence the computation of QoI

Context-rules	Description
$Cov_j(S_i, S_m) = 1$	Common coverage area between sensors S_i and S_m . Observations of S_i and S_m may be used together for j^{th} event detection and QoI computation
$Cov_j(S_i, S_m) = 0$	No common coverage area between sensors S_i and S_m . Observations of S_i and S_m may not be used together at this instance
$0 < Cov_j(S_i, S_m) < 1$	Some overlapping coverage area between sensors S_i and S_m . Observations from these sensors may be considered together only if the event occurs in the overlapping area

sensing is almost the same for both in the same direction. Therefore, the two sensors may equally contribute to detect an event occurring in their field of sensing.

Similarly, there are other situations that can be of valuable context clue. For example, in Figure 3.3(e), the two sensors have a 360° field of sensing. Therefore, the observation from both of them can be considered at a certain time. For instance, when two audio sensors are placed in such a manner. Obviously the detection of some information item (e.g. a person is shouting) can be based on the outcome of these sensors. In Figure 3.3(f), the two sensors have some overlapping field of sensing. Therefore, any events occurring in that overlapping region may be decided based on their observations together. However, the determination of events in the overlapping region is not always obvious from the sensory stream. Figure 3.3(g) refers to the case when the agreement/disagreement among the sensors can be computed over a timeline. For example, at time t_2 the observation of sensors S_1 and S_2 can be combined together, while at time t_5 the observation of the sensor S_3 can be combined with the group of sensors S_1, S_2 . In Figure 3.3(h), the two sensors are placed apart facing each other and have a uni-directional field of sensing. Besides these situations, there are other cases that can be considered to select appropriate sensors when computing QoI.

Other context information such as the changes in the lighting conditions also have an effect on the QoI computation. For example, a video camera becomes useless at night (poor lighting); therefore, emphasis should be placed on another device. Overall, in order to handle all these situations we propose to use some context-dependent rules

when computing QoI. Some of these rules are stated in Table 3.1. Note the function $Cov_j(S_i, S_m)$ determines whether or not two sensors, S_i and S_m , have the similar coverage area with respect to the determination of j^{th} information item. Depending on the current context (e.g. coverage, lighting condition) this function will produce an updated list of sensors whose observations are eligible for considering them in the decision making process.

3.4.5 Aggregating Quality Attributes

In the above section, we have shown how the proposed quality attributes are computed for an information item. In our approach, the quality of an information item (I_j) is represented by the three quality attributes ($q_{b,j}, b = 1 \dots 3$) described earlier. The current value of these attributes show how much is the QoI of the system in identifying a particular information item. Now, given a multi-sensor system is capable of detecting r different information items, we have two obvious choice to aggregate the quality attributes for representing QoI of the overall system: information item-level and quality attribute-level. In the following we examine this two approaches.

1. Information item-level aggregation: In this approach (see Figure 3.4(a)), the quality attributes of an information item I_j are aggregated using a linear weighted sum fusion scheme [107] to obtain the information item-level quality, QoI_j , as per eq. (3.13). This is further aggregated to obtain QoI of the overall system for all the information items using eq. (3.14).

$$QoI_j = \sum_{b=1}^k q_{b,j} \times w_{q_b}, 1 \leq j \leq r \quad (3.13)$$

$$QoI = \sum_{j=1}^r QoI_j \times w_{I_j} \quad (3.14)$$

Where, $k = 3$ and r is the number of information items. $w_{q_b} \in [0, 1]$, $\sum_{b=1}^k w_{q_b} = 1$ is the weight of the b^{th} quality attribute. The weight w_{q_b} is chosen based on the requirements of the user or the application. For instance, a system that is more sensitive to accuracy than that of timeliness will give more priority to accuracy

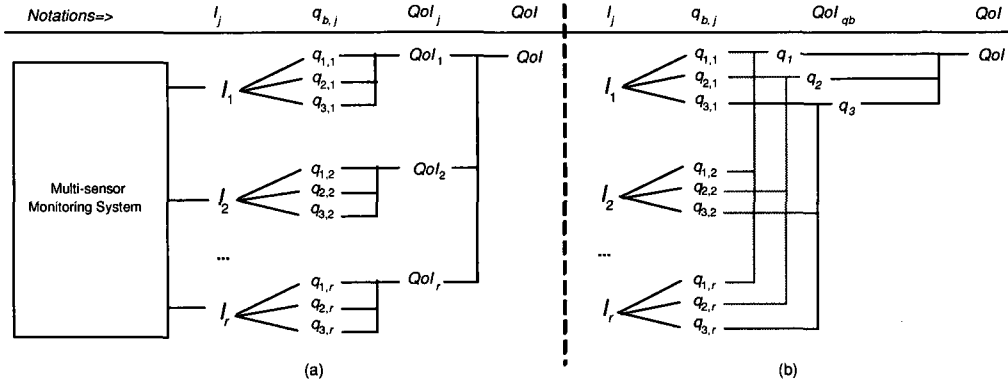


Figure 3.4: Two approaches of aggregation of the quality attributes where (a) quality attributes of an information item are aggregated to obtain information item-level quality (QoI_j), which are then aggregated for all the information items to obtain the system-level QoI and (b) each quality attribute for all the information items are aggregated to obtain the attribute-level quality (QoI_{q_b}), which are then aggregated to obtain the system-level QoI .

while computing QoI_j . Also, $w_{I_j} \in [0, 1]$, $\sum_{j=1}^r w_{I_j} = 1$ represents the weight assigned to the information items according to their importance that the user or application decides.

2. Quality attribute-level aggregation: In this approach (see Figure 3.4(b)), the value of a particular quality attribute for all the information items is aggregated to obtain the attribute level quality, QoI_{q_b} , using eq. (3.15). The attribute level quality values are aggregated further to obtain the overall system level quality using eq. (3.16).

$$QoI_{q_b} = \sum_{j=1}^r q_{b,j} \times w_{I_j}, 1 \leq b \leq k \quad (3.15)$$

$$QoI = \sum_{b=1}^k QoI_{q_b} \times w_{q_b} \quad (3.16)$$

Where, the letters b , j , r , and k all have the same meaning as before. q_b refers to the individual quality attribute value over all the information items.

In summary, the above two approaches will produce different system-level QoI when the relative importance of the quality attributes (w_{q_b}) and the information items (w_{I_j}) vary between them. The fundamental difference we see in them is the different level of abstraction when the aggregation takes place. In one of the approaches, the aggregation first takes place at the information item-level to produce QoI_j and then at the system level, while in the other approach the aggregation first takes place at the quality attribute-level to produce QoI_{q_b} and then at the system level. In real-life scenario, different applications may utilize this different level of abstraction to serve the request.

Note, we used a linear weighted sum fusion scheme in the above two aggregation approaches. This is due to the fact that it allows an application to set priority to a particular quality attribute or to an information item of interest. Such a priority-based approach is also studied in existing literature, for example in [111], where the authors study the trade-off between accuracy and timeliness quality attribute with the query cost by adjusting priority to the different quality attributes. However, beside using linear weighted fusion scheme, other aggregation function may be explored, such as, Count, Maximum, Minimum and the different forms of Average function. The choice among these functions will depend on the need of a particular application.

3.5 Experiments for QoI Assessment

In order to demonstrate the utility of the proposed QoI modeling approach, we have designed and implemented a prototype of a monitoring system and deployed it in our laboratory, which is used as a simulated smart home environment. The QoI model is used to dynamically evaluate the quality of the high-level information items based on the observation of multiple sensors. To demonstrate our methodology we consider two high-level information items, which are person detection (I_1) and person identification (I_2). Our objective in this section is to demonstrate how to assess the QoI in a given multi-sensor system, which is deployed to identify various high-level information of interest. The dynamic quality measure will be finally used in the adaptive interaction model proposed in this thesis.

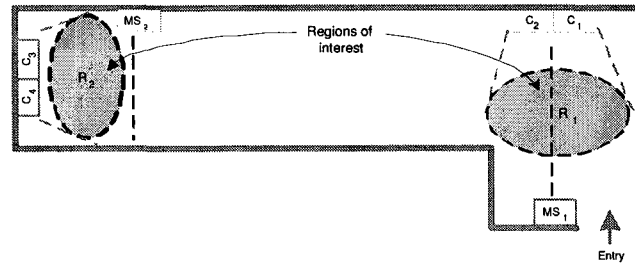


Figure 3.5: Layout of the simulated smart home experimental environment.

3.5.1 System Setup

The developed system utilizes different types of sensors that include four cameras (C_1 , C_2 , C_3 , C_4) and two motion sensors (MS_1 and MS_2). The sensors are placed strategically in the environment to obtain the information of interest, which are I_1 and I_2 as mentioned above. Figure 3.5 shows the physical environment where the sensors are placed for experimental purposes. The placement of these sensors also reflects some spatio-temporal contextual situations as highlighted in Figure 3.3, in particular Figure 3.3(a), 3.3(b), 3.3(c), and 3.3(d). For example, the placement of the two cameras, C_1 and C_2 , in Figure 3.5 relates to the contextual situation in Figure 3.3(d), where the two sensors are in close proximity with a similar field of sensing such that both of their observations may be fused to obtain a corresponding information item. Furthermore, in Figure 3.5, we specify two regions of interest (R_1 and R_2) in the environment where it is more important to detect the occurrence of events. This is done for ease of demonstration, however, a real environment may include multiple regions of interest. The sensors C_1 , C_2 , and MS_1 are deployed to cover region R_1 , whereas the sensors C_3 , C_4 , and MS_2 are deployed to cover region R_2 . Region R_1 is basically the entry of the lab, which is monitored to determine who is entering into the lab. In region R_2 , the two cameras are placed on top of a smart mirror interface [55], which is a service access point for controlling the household appliances and accessing several internet-based services in a personalized manner. Therefore, it is necessary to identify the persons in this area for providing him with personalized services through the smart mirror.

In this setup, we have chosen a variety of sensors in order to utilize heterogeneous

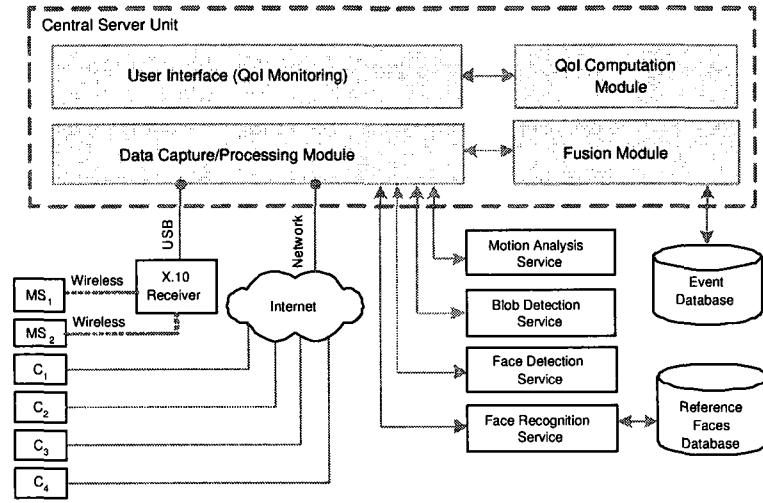


Figure 3.6: Architecture of the prototype for QoI assessment.

sensor observations for different detection tasks. For example, C_1 , C_2 , C_3 and C_4 are IP web cameras, which capture image sequences at 20 frames/sec. with a resolution of 320×240 pixels. The two motion sensors we used detect motion based on Radio Frequency (RF) signals. We also used the cameras for motion sensing, which works based on image change detection.

3.5.2 System Implementation

Figure 3.6 shows the architecture of the developed prototype system. The software development is done in C# programming environment. All the sensors (in our case, the cameras and motion sensors) are connected with a *Central Server Unit*. The four cameras are connected with the server via the network. The camera control classes are installed in the central server unit, which provides a handle within the C# environment to capture the camera image sequences. The two motion sensors that we have used are Hawkeye II Motion Sensor, which send wireless Radio Frequency (RF) signals to the X.10 transceiver connected to the computer via a USB port. An ActiveHome Script SDK is also installed in the server unit to capture the X.10 based motion signals.

In Figure 3.6, it is shown that the Central Server Unit incorporates several modules that accomplish different intended tasks. The *Data Capture/Processing Module* receives

the data/streams from different sensors and processes them using some external media processing services deployed in a separate in-house server. For example, the *Motion Analysis Service* that resides outside of the central server unit is used to analyze and score the motion signals. Similarly, the *Face Detection Service* and *Face Recognition Service* are used to find the presence of faces and identify the detected faces based on captured image sequences, respectively. These external services are deployed using a web service mechanism and are accessed remotely from the central server unit. We will elaborate on the underlying data processing mechanisms that are wrapped in these services in Section 3.5.3.

The decisions obtained by processing multiple sensor data are then fused together at the *Fusion Module* to obtain the overall decision about the occurrence of an event. Note, our fusion module uses a decision-level fusion strategy for aggregating the individual decisions obtained from the sensors. The system uses two databases that are implemented in MySQL 5.0 and hosted in separate machines. They can be remotely accessed via HTTP protocol. The *Event Database* is used to store the event information which the system detects and the *Reference Face Database* stores the facial features of some test participants in order to later verify the faces that are detected during real-time testing.

One of the core modules in the implemented system is the *QoI Computation Module*. This module utilizes the different observation scores that are available at the Fusion Module at each instance to dynamically assess the different QoI attributes. It also computes the overall QoI of the system using our proposed model. The values of the QoI attributes are stored in the *Event Database* along with the information items they characterize. The *User Interface* enables monitoring the results of QoI that are obtained dynamically. Figure 3.7 shows an example snapshot of the running system interface, which intuitively displays the different camera views, the motion status, the event logs and the runtime QoI of the information items and the overall system.

3.5.3 Information Item Processing

Information items are the high-level events of interest that are obtained from the processing of different sensory media. As mentioned earlier, we demonstrate our methodology for two high-level information items that are person detection (I_1) and person identifi-

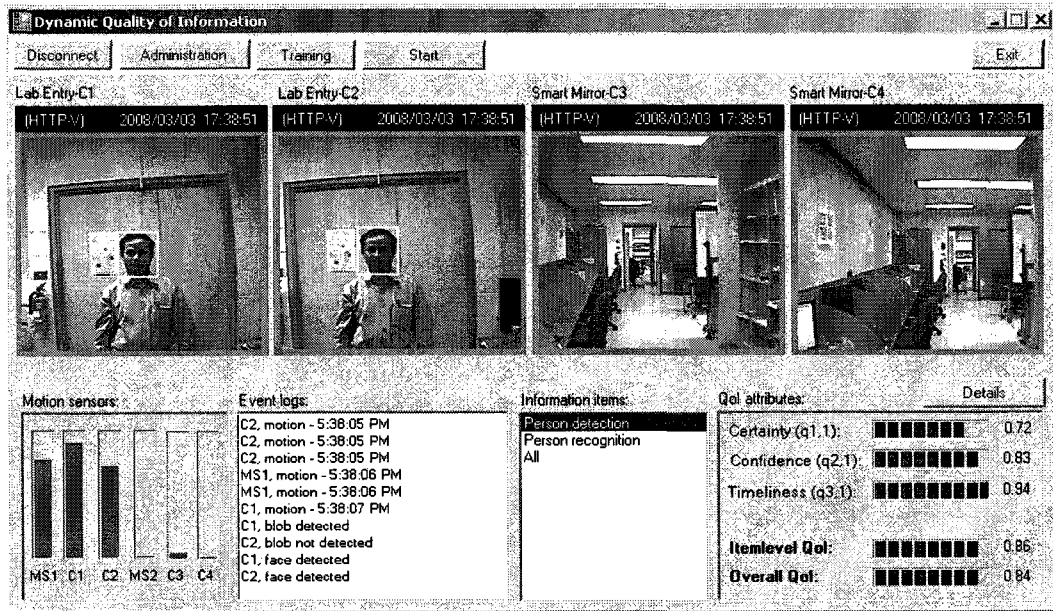


Figure 3.7: UI snapshot of the running system.

cation (I_2). We show that the different sensory media streams are processed in different ways for the identification of these information items as summarized in Table 3.2. For example, we have considered RF-based motion detection, image-based motion detection, blob detection and face detection to determine the existence of a person. Depending on these processing, we obtain the observation scores and fuse them dynamically to determine the information item and assess QoI in the system. In the following, we elaborate these steps of sensory media processing and the fusion of observations that is used to determine an information item.

3.5.3.1 Sensor Media Processing

As shown in Table 3.2, several processing tasks are performed on the sensor media in order to detect and identify a person within the specified regions of interest in the environment. Each of these processing tasks considers a different modality for obtaining the information item of interest. Consequently, the outcome of these tasks are considered as evidence from multiple sensors. We now describe each of these tasks.

The RF based motion sensor we used (e.g. MS_1) for person detection generates a

Table 3.2: Identification of information items based on different sensor media streams

High-level information items	Processing tasks	Streams processed for	Sensors used
I_1 : Person detection	RF	Radio Frequency (RF) signal for motion detection	MS_1, MS_2
	IC	Image Change (IC) detection for motion detection	C_1, C_2, C_3, C_4
	BD	Blob Detection (BD) for person existence from image sequences	C_1, C_2, C_3, C_4
	FD	Face Detection (FD) from the captured image sequences	C_1, C_2, C_3, C_4
I_2 : Person identification	FI	Face Identification (FI) by facial feature extraction and matching with the previously stored facial features	C_1, C_2, C_3, C_4

number of motion signals in a specified period given any movement in the environment that is within the range of RF signals. During training, we observe 100 instances of motion events and record the number of signals it produces for each of the motion events. We then obtain the mean and the variance of the number of signals for a specified time interval, which we later use to obtain the probability score for a given motion event during the real-time testing session.

The other type of motion sensing is performed by using camera images based on significant change detection [89] in image sequences. Change detection is performed by comparing the current image with a reference image frame in terms of the number of different pixels. The camera API allowed us to define three parameters for motion setting - a window to define the area in the image where the motion would be detected, the sensitivity of the detection, and a percentage value to define how much change in the image generates a motion. We set the window size and position in such a way so that it covers the region of interest. The sensitivity value was set to 80%. In our given setting, we experimentally determined the image change percentage. During training, we observe 50 instances of person entrance in the lab and record the percentage of image change for this event. We obtain a mean and variance of the image change from this training and use these values to determine a probability score of a new motion event (e.g. person entrance) during real-time system testing.

Person detection based on blob detection works as follows. We have used an adaptive

Gaussian method [104] to model the background from the captured image. Blob detection is performed in two main steps: 1) by segmenting the foreground from the background using simple matching on the three RGB color channels, where the matching is characterized as a pixel value being within 2.5 standard deviations of the distribution, and 2) by using the morphological operations (erode and dilation) to obtain connected components, which are the blobs. We assume that the area of the blob corresponds to a human when it is greater than a threshold that is determined experimentally. Similar to motion detection, we trained the system with 50 instances and obtained the area of the blob. We determined the mean and variance of the blob area, which we later use during the testing phase to obtain the blob detection probability as a cue for determining a human presence in the monitored regions of interest.

The detection of a person can also be carried out by extracting the facial features such as skin color, eyes, and so on. Often face detection gives better performance than detecting blobs when a person comes within a reasonable distance. We use the Verilook [2] face detection solution, which was available to us, to detect the facial features as evidence of person detection. We obtain a score from the face detection process which we normalize to get a observation score in a probability scale.

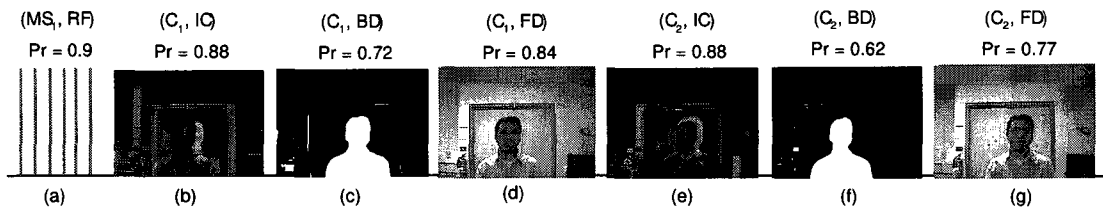
For the other information item (i.e. person identification), we only invoke the face identification service based on the positive results from face detection. In this case also, we use Verilook [2] to extract the facial features from the current image and match it against the stored facial features. The stored facial features are used as the reference data set that correspond to the volunteers who participated during training and testing of the system. The face identification service matches the currently extracted features with that of the reference data set and provides a matching score between $[0, 1]$, which we consider as the observation probability score. Note, in this section we only provided a brief analysis of the different media processing tasks using certain approaches, which can be changed given other better approaches are available.

3.5.3.2 Sensor Observation Fusion

This process occurs while testing the system in real-time (see Section 3.5.4). Before fusion takes place, the individual sensor modality is processed at every sampled instance using

Table 3.3: List of sensors and corresponding processing tasks to determine I_1 and I_2

Regions	Information items	MS_1	MS_2	C_1	C_2	C_3	C_4
R_1	I_1	RF	–	IC, BD, FD	IC, BD, FD	–	–
	I_2	–	–	FI	FI	–	–
R_2	I_1	–	RF	–	–	IC, BD, FD	IC, BD, FD
	I_2	–	–	–	–	FI	FI

Figure 3.8: The observation scores (P_r) from the individual media processing tasks based on different sensory modality at an instance for determining I_1 in R_1 .

the approach described in Section 3.5.3.1, which provides the observation scores from the individual processing tasks. Based on the different observations, we decided which observation scores to fuse together in obtaining an information item at a particular instant. For this we look at Figure 3.5 and Table 3.2 to find out the relevant sensor observations that contribute to determining information items I_1 and I_2 in the two regions of interest. This is summarized in Table 3.3. As an example, we observe from this table that I_1 in R_1 is determined based on seven observation scores as a result of performing RF by MS_1 , IC by C_1 , BD by C_1 , FD by C_1 , IC by C_2 , BD by C_2 , and FD by C_2 . Figure 3.8 shows an instance where the observation scores from the seven processing tasks, (a)-(g), performed on the media streams of MS_1 , C_1 and C_2 , are used in determining I_1 in R_1 . Note, this is a static selection of sensors or tasks based on spatial context information. We will describe in Section 3.5.4.4 how this selection may change dynamically based on other context information. We now describe the fusion process that takes place dynamically for determining an information item.

The observation scores that we obtained from real-time media processing are grouped into two subsets (e.g. ϕ_1 and ϕ_2) as described in Section 3.4.1. In the case of determining I_1 in R_1 , let us consider the scores obtained by the different tasks are $\{0.85, 0.92, 0.34,$

0.59, 0.75, 0.47, 0.82} (see Table 3.4, row 3). We now divide them into $\phi_1 = \{0.85, 0.92, 0.59, 0.75, 0.82\}$ as supporting evidence for I_1 and $\phi_2 = \{(1-0.34), (1-0.47)\}$ as opposing evidence of I_1 . The scores in each of the groups are then fused using eq. (3.3) iteratively to obtain two aggregate scores $P_{\phi_1}^j$ and $\widehat{P}_{\phi_2}^j$ (see Table 3.4, row 6). Note the fusion process considers the agreement/disagreement among the sensors (among different tasks) as well as their previous confidence for the target information item. As shown in Table 3.4- row 4, the agreement/disagreement between any two observation tasks or sensors is computed using eq. (3.4). We consider $\beta = 0.6$ as an example to give more weight to the current agreement/disagreement value. The initial agreement/disagreement between two sensors or the corresponding tasks is considered as 0.0, which changes over time. The previous confidence at the start of testing session is the past accuracy of the sensors for the related tasks, which is computed using eq. (3.8) during the pre-processing step. We obtain these values as $\{0.90, 0.88, 0.70, 0.84, 0.92, 0.76, 0.86\}$ (see Table 3.4, row 2), which will evolve over time according to the confidence computation process and be used subsequently, as will be described in next section. The value of $P_{\phi_1}^j$ and $\widehat{P}_{\phi_2}^j$ that we obtain are 0.95 and 0.78, respectively. Now based on these two aggregate scores, we determine information item I_1 when $P_{\phi_1}^j \geq \widehat{P}_{\phi_2}^j$, otherwise we decide a non-occurrence of an event.

3.5.4 Dynamic QoI Assessment and Result Analysis

In this section we describe how to assess the QoI of the system based on the different processing tasks performed on the sensory media streams. The test is performed in real-time in our experimental smart environment for obtaining information items I_1 and I_2 (in R_1 and R_2 in Figure 3.5). Based on the proposed method, the system dynamically computes the QoI over the period of real-time testing session. Our experiments were conducted for eight hours at different times and days. During this period, we took samples at every 2 seconds (1800 samples per hour) and processed the media captured by different sensors for identifying those information items. The choice of a 2-seconds sampling interval is determined experimentally based on the system settings we adopted. However, samples may be taken at a larger or smaller interval based on the real-time timing constraints and the definition of information granularity. As mentioned in Section 3.4.4, the dynamic QoI computation is also influenced by various contextual information.

Table 3.4: The QoI processing steps shown with an example to dynamically compute the QoI attributes ($q_{b,j}$) at an instance $t = 1$, given the current observation scores and the initial confidence (pre-computed accuracy) of the individual processing tasks for information item I_1 in region R_1 .

Instance	QoI Processing steps	(MS_1, RF) T_1	(C_1, IC) T_2	(C_1, BD) T_3	(C_1, FD) T_4	(C_2, IC) T_5	(C_2, BD) T_6	(C_2, FD) T_7
$t = 0$	Initial confidence (pre-computed accuracy)	0.90	0.88	0.70	0.84	0.92	0.76	0.86
$t = 1$	Current observation (probability score)	0.85	0.92	0.34	0.59	0.75	0.47	0.82
	Agreement/ disagreement (pair-wise)	$(T_1, T_2=0.52), (T_1, T_3=0.37), (T_1, T_4=0.29), (T_1, T_5=0.48), (T_1, T_6=0.22),$ $(T_1, T_7=0.56), (T_2, T_3=0.29), (T_2, T_4=0.20), (T_2, T_5=0.40), (T_2, T_6=0.13),$ $(T_2, T_7=0.48), (T_3, T_4=0.52), (T_3, T_5=0.49), (T_3, T_6=0.44), (T_3, T_7=0.41),$ $(T_4, T_5=0.41), (T_4, T_6=0.53), (T_4, T_7=0.32), (T_5, T_6=0.34), (T_5, T_7=0.52),$ $(T_6, T_7=0.25)$						
	Construct group	$\phi_1 = \{0.85, 0.92, 0.59, 0.75, 0.82\}$ and $\phi_2 = \{(1 - 0.34), (1 - 0.47)\}$						
	Fused observations	$P_{\phi_1}^j = 0.95$ (winning group) and $\hat{P}_{\phi_2}^j = 0.78$ (losing group)						
	Certainty ($q_{1,1}$)	Certainty for I_1 equals the winning group's fused score. That is, $q_{1,1} = 0.95$						
	Confidence in individual tasks (f_i^1)	0.90	0.89	0.69	0.85	0.92	0.75	0.87
	Confidence ($q_{2,1}$)	Confidence for I_1 is the average of the confidence values of sensors in winning group, that is, $q_{2,1} = 0.89$.						
Timeliness ($q_{3,1}$)	Timeliness is calculated based on the maximum acceptable time taken in obtaining the decision about I_1 . In this instance, timeliness is found as $q_{3,1} = 1$.							

Therefore, in the following we first show how the different QoI attributes are computed during testing session (Section 3.5.4.1 – Section 3.5.4.3) and then analyze the context effect on QoI in Section 3.5.4.4. In Section 3.5.4.5 we show how QoI attributes are aggregated to determine the overall QoI of the system. Finally in Section 3.5.4.6 we provide the verification of the QoI assessment results. Note, for clarify of presentation, we consider the determination of information item I_1 in R_1 as an example case to describe the computation of QoI attributes.

3.5.4.1 Certainty Assessment

To assess certainty, the sensory media streams are first processed and the observation scores of relevant media processing tasks (referred as T_1 to T_7 in Table 3.4) are obtained.

The description of each of the media processing tasks are provided earlier in Section 3.5.3.1. The observation scores are then fused to obtain two aggregate scores, $P_{\phi_1}^j$ and $\widehat{P}_{\phi_2}^j$, as described in Section 3.5.3.2. Based on these scores we determine the occurrence or non-occurrence of events. Finally, the fused score of the winning group is considered as the certainty ($q_{1,1}$) in information item I_1 , which is obtained using eq. (3.5). Note, however, if all the observation scores belong to one group (either supporting or non-supporting), the fused score of that group (either $P_{\phi_1}^j$ or $\widehat{P}_{\phi_2}^j$) will be considered as the certainty value. Table 3.4 (row 7) shows the certainty value for the example case of I_1 . Note, we also maintain a mean $\mu_{q_{1,j}}(t)$ and variance $\sigma_{q_{1,j}}^2(t)$ of the certainty attribute, which is computed using eq. (3.6) and (3.7), respectively.

Figure 3.9(b) shows the instantaneous certainty value (in solid line) and its time-varying mean (in dotted line) for all the sampled instances with respect to I_1 in R_1 . Note, in this figure the x-axis represents 81 selected sample instances over the entire test session. These samples were chosen as discrete-event when there were event occurrences in the test sequences for which we wanted to evaluate our system’s performance. Hence, we ignored other sample instances when there were no event occurrences. In Figure 3.9(b), we observe that there are some fluctuations in the certainty value over the period. This is expected as certainty refers to the current detection probability, which is obtained by processing multiple sensor streams and subsequently fusing the observation. In contrast, the mean certainty value has little fluctuations that is usual by definition.

In order to explain the multi-modality issue when computing QoI, we present Figure 3.9(a) that shows how many sensor-specific tasks (modalities) were used at the selected instances. We like to point out that we obtained more certainty if multiple sensors are used in a decision making as opposed to using a single sensor, which in turn results in better QoI. For example, at the 15th instance the system used three modalities for decision making and obtained less certainty as opposed to the 19th instance when all the seven modalities were used. A similar trend can be observed in the presented results. It is worth mentioning that the number of sensors or modalities are not the only factor to have better or worse certainty scores. Some context parameters (e.g. lighting conditions) have an effect on the sensor data, for example, on the camera images and hence on the media processing.

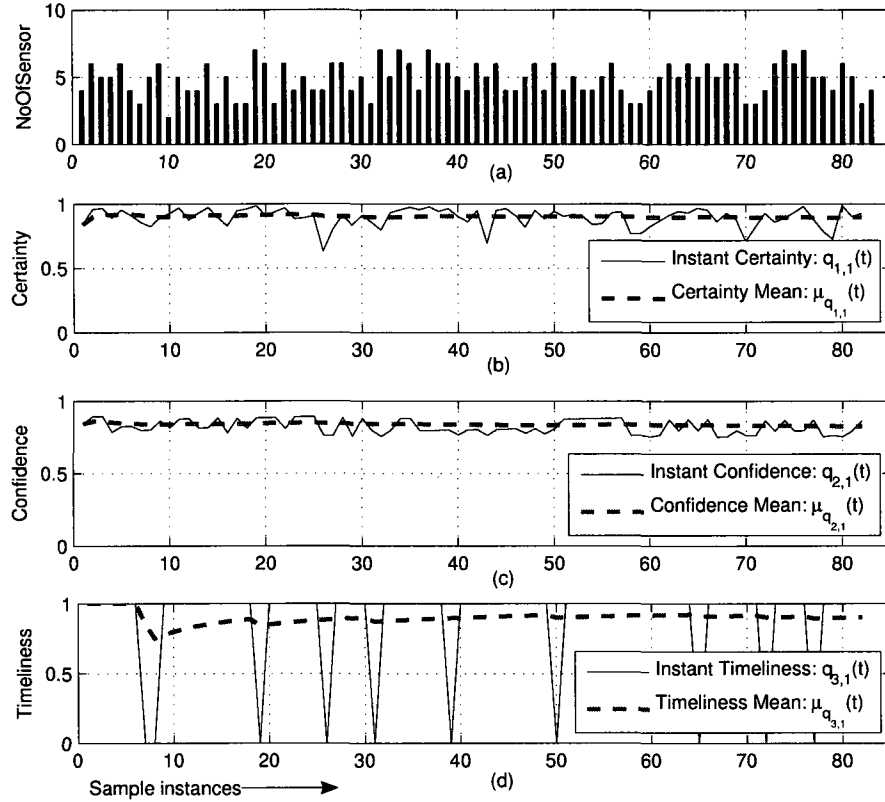


Figure 3.9: Number of sensors and value of the QoI attributes with time-varying mean

3.5.4.2 Confidence Assessment

Confidence is computed using eq. (3.9), where the reward or punishment factor is determined by using eq. (3.10). As mentioned earlier, the confidence of the sensors in the winning group will increase and for the losing group it will decrease. This is shown in Table 3.4 (row 8). Note some of the values, although increased from its initial value, are not visible due to the precision point. Also note that in this approach we compute the confidence in each of the sensor modalities (e.g. confidence in IC based on C_1), which are subsequently used in eq. (3.11) to obtain the instantaneous confidence ($q_{2,1}$) in information item I_1 based on the winning group observations. In Table 3.4, the ninth row shows the dynamically computed confidence based on the current observation and the pre-computed accuracy. We also compute the mean $\mu_{q_{2,j}}(t)$ and variance $\sigma_{q_{2,j}}^2(t)$ of the confidence attribute. Figure 3.9(c) shows the instantaneous confidence value and

its time-varying mean for all the sampled instances with respect to I_1 in R_1 . Note, the confidence varies over the period as the sensors or modalities used to obtain the final decision about an event also varies. Like certainty, we also observe from Figure 3.9(c) and Figure 3.9(a) that multiple sensors in general provide higher confidence, however, it will also depend on the current confidence value of the participating sensors.

3.5.4.3 Timeliness Assessment

For computing timeliness, we first determine the usual time taken in obtaining an information item, which is T_j as described in Section 3.4.3. We also determine what an allowable value of jitter (Th_j in eq. (3.12)) according to the timing constraint set by the application or the user. The value of T_j and Th_j is obtained during the pre-computation phase. Based on these values we assess the timeliness QoI attribute. We observed that most of the time the system could identify I_1 within $1690ms$ during the pre-computation phase, which we consider as T_j . There are some cases when the system took upto $173ms$ more than the T_j value. We consider this value as maximum acceptable jitter (Th_j). Now, using eq. (3.12) we find the timeliness of the system in obtaining I_1 as 1 in the example case at a particular instant due to the timely detection of I_1 (see Table 3.4, row 10). Similar to other quality attributes, we compute the mean $\mu_{q_{3,j}}(t)$ and variance $\sigma_{q_{3,j}}^2(t)$ of the timeliness attribute. In Figure 3.9(d) we show the instantaneous timeliness value and its time-varying mean with respect to I_1 in R_1 . Unlike certainty and confidence, we see sharp fluctuations in the instantaneous timeliness value, which refers that the system did not identify the target information item in time and hence according to eq. (3.12) we obtain a 0 value in those instances.

In the case of timeliness attribute, more sensors does not necessarily provide better QoI. We observed some instances when the timeliness was poor due to more processing involved while considering multiple modalities. Timeliness was poor in some other cases as well when less number of sensors were used. We note this may be related to implementation issues.

Table 3.5: Aggregation of quality attribute values for determining the overall QoI of the system using the proposed two approaches.

Information Items	Quality attributes			Information item level quality (approach 1) $w_{q_1} = 0.4, w_{q_2} = 0.4, w_{q_3} = 0.2$ $w_{I_1} = 0.7, w_{I_2} = 0.3$		Attribute level quality (approach 2) $w_{I_1} = 0.7, w_{I_2} = 0.3$ $w_{q_1} = 0.4, w_{q_2} = 0.4, w_{q_3} = 0.2$			
	$q_{1,j}$	$q_{2,j}$	$q_{3,j}$	QoI_j	QoI	QoI_{q_1}	QoI_{q_2}	QoI_{q_3}	QoI
I_1	0.95	0.89	0.92	0.92	0.89	0.90	0.87	0.90	0.89
I_2	0.78	0.82	0.87	0.81					

3.5.4.4 Context Effect Analysis

In Section 3.4.4 we introduced the context issues that could influence the computation of QoI. In this section, we study the effect of context on the experimental data in terms of sensor or modality selection. In our experiment we used the sensor placement and coverage area as a static context information and it is accordingly summarized as a list of sensors to be used in determining the target information items. This is presented in Table 3.3. Beside sensor placement and coverage, we also utilize the lighting conditions information to refine the selection of sensors. In such case, we ignore the sensors that are sensitive to poor lighting conditions in the environment, for example a camera. Although in the case of a night vision camera this would change, in our experimental environment we do not install any such camera. To study the impact of lighting in our installed settings, we intentionally switched on/off some lights. We used the dawn/dusk sensing capability of MS_1 to know when it is dawn or dusk and use this information in selecting the right set of sensors for a particular task.

3.5.4.5 QoI Attribute Aggregation

Here we show how the different QoI attributes are aggregated using the approaches mentioned in Section 3.4.5. We evaluate the information item-level and quality attribute-level aggregation by assigning priorities for the information items and the QoI attributes.

In Table 3.5 we show the two aggregation approaches of the quality attributes with two instances of data for I_1 and I_2 . In approach 1, QoI_j is the weighted average of the

individual quality attributes as per eq. (3.13). For example, $QoI_1 = 0.92$ is obtained from 0.95, 0.89 and 0.92 by considering $w_{q_1} = 0.4$, $w_{q_2} = 0.4$, $w_{q_3} = 0.2$ weights respectively to each of the $q_{b,j}$ for I_1 . Also, $QoI = 0.89$ is obtained by aggregating QoI_j values 0.92 and 0.81 using eq. (3.14) for both I_1 and I_2 , where we considered their weight as $w_{I_1} = 0.7$ and $w_{I_2} = 0.3$, respectively. On the other hand, in approach 2, $QoI_{q_1} = .90$ is obtained from $q_{1,j}$ of all information items (I_1 and I_2 in this case) using eq. (3.15), that is by aggregating 0.95 and 0.78 using $w_{I_1} = 0.7$ and $w_{I_2} = 0.3$, respectively. Subsequently, the values of QoI_{q_1} , QoI_{q_2} and QoI_{q_3} are then aggregated using eq. (3.16) to obtain system-level QoI.

It is obvious from the QoI values in Table 3.5 that if the priorities for information items and quality attributes remain the same in both the approaches, the same system-level QoI will be reported. However, if the priorities are set differently, the system-level QoI will be different. The fundamental difference between the two aggregation approaches is that, an application can obtain the quality value earlier in approach 1 than that of approach 2. That is, approach 1 can publish the quality value as soon as it obtains an information item, whereas in approach 2 the quality values are gathered after the determinations of all the information items. Hence, approach 1 looks more suitable for real-time systems than approach 2. However, depending on the requirements of an application, it will decide which approach to follow to obtain and publish the QoI value.

3.5.4.6 Result Verification

In this section, we evaluate the suitability of the proposed method of QoI computation. For this, we analyze the results of certainty, confidence and timeliness attributes in turn, and verify them. This is followed by a brief analysis between the proposed approach and other existing approaches.

As mentioned earlier, in our method we adopt a late threshold scheme on the fused scores that are obtained from different complementary media processing tasks to determine the certainty of occurrence of an information item. The certainty of a media processing task depends on the type of algorithm used, the nature of event, the environment context and other factors. If these factors are suitable for a particular task, the certainty score will likely be higher. In a different experimental setup the certainty

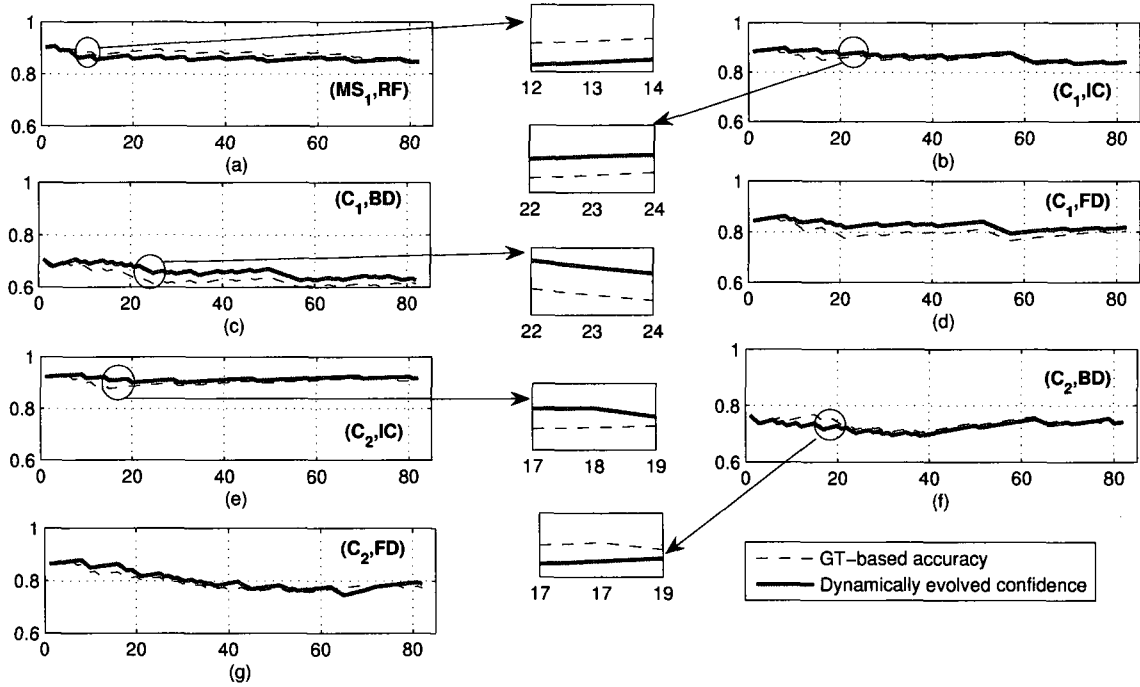


Figure 3.10: The verification of dynamically evolved confidence values of the seven processing tasks based on MS_1 , C_1 and C_2 for determining information item I_1 in R_1 . In the figure, the x-axis represents the number of event instances whereas the y-axis represents the confidence/accuracy value.

score may be different, however, it does not affect the QoI assessment. The goal of the proposed approach is not to use a particular media processing mechanism, rather we investigate how to dynamically compute the QoI of a high-level information item, which the user or an application are interested in.

The confidence attribute of the QoI metric is computed dynamically with an aim to study its use as an alternative to accuracy, which usually requires GT to determine whether an event has really occurred in the monitored environment. In reality, the GT will not be known to a running system. So, our proposed mechanism considers multi-sensor observation scores that are obtained on-the-fly and adopts a reward and punishment mechanism to increase or decrease the confidence of a particular sensor or sensor modality. Therefore, we like to verify how the dynamically computed confidence

matches the GT-based accuracy. The GT-based accuracy means that whenever the GT is known, it is possible to check the system's output with respect to that value and obtain GT-based accuracy. In our experiment, we separately recorded GT (the presence or absence of event in reality) during the testing session and used eq. (3.8) at every sample instance to obtain GT-based accuracy. Figure 3.10 shows the evolution of the confidence values of the sensors with respect to the information item I_1 in R_1 , compared to the accuracy values computed based on GT. Note, in this figure as mentioned earlier the initial value of confidence during testing is set as the accuracy that is obtained during the pre-processing steps. Hence, in Figure 3.10 we see the confidence and accuracy value converge at the starting instance. We further observe from Figure 3.10 that the dynamically evolved confidence based on our proposed method is comparable to the accuracy of the individual tasks that are computed based on GT in a static manner. It is, however, worth mentioning some cases based on the presented results in this figure, for example,

(a) At the 12th event instance, the fused scores $P_{\phi_1}^j \geq \hat{P}_{\phi_2}^j$ where $\hat{P}_{\phi_2}^j = 0$. That means all the sensor processing tasks support I_1 resulting in an increase in all their confidence values. The accuracy is also increased at this instance as there was an event as per GT. In Figure 3.10(a), we highlight this situation with an example in the case of (MS_1, RF) . The confidence evolution of other tasks also follows the same pattern at this instance. In essence, the above case demonstrates that the proposed method works as it reflects the reality that the evolved confidence is proportional to the GT-based accuracy.

(b) The observations based on different sensors or sensor modalities may be divided and contradictory to each other. For example, in Figure 3.10, at the 22nd event instance (MS_1, RF) , (C_1, IC) , (C_1, FD) , (C_2, IC) and (C_2, FD) are in support of I_1 (Φ_1 group), while (C_1, BD) and (C_2, BD) are not in support of I_1 (Φ_2 group). As in this case, $P_{\phi_1}^j \geq \hat{P}_{\phi_2}^j$, the confidence and accuracy of Φ_1 group members will increase, while the confidence and accuracy of Φ_2 group member will decrease. This is due to the fact that sensors in Φ_2 group did not agree on the occurrence of the event whereas in reality there was an event. This increase and decrease of confidence and accuracy is highlighted in Figure 3.10(b) and 3.10(c), respectively. This example also shows that the proposed method dynamically computes the confidence that is comparable to GT-based accuracy.

(c) Another case may arise when there is a contradiction in observation among the different observations. At the 18th event instance the non-supporting observation wins, that is $\widehat{P}_{\phi_2}^j > P_{\phi_1}^j$. This situation generates a false negative case and hence the accuracy of Φ_2 group members will decrease. However their confidence will increase as the final decision is obtained based on this group. On the other hand, although Φ_1 group could not win which results a decrease in confidence dynamically, their observation is inline with the GT and hence the accuracy of observation of this group will increase. We highlight this case in Figure 3.10(f) and 3.10(e), respectively. Note that in our experiment this is just one of the very few instances when the fusion results in negative outcome. This is due to the fact that we used multiple sensor observations in determining an event. We further observe from Figure 3.10 that the dynamically computed confidence is closely comparable to the GT-based accuracy, that is they have machine pattern of ups and downs. Therefore, the proposed method can offer benefit in a dynamic multi-sensor environment to know the QoI attributes without the need for GT with increased trust.

In a similar fashion, our method dynamically computes the confidence values for information items I_2 in R_1 , I_1 in R_2 and I_2 in R_2 . We skip to show these confidence evolution graph for brevity, as they also show a similar pattern of evolution. However, we observed that Mean Square Error (MSE) per event instance between the confidence computed using our method and the accuracy computed based on the GT for all the information items is quite low. This implies that the difference between the dynamically computed confidences is comparable to the GT-based accuracy, which justifies the multi-sensor fusion model we used.

For the verification of timeliness attribute values, we observe the difference between the maximum time taken during the pre-processing step and the real-time testing scenario with respect to processing of media streams and obtaining a decision about an information item. Note that, the time taken also depends on the type of media processing algorithms being used, the processing power of the hardware, the concurrency in processing and so forth. Therefore, we observe some variations from the reference time to the time measured in the running case.

We now analytically provide a comparison between the proposed approach and the other existing approaches. In the proposed approach, we attempted to provide a mech-

anism to dynamically compute the QoI using multi-sensor evidence in terms of several quality attributes, whereas most of the existing approaches primarily focus on accuracy issues such as [94, 68, 81]. In our approach, we stress the need for such quality estimation of the higher level events that a context-aware system requires to provide relevant service for the user. On the other hand, other approaches often provide mechanisms for data quality estimation such as [111, 46]. Finally, we provide a framework for modeling and dynamically assessing QoI in a multi-sensor environment, while some of the existing works concentrate on a particular quality attribute.

3.5.5 Observations based on QoI assessment

Based on the proposed mechanism and the experiment, we make several observations, which are:

- Although we have used several sensors for identifying an information item by processing the media for different supporting tasks, we were not particularly critical about the specific algorithm we used. For example, we used the same face detection software for all the camera images. It would have been good to use different face detection software from different cameras to analyze their impact in QoI. Therefore, the availability of any new algorithm or technique may easily be adopted in our model.
- We use of context information is very helpful in selecting the number of sensors or modalities, which will be used for QoI computation. For example, if we did not consider the lighting condition context, we would end up processing all the media and probably obtain a confident decision, which would have been wrong. Therefore, further investigation about context-aware QoI computation may provide more insight.
- When to compute QoI has been a concern for us, due to the different sampling rates of the media and processing capability of the system. For example, motion processing results are obtained prior to blob detection or face detection results. We handled this situation by defining a time interval to recompute the QoI. In our

case, we experimentally set this time interval as 2 seconds based on the observation of the processing speed. In order to reduce the processing load, we consider a few samples of media streams to be processed in each interval of QoI computation. For example, we only process five image frames per camera at every sampled instance. The media processing score that is obtained within this time interval are processed for computing the QoI attribute values. However, in large-scale deployment, more processors would be needed to better process the media and reflect the QoI of the system. Nevertheless, these are also implementation issues, which can be addressed when real development and deployment are planned.

- We have observed that the motion detection results based on image frames provide an incorrect result if a person walks very slowly toward the camera in a straight path. The situation gets even worse if the person's clothing was similar to the background, which makes the foreground indistinguishable from the background. The same applies for the blob detection tasks. To handle such inaccuracies, we used multiple modalities for determining high-level information. For example, we combined the observation of used RF-based motion processing, image change based motion processing, blob detection and face detection results to determine the existence of a person in a specific area of the environment.
- We have computed QoI based on the regions we defined in the environment. There is a particular benefit in this approach. For example, by observing the dynamically computed QoI along its attribute in a region, we can rethink the deployment of the sensors in the environment.

3.6 Summary

In this chapter, we described a mechanism to model and assess quality of context information in a multi-sensor environment. The proposed method characterizes the QoI in terms of three quality attributes, which are certainty, confidence/accuracy and timeliness. The QoI computation mechanism considers the current sensor observation, the agreement/disagreement among the participating sensors, current context information

and the prior confidence of the sensors in performing various detection tasks. QoI, computed using our method, eventually reflects how good the obtained information is in a real-time scenario where the ground truth is not available to verify the observation. To demonstrate the applicability of the proposed method, we performed experiments by placing several sensors in strategic locations in a lab environment. The outcome of this QoI modeling is used to dynamically adjust the levels of context-aware interaction in ambient environment, which is presented in the subsequent chapters.

Chapter 4

QoI-based Dynamic Levels of Interaction

In the previous chapter, we explained our approach to model quality of context information and dynamically assess the value of different QoI attributes. In this chapter, we particularly focus on the different levels of implicit interaction performed by an environment in order to support the user to reduce their cognitive load while interacting with the emerging sensor-rich multimedia environment. We also provide a model to associate quality of context information with the level of implicit interaction.

4.1 Different Levels of Interaction

The domain of pervasive and ambient intelligence aims to realize calm, context-aware and adaptive systems and environments [5]. Interaction in such environments is challenging especially when our goal is to provide some automatic support to the user based on the context, which can be facilitated by adopting implicit interaction mechanism [95]. However, due to the imprecision of context, the implicit actions performed by an automated system may lead to wrong actions [53, 36]. Therefore, the question remains as to how much to automate and what should be an acceptable level of implicit interaction or automation such that the mis-automation would be minimum.

In order to address the above issues, we highlight the study done by automation

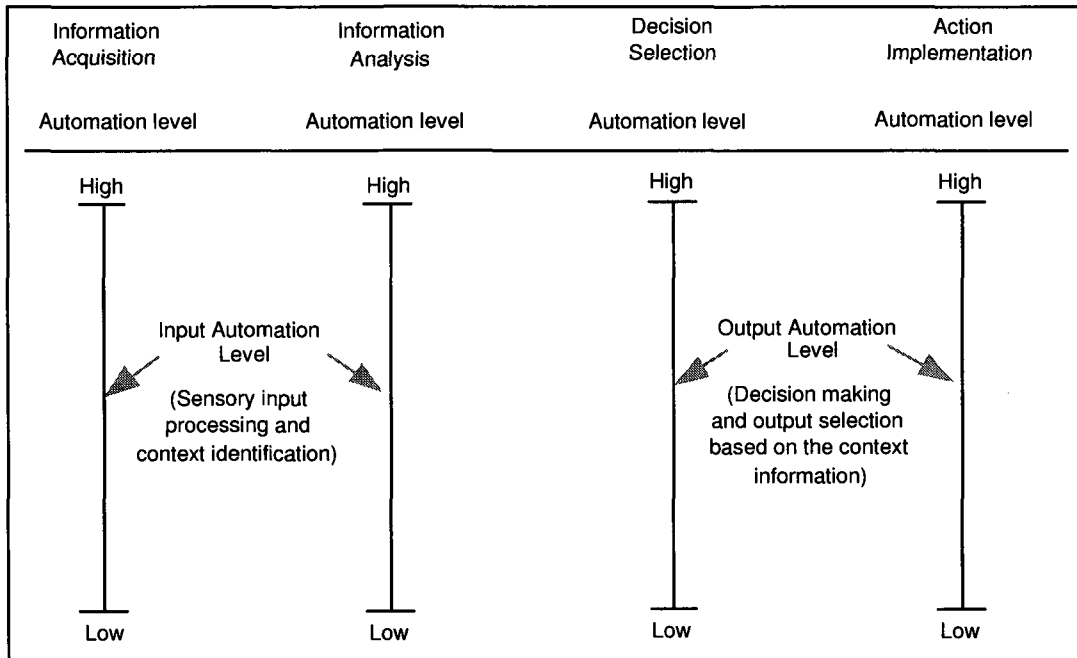


Figure 4.1: Level of automation

researchers [41, 84, 99]. More specifically, in [99] the authors propose four classes of functions usually required in an automation, which are information acquisition, information analysis, decision selection and action implementation. The functions of information acquisition and information analysis are related to input automation while the decision selection and action implementation essentially relates to output automation [103]. Each type of these functions can be carried out manually, semi-automatically or at varying levels of automation in a scale between 0 (low) to 10 (high). Adapted from [84], Figure 4.1 depicts different levels of automation associated with different types of functions.

In Figure 4.1, the highest level of automation essentially means full automation, while the lowest level means manual operation or explicit actions performed by the user. It is worth mentioning that there is a relationship between the input automation and the output automation functions in that the result of the input automation functions are used in many cases to determine the level of output automation and the actual output to be presented. We further describe the level of input and output automation in the following sections.

4.1.1 Input Automation

In this thesis, we use context information collection and processing as the input automation functions. For the actual level of input automation with respect to human environment interactions, we consider the following:

Manual: This level requires explicit user interaction with the environment. It suggests that the user will access any service they wish using some kind of interface. For example, if a user wants to switch on a lamp, he can use a software interface or manual switch to accomplish this task.

Semi-automatic: According to this input automation level a user may explicitly convey his/her intention or goal (user inputs) using some kind of interface whereas the system will take care of the rest. This will include determining the context and fulfill the user initiated goal as per the determined context. For example, a user may ask a system to show him a movie (user's goal) and the system first identify the user's context (user's location, surrounding, activity etc.), fetch his profile and interest, select the best matching movie and start showing it on a TV screen where the user is located. The selection of the actual output (in this case the movie) falls in the output automation mechanism, which will be discussed in the next section.

Automatic: In this input level of automation, the environment implicitly interacts to support users without requiring the users to explicitly provide any commands. For example, a smart home environment might automatically activate the security system as soon as it identifies the residents have gone to sleep. Therefore, in the case of full input automation, the environment automatically attempts to identify user's context and requirements and accordingly automates the output. Existing research in context-awareness paradigm have proposed several mechanisms for automatic context capture and context-adaptive service selection. However, the ambiguity of context remains an issue for them as well [12, 36, 90]. Therefore, we proposed a mechanism to model and assess QoI as presented in Chapter 3. In this chapter, Section 4.2 describes how the values of QoI is utilized in our context-adaptive interaction model.

4.1.2 Output Automation

As noted earlier, the automation of output is related to decision making and output selection. Similar to the level of input automation, it is also possible to have a varying level of output automation in terms of environment interaction. Based on the many different levels defined in [84], we consider the following levels of output automation:

Action: This level of output automation suggests automatic execution of services. For example, when a user sits in front of a TV, the TV is automatically turned on and a favorite channel is selected. User has less control over this kind of fully automated actions, which they do not like in all situations as pointed out by many researchers [103, 37, 21, 82]. Besides, it is not always possible to offer full automation with certainty due to ambiguous input context and hence other levels of output automation are also needed.

Action suggestion: In this level of output automation, the environment only suggests to the user a list of possible relevant services instead of automatically executing them. Therefore, the user need to further interact with the system to select any particular suggestion or change the suggestions and invoke them. Research in recommendation system [112, 26, 8] usually follow this kind of information suggestion approach.

Information: Output automation may only be limited to displaying some information about the users intention or showing general messages about what the user can do. When an environment operates in this mode, it will only provide various information to the user through some interfaces.

No action: This is the basic level of output automation which suggest that the environment should not take any action by itself. This case is common when an environment is not sure what actions to perform or when the user wants to limit the implicit interactions performed by the environment.

It is worth mentioning that the level of input and output automation interleave with each other. Based on the level of input automation, the output automation level of an environment is set to any of the above four levels. In fact, the above output automation levels may change dynamically depending on the outcome of input automation. In this thesis, we provide a model to dynamically adjust the level of output automation (implicit interactions) based on the input quality of context information as explained next.

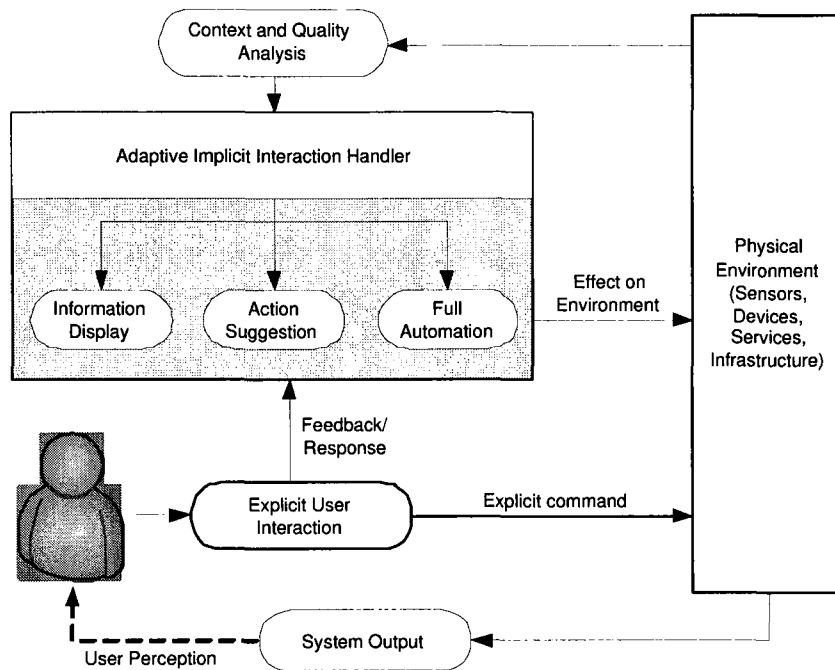


Figure 4.2: QoI and Context adaptive interaction system

4.2 QoI based Levels of Interaction

We have stated that input automation is concerned with the automatic context identification, which in turn influences the output automation. However, the sensor-based observations result in imprecise context determination and hence it is important to consider the quality of context information. To address this, in Chapter 3 we proposed a model to evaluate QoI in terms of three quality attributes, which are certainty, confidence and timeliness. We argue that instead of having a fixed level of output automation, a smart system or environment can leverage the different quality attributes and dynamically switch between the different levels of output automation.

The incorporation of QoI in the context-adaptive interaction model is depicted in Figure 4.2. In this figure, we highlight several points. User's context is processed by the Context and Quality Analysis module, which passes the identified context along with QoI attribute values to the Adaptive Implicit Interaction Handler (AIIH). Based on the current context and QoI, the AIIH attempts to implicitly adjust the interaction

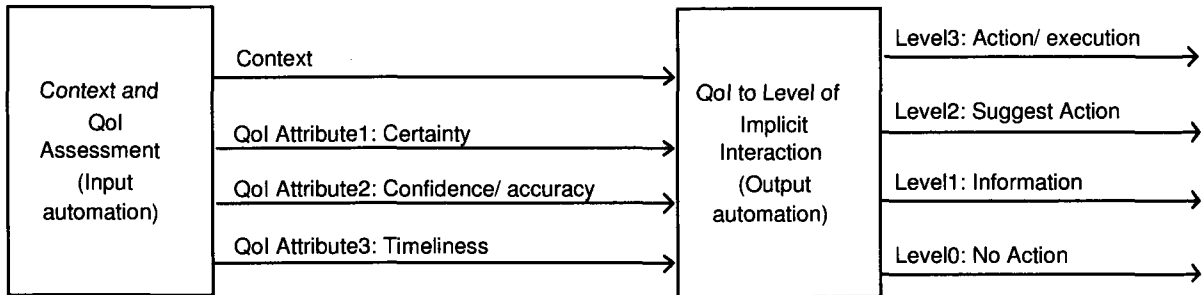


Figure 4.3: QoI-based dynamic level of automation

to any of the levels described in Section 4.1.2. Depending on the implicitly selected levels of interaction, the user perceives system's response in terms of the changes in the environment (e.g. playing a movie on a big screen TV) or obtains notification via some interfaces that also reflects the context and QoI values to the user. At this point, the users provide their feedback in terms of further interaction that basically leads to a mixed-initiative interaction mode. In the proposed approach, the system-driven dynamic interaction level can be overridden by a user using the interface that he/she uses to interact with the system.

We further elaborate on the dynamic adjustment of interaction levels using Figure 4.3, which shows a schematic view of the interrelationship between QoI and the level of output automation. This figure shows that based on the QoI, a pervasive intelligent environment will either execute a service (action), suggest relevant services (suggest action), provide information (information) or take no action at all. Dynamically determining any of these levels helps to overcome the limitations of many context aware applications and recommender systems that provide single level of assistance, for example only providing suggestion. Instead, the proposed approach facilitates a multi-level assistance, which has the goal to maximize the user's satisfaction by providing maximum automation support based on higher QoI, while minimizing user's distrust and dissatisfaction from the mis-automation as a result of lower QoI.

We now briefly analyze the different QoI attributes and explain how they influence the environment interaction. For this, recall the definition of QoI attributes. The value of each of the three QoI attributes ranges from 0 to 1. Accordingly, a value of 1 for

the certainty attribute refers to the fact that the environment has identified the current context with full certainty, while a 0 value would refer to the opposite. The same applies for the confidence and timeliness attribute. However, instead of using this absolute score, we define three levels (high: 0.70 to 1.0, medium: 0.40 to 0.69, and low: 0 to 0.39) for each of these scores to obtain a discrete set of combinations, which will be used to determine the output level. Based on these value ranges of the QoI attributes, we define a rule based approach to define the output level. Table 4.1 shows some of these rules.

Table 4.1: Mapping Rules between QoI and Level of Output

Rule#	Certainty	Confidence/accuracy	Timeliness	Output Level
1	High	High	High	Action
2	High	High	Medium	Suggest action
3	High	High	Low	Suggest action
4	High	Medium	High	Action
5	High	Medium	Medium	Suggest action
6	High	Medium	Low	Suggest action
7	High	Low	High	Suggest action
8	High	Low	Medium	Suggest action
9	High	Low	Low	Information
10	Medium	High	High	Action
11	Medium	High	Medium	Suggest action
12	Medium	High	Low	Information
13	Medium	Medium	High	Suggest action
14	Medium	Medium	Medium	Information
15	Medium	Medium	Low	Information
16	Medium	Low	High	Information
17	Medium	Low	Medium	Information
18	Medium	Low	Low	No action
19	Low	High	High	Suggest action
20	Low	High	Medium	Suggest action
21	Low	High	Low	Information
22	Low	Medium	High	Information
23	Low	Medium	Medium	Information
24	Low	Medium	Low	No action
25	Low	Low	High	No action
26	Low	Low	Medium	No action
27	Low	Low	Low	No action

In Table 4.1, Rule # 1 refers to the situation when the QoI attribute values are all high. Therefore, the environment will be operating on maximum automation level which

is in action mode or service execution mode. According to Rule # 2 and Rule # 3, the environment will be operating on suggest action mode, as the timeliness of the identified context information is at the medium and low level, respectively. This is chosen to avoid over-automation as lower timeliness would mean the user's context information might not be as timely as to invoke an action. Rules # 7-8 means a higher level certainty but lower level of confidence and hence, the environment will be operating in suggest action mode to avoid any mis-automation due to lower quality. However, according to Rule # 9, the confidence and timeliness are both low, which directs the environment to only provide some sort of information, such as to inform the user to explicitly ask for any particular service. Rule # 18 and Rules # 24-27 leads to no action by the environment due to lower quality of the attributes. It is worth mentioning that the actual range of values of the QoI attributes for the output level is adjusted and will be determined through empirical study.

Note, using our approach it is possible to let the user set the level of implicit interaction they want from an environment. In this case, the environment will try to interact according to users adjustment, however it will dynamically change the level as per the variation of QoI values.

4.3 Summary

This chapter discussed different levels of input and output interaction in the context of human-environment interaction. It also presented a mechanism to dynamically adjust the level of interaction based on the quality of context information. Finally, it is important to note that once the level of output automation is determined dynamically, the environment will need to select the right services according to the adjusted level. In the above approach, if the level of output automation is either action or suggestion, a pervasive intelligent environment system need to adopt a mechanism to determine which services to choose, given there are numerous available services. We address this issue in the next Chapter.

Chapter 5

Context-aware Service Provisioning

The primary goal of a quality-driven context-aware interaction model is to provide ambient services to users based on their needs in different context. However, depending on the quality of context information, we proposed to dynamically determine whether to invoke the services, recommend the services or provide clue to support user (see Chapter 4). Nevertheless, in the case of automatic service invocation or providing recommendation, we would require a mechanism to select a set of relevant services. However, which of the services should be provided to the user given the current context is a challenging issue.

In this chapter, we provide mechanisms to dynamically compute the gain of ambient media services, and show how this gain value is used to select a set of services for the user in different contexts. Our service selection mechanism also incorporates the cost of the services and makes a trade-off between the gain and cost while selecting services. The motivation behind the proposed service selection strategy is to provide a cost-effective set of services that would maximize the user's satisfaction level in the environment. In the following, we formally define the service selection problem and explain the proposed methodology for solution.

5.1 Service Provisioning Problem Specification

We describe the problem of the ambient media service selection in a pervasive environment as follows:

- Let there be a system in an environment designed for providing a set $\mathbf{S} = \{S_1, S_2, S_3, \dots, S_n\}$ of n number of services based on different context.
- Within set \mathbf{S} , let there be k different types of services. For example, the service may be of type visual, audio, audio-visual, smell, etc. Some of the services can run simultaneously while others may not depending on their type. For instance, one audio service (such as music) can be provided concurrently with more than one visual services (such as email and weather forecast services); however, if an audio-visual service is running, other audio service(s) should not be executed in parallel with it, unless there is an explicit user intervention. This seems practical as a user might not perceive an audio with another audio or audio-visual service concurrently.
- Let each service S_i , $1 \leq i \leq n$ be characterized by two attributes $\langle g_i^{C_x}, Cost_i \rangle$, with the following reasoning:
 - $g_i^{C_x} \in [0, 1]$: The gain of a particular service S_i to a user in a particular context C_x . The gain of using a service need to be computed dynamically. We provide the details of gain computation in Section 5.2.
 - $Cost_i$: Cost of using a media service S_i in the environment. $Cost_{tot} = \sum_{i=1}^n Cost_i$ is the total cost of using all n services at a time. The cost of using a service may be incurred from various sources, e.g. energy consumed, media subscription, and processing cost. We will elaborate on this in Section 5.3.2.

Our objective from the above definition is the following:

1. To dynamically compute the gain of media services in different contexts.
2. To obtain a subset $\Phi \subseteq \mathbf{S}$ of services based on k types of services such that the overall gain $G_{\Phi}^{C_x}$ of the selected services to a user in context C_x is maximized subject to the total cost constraint $Cost_{\Phi} \leq Cost_{spec}$, where $Cost_{spec}$ is the user specified cost constraint.

Table 5.1: The different factors contributing to gain estimation

Factors influencing gain	Attributes in each factor
User's context	Where (user's location), when (time of presence), what (the activities), with whom (the companion of the user), what mood (of the user).
User's profile	Some user-specific information and their preferences in different media attributes, e.g. genre, actor, actress.
Interaction history	User's interaction history analysis, e.g. statistics, frequent items.
Media reputation	Reputation of individual media service, which is often collected by collaborative filtering approach.

5.2 Dynamic Gain Estimation

The objective of gain computation is to determine the extent a service is useful or satisfying to a user. Essentially, gain in using a service evolves over time for a particular user depending on the context and situation she/he is in. We have identified several key factors that may influence the gain in a service as well as the overall gain from selecting several services together. Table 5.1 summarizes these factors. As shown in this table, the key factors that influence the computation of gain in ambient media services are user's context, profile, interaction history, and service reputation. In the following, we first briefly introduce these factors and later show how the gain in a service is computed.

5.2.1 User's Context

Context has been the epitome of service personalization. It greatly influences the selection and delivery of ambient media. For example, if a user is with his/her children, the system should select media (on request) that are properly censored. In general, context can be characterized using several attributes such as where (user's location, C_{loc}), when (time of presence, C_{time}), with whom (the companion of the user, $C_{companion}$), what (the activities, $C_{activity}$), and even what mood (physiological states, C_{pstate}) as sketched in

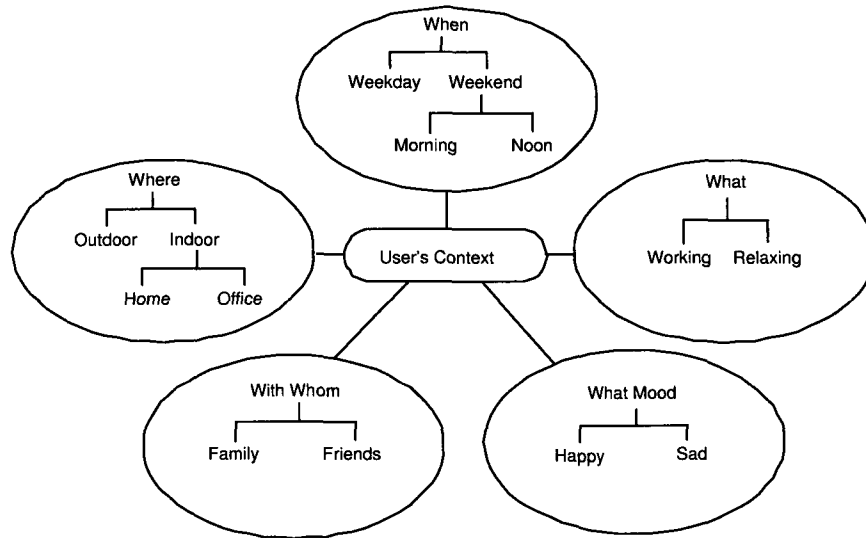


Figure 5.1: The different context parameters

Figure 5.1. Please note that each of the top level attributes is further represented in a hierarchical fashion with corresponding leaf nodes, although Figure 5.1 shows only few leaf nodes as examples. This characterization is derived from existing work in context awareness such as [93], [96], [7], [27], [34]. Several sensory inputs are usually processed to obtain these context attributes.

By considering the above context attributes, there may be numerous context situations based on the value of these attributes, which can be represented as,

$$C_x = C_{loc} \times C_{time} \times C_{companion} \times C_{activity} \times C_{pstate} \quad (5.1)$$

Where, C_x is the context identifier of a particular situation, x is the total number of combinations. For example, in the case of a particular user, we may derive several context identifiers, such as $C_1 = \text{User1}(\text{home, weekend morning, family})$, $C_2 = \text{User1}(\text{home, weekend morning, friends})$, $C_3 = \text{User1}(\text{home, weekend noon, family})$, and so on based on Figure 5.1. Note, the values of the different context attributes may be considered as optional, i.e. at a particular situation we may not have the values for all the context attributes. In that case also, a system should work with partial context information. In our approach, we propose to use a discrete context model to identify each of the context situations. This helps us to avoid handling unlimited context situations having slight

variations among them. However, how many values each of the context attributes have will depend on the level of granularity user needs and the particular context identification framework in use. We consider these context identifiers as the context dimension in the gain computation model. We will show in our method that this gain will vary from context to context for a particular user.

5.2.2 User's Profile

Maintaining the user's profile is one of the popular approaches in personalizing service selection. User's profile can contain static user-specific information (e.g. user name, age), as well as their preferences for different media-related attributes (e.g. movie genre, actor, actress, preferred news types, and singer). However, in a pervasive computing environment these preference data need to be updated dynamically, as user's preferences and needs for media services change over time based on different contexts.

We maintain the user's media-related preference attributes as a set of $\langle \text{media type, attribute, score} \rangle$ tuple, which we call AMP to refer to ambient media preference for a user [58]. The media type in AMP refers to the type of media, for example, movie, music, and news feed. The attribute refers to the metadata of the particular media type. The scores reflect a degree of preference of the user on the different attribute values of a media type at a certain context. Note that user's media preferences represented in [112] uses the $\langle \text{feature, weight} \rangle$ pairs, which is similar to ours except that in our case we maintain a normalized score for the data items in each of the attributes to reflect the relative weight of one data item to another. The scores in the AMP for a user change over time, which reflect their changing need for media services in different contexts. Note that the AMP we define does not identify a particular media service per se, rather it provides an abstraction to the preferences of the users from a meta level based on different context. For example, if $\langle \text{movie, genre, score} \rangle$ refers to a movie attribute that has two data items, $\langle \text{movie, action, 70\%} \rangle$ and $\langle \text{movie, comedy, 30\%} \rangle$, this reflects the fact that the user likes action movies more than comedy movies. Figure 5.2 shows a hierarchical structure of the AMP of a user in a particular context. Note, we have considered lighting and volume in the same level as media type. This is done due to the fact that users usually have a preference of lighting, volume or similar aspects while they experience traditional

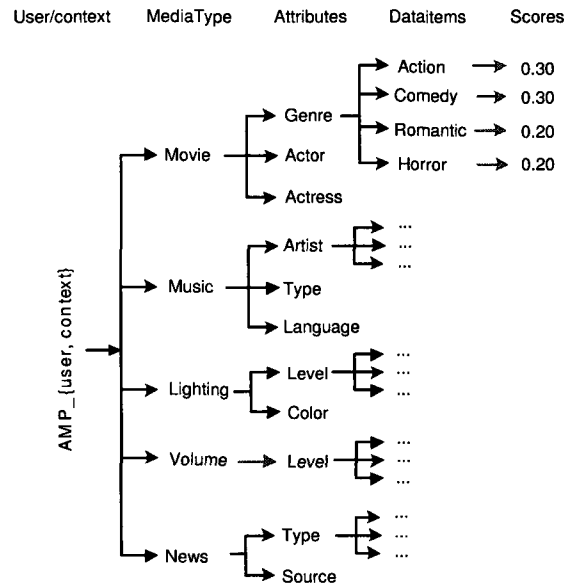


Figure 5.2: The AMP structure of a user in a particular context. Note, the scores of the attribute values will be used in computing the gain of each of the media services

media. For example, a user may prefer to watch movie under low light, while he/she may wish to listen to music with high volume. Also, in this figure we have shown only few attributes related to different media types for clarity, other media-related attributes can be considered in a similar fashion.

The AMP attributes are the core ingredients in our model in determining the service gain dynamically. These preferences can be either explicitly provided by the user or implicitly collected by the system. During the system initialization phase, the user may choose to provide few entries of these preference attributes while the system can later use their interaction history to automatically update the initial scores provided by the user. The motivation behind involving the user at the initial phase is that the proposed method attempts to estimate the gain or the extent a media service would be useful to a user, and hence justifies their involvement. Later, we will describe in Section 5.2.5 how the AMP attributes change over time by using the user's interaction history.

5.2.3 Interaction History

We keep track of the user's interaction history in the environment in order to collect media service usage statistics. We use these statistics to update the scores of the data items for different attributes in the AMP. The interaction history of a service that has not been used at least for a certain time will not be considered in updating AMP. Usually, several approaches can be adopted to incorporate interaction history data and update the AMP. In Section 5.2.5 we describe one such approach for updating the AMP.

In addition to the usage of statistics, it is also possible to leverage some data mining techniques [45] to obtain different patterns of service usage. For example, we can obtain *frequent itemsets* [9] from the historical data. The *frequent itemsets* may provide the recurrent patterns of service usage information where two or more services co-occur together frequently. For example, the system may determine that movie and RSS feed services have frequently been selected by the user in a particular context. Ideally, such patterns can be utilized to construct the attribute set in the AMP or to update the scores of the data items for the given attributes in AMP related to the discovered pattern. However, we leave these possibilities for our future work.

5.2.4 Media Reputation

We also propose to incorporate the reputation [113] of individual media services to compute their gain. Reputation often refers to how good or bad a service is in terms of content, delivery and other factors, although it may be quite subjective in different scenarios. Reputation about an entity is usually maintained by external parties, for example, *MOVIEmeter* an an Internet movie database (www.imdb.com) reflects the popularity ranking about a given movie. The reputation of a media service may be obtained by the user's comments, collaborative feedback, or other means, for example, based on association [16]. The detail of how the reputation of a service can be obtained is out of scope of this chapter.

In our model, we assume a normalized value of the reputation (R_i , where $0 \leq R_i \leq 1$) of a particular media item that is available to us. We cannot only use this reputation score for determining the gain of a service as it is not associated with the pervasive

```

t1 - Home, Weekend Evening, Alone
    Movie (title =Gladiator, Genre =Action, Actor =Russell Crowe, Actress = Connie Nielsen);
    Light (level = Low);
    Renderer (display= TV);
    Volume (level = Medium);

t2 - Home, Weekend Evening, Alone
    Movie (Kill Bill, Action, Uma Thurman, Lucy Liu);
    ...

t3 - Home, Weekend Evening, Alone
    Movie (Rambo, Action, Sylvester Stallone, Julie Benz);
    ...

t4 - Home, Weekend Evening, Alone
    Movie (Bean , Comedy, Rowan Atkinson, Pamela Reed);
    ...

```

Figure 5.3: Example scenario of movie service usage over time in a particular context. Detail interaction data is suppressed for clarity of the figure

context and the preferences of a user. Hence, we need to integrate the user's contextual preference with the reputation score for determining gain in services. We describe this in the following section.

5.2.5 AMP Update Procedure

Let us consider Fig. 5.2, which shows a portion of a sample AMP for a user, and Fig. 5.3, which shows a sample service usage history obtained based on user's interaction. Note that a user may select any type of media at a particular instance, such as a movie and a news feed. However, in Fig. 5.3 we only show the case of movie service to demonstrate the update mechanism of AMP.

Note, Fig. 5.3 shows that a user has watched a movie alone while at home during weekend evening on a TV in low light and medium volume. Therefore, after the t_1 instance the AMP for this user is recomputed to update the preference for each of this selection. This is done in two steps:

- (a) In the first step, the AMP score of an attribute value (e.g. genre = action) that

corresponds to the selected media in the interaction history is updated using eq. (5.2).

$$w'_{a_j, U_i, C_x}(t) = \frac{1 + w_{a_j, U_i, C_x}(t-1) \times noInteractions(t-1)}{1 + noInteractions(t-1)} \quad (5.2)$$

Where, $w'_{a_j, U_i, C_x}(t)$ is the updated score of the selected dataitem for a particular user (U_i) at a particular context (C_x). The index a_j refers to the attribute value that is present in the selected media. The term $w_{a_j, U_i, C_x}(t-1)$ is its past score. Also, the term $noInteractions(t-1)$ is the number of interactions corresponding to the attribute (e.g. genre) being considered, which also represents that from the user interaction the scores of the dataitems of this particular attribute has been updated $noInteractions(t-1)$ times.

(b) In the second step, the scores of all the remaining attribute values for the same media (e.g. genre = comedy) are updated using eq. (5.3).

$$w'_{a_{jj}, U_i, C_x}(t) = \frac{w_{a_{jj}, U_i, C_x}(t-1) \times noInteractions(t-1)}{1 + noInteractions(t-1)} \quad (5.3)$$

Where, a_{jj} refers to the remaining attribute values of a particular attribute. We use the above equations to gradually update the scores of the attribute values of the remaining attributes of the selected media in the AMP (e.g. actor attribute values). For example, by considering the initial scores as of Fig. 5.2 and assuming $noInteractions(t-1) = 10$, the updated dataitem scores of the *genre* attribute after instance t_1 become:

$$w'_{Action, U_i, C_x}(t) = (1 + 0.30 * 10)/11 = 0.36, \text{ using eq. (5.2)}$$

$$w'_{Comedy, U_i, C_x}(t) = (0.30 * 10)/11 = 0.27, \text{ using eq. (5.3)}$$

$$w'_{Romantic, U_i, C_x}(t) = (0.20 * 10)/11 = 0.18, \text{ using eq. (5.3)}$$

$$w'_{Horror, U_i, C_x}(t) = (0.20 * 10)/11 = 0.18, \text{ using eq. (5.3)}$$

Please note, in [57] we used a slightly different approach to update AMP and also defined a *window of interval* to determine the frequency of AMP update. However, here we report the process to update the AMP scores after every instance of service usage with an aim to immediately reflect the preference of the users.

5.2.6 Gain Estimation

Based on the updated AMP scores, we compute the gain of the media services by following a three-step process. These are a) gain based on AMP b) gain based on AMP and reputation, and c) overall gain computation. We describe these steps in the following.

(a) **Gain based on AMP:** This step computes gain based on the AMP scores. The gain is computed for media services that are available in the media repository. We give an illustrative example of gain computation in Fig. 5.4. In this figure we consider a set of sample AMP scores as updated after time instance t_1 (see Fig. 5.3) and based on these scores we estimate the gain of the media services in the media repository. The process of gain estimation can be precisely expressed as,

$$g_{i,amp}^{C_x} = \sum_{t=1}^m Score_{attValue}^t * \beta_t \quad (5.4)$$

Where, $g_{i,amp}^{C_x}$ is the gain of the media service S_i based on AMP scores, which is considered as a temporary gain at this step. $Score_{attValues}^t$ refers to the score in AMP of an attribute value of a media service S_i , as computed using Eq. (5.2) and (5.3). The variable t refers to all the attributes related to a particular media service type (e.g. music) that appear in AMP. $g_{i,amp}^{C_x}$ is computed as the weighted average of the AMP score of the attributes related to a media and the weight is assigned to the attribute itself. The weight β_t in Eq. (5.4) refers to the relative importance of the attribute to a user. For example, a user might be more interested in the *genre* attribute than the other attributes and hence the *genre* attribute should have a higher weight when computing the gain. In the particular example shown in Fig. 5.4, we consider the weights 0.5, 0.3 and 0.2 corresponding to *genre*, *actor* and *actress*, respectively. In a similar fashion, the gain of all the available media services can be computed dynamically.

(b) **Gain based on AMP and media reputation:** In our approach, we also consider service reputation as a factor that influences the computation of gain. Reputation usually refers to how good or trustworthy a service or vendor is in terms of content, delivery and other aspects. A reputation system collects feedback about the participants

	Genre	Score	Actor	Score	Actress	Score
AMP movie	Action	36%	Russell Crowe	15%	Connie Nielsen	10%
	Comedy	27%	Sylvester Stallone	20%	Uma Thurman	20%
	Romantic	18%
	Horror	18%
			Johnny Depp	15%	Meg Ryan	25%
Media repository	Movie1= Proof of Life (Action, Russell Crowe, Meg Ryan)					
	Movie2 = Gladiator(...)					
	Movie3 = The Libertine (...)					
	...					
Gain (Media)	Gain (Movie1) = score_genre (action) * weight_genre + score_actor (Russell Crowe) * weight_actor + score_actress (Meg Ryan) * weight_actress					
	= 0.36 * 0.5 + 0.15 * 0.3 + 0.25 * 0.20 = 0.28					
					
	<u>Considering the weight :</u> Genre = 0.5 Actor = 0.3 Actress = 0.2					

Figure 5.4: Gain estimation of a media service based on AMP, given a set of sample AMP scores

and products and helps to determine trustworthiness among the participants [91]. With respect to the ambient media services, reputation can play a role when determining a particular service gain based on the ratings provided by different users in a collaborative manner. For example, a particular movie may have been rated positively higher and such a rating score may be considered along with the AMP scores, when determining the gain of a service to a user. We should, however, note that in cases when reputation is not available, the proposed gain estimation method will ignore this value and estimate gain only based on the AMP scores.

To compute gain based on AMP and reputation, we aggregate the temporary gain as computed using Eq. (5.4) with the reputation of the media services. This is done using Eq. (5.5) in the following.

$$g_{i,amp,R_i}^{C_x} = \alpha * g_{i,amp}^{C_x} + (1 - \alpha) * R_i \quad (5.5)$$

Where, $g_{i,amp,R_i}^{C_x}$ is the gain of the media service S_i based on AMP and the reputation R_i of that service. α and $(1 - \alpha)$ are the weights assigned to the AMP-based gain and the

reputation, respectively. Therefore, in Fig. 5.4, if we assume the reputation of Movie1 as 0.74 and the value of α as 0.70, the updated gain of Movie1 becomes 0.42 according to Eq. (5.5). The value of α should be set as per the importance of AMP scores and reputation to a user.

It should be noted that the gain estimation method uses AMP and reputation to compute gain in media services, where the reputation is subject to the availability. However, we may also ignore reputation if it is not available. This is clear from Eq. (5.5), where we can set $\alpha = 1$ to ignore the reputation and give full importance to the AMP-based gain estimation. We also like to mention that even if the reputation of a service from an external source is not available, we can obtain it gradually by receiving direct feedback from the user.

(c) **Obtaining overall gain of a service:** In this step, we reevaluate the gain that is computed using Eq. (5.4) or (5.5). The reevaluation is based on the perspective that a particular media service (e.g. a movie) will have less gain to a user if it has been selected for his/her in the past. In particular, we need a mechanism that would restrict the selection of some media services multiple times in a row. Nevertheless, in the ambient context, this is not true for all services. There are cases when the media services would have no decay in gain (e.g. RSS feed, music). We define some rules in order to handle these cases that reflects the user's requirements.

5.3 Media Service Selection

Once we obtain the gain score for the services in a particular context, the pervasive environment system may either choose a single service or a set of media services for the user that would maximize his/her overall gain. Therefore, we need a mechanism to fuse the gain of multiple services, which we explain in Section 5.3.1. As mentioned before, the selection of a service also depends on its cost and hence in Section 5.3.2 we give an indication of how the cost of a service can be computed. We then provide the details of how to select a single service or a set of services.

5.3.1 Gain Fusion

Gain fusion refers to the process of combining the gain obtained from two or more services when they are offered simultaneously to a user. For example, let a music service and an email service individually provide a gain of 0.60 and 0.70, respectively; the overall gain should be more when they are used simultaneously.

To fuse the gain of any n number of services, we adopt a normalized summation strategy that is commonly used as an aggregation method. For the gain fusion approach, we first normalize the gain of all the n services and then compute the overall gain by summing up the normalized gains of the individual services. The gain normalization also ensures that the sum of all the normalized gains equals 1. The normalized gain of a service S_i , $1 \leq i \leq n$ is computed using the following formula:

$$\tilde{g}_i = \frac{g_i^{C_x}}{\sum_{k=1}^n g_k^{C_x}} \quad (5.6)$$

where, \tilde{g}_i is the normalized gain of using the service S_i .

Now the overall fused gain of using the r number of services together would be computed as follows:

$$G_{group} = \sum_{i=1}^r \tilde{g}_i \quad (5.7)$$

This gain fusion approach will be used later when multiple services need to be selected together.

5.3.2 Cost Computation

Besides gain, our service selection mechanism also considers the cost of using a media service. This is done in order to have a trade-off between the gain and the cost to provide a cost-effective set of services to the user in the pervasive environment. The cost of using a service may be incurred in various forms, such as the cost of media subscription, energy consumption, internet connection and so forth. Here, we describe how the cost of a particular media service can be estimated. For this, let us consider the case of a movie service. Typically a movie service may be of different types, for example, new releases, library titles and adult titles. We may have varying cost for a movie belonging to one

of these types. We may also consider the length of the movie and calculate the internet connection cost or even energy consumption cost by determining the cost per unit time. Note that the cost of energy would depend upon on the energy consumption of the device which is used to play or display the service; for example, playing a movie on a laptop or normal TV may be cheaper compared to playing it on a big-screen display device which consumes more power. We illustrate the computation of the cost of selecting a movie service with the following example. Let the price of a new release movie of length 90 minutes be \$3, and we opted to play it on big display that consumes energy of \$0.02 per minute. In this case, an estimated cost of selecting this movie would be $\$3 + 90 \times \$0.02 = \$4.8$. In a similar fashion the cost of using other services may be computed.

It is important to mention that there are some services whose cost would be unavailable. For example, the weather forecast and web email are freely available and the cost of running those services would be virtually negligible given a fixed Internet connection exists. Nevertheless, even for these services, their energy consumption cost could vary depending upon the device on which they are played or displayed. Moreover, the cost becomes an important issue in a pervasive mobile environment where the cost of using the Internet is presumably high.

5.3.3 Selection of a Single Service

We propose to use a greedy strategy where only one service is to be selected from a pool of services. As each service provides some gain to the user at some cost, the greediness of having more gain at less cost would provide better results compared to being greedy about only maximizing gain or about only minimizing cost. Therefore, we propose to select a service that has a maximum gain/cost ratio given the cost constraint.

Precisely, we select a service $S_q, q \in \{1, 2, \dots, n\}$ such that,

$$\frac{\tilde{g}_q}{Cost_q} = \max \left\{ \frac{\tilde{g}_i}{Cost_i} \mid 1 \leq i \leq n, Cost_i \leq Cost_{spec}, Cost_i \neq 0 \right\} \quad (5.8)$$

where, n is the number of services. Note that, in the case a service cost is negligible to zero or the user does not want the system to consider service cost, a service with highest gain will be selected.

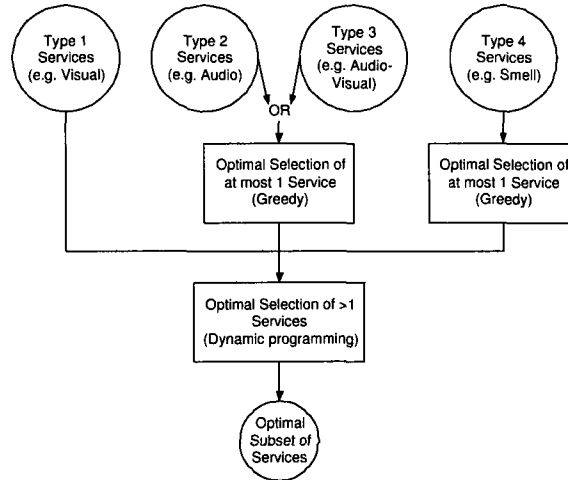


Figure 5.5: Example schematic view of the process of optimal service selection: A hierarchical approach for $k = 4$ types of services

5.3.4 Selection of Multiple Services

As mentioned earlier in Section 5.1, ambient media services may be of k different types and each of them might have different requirements for the output interface, and not all types of services may be executed at the same time to have better perception. For example, as shown in Figure 5.5, multiple services can be selected from visual type services because a user can comfortably perceive more than one of this type in a given interface. However, only one service can be selected from either audio or audio-visual groups, as selecting two from this group at the same time would be distracting. In the case when only one service is to be selected, we adopt a greedy strategy as explained in Section 5.3.3. On the other hand, when we need to select multiple services we adopt a dynamic programming solution. We model the problem of ambient media service as a gain and cost based Knapsack Problem (KP) for obtaining an optimal set of services. The Knapsack Problem is *NP*-Complete and may be solved using greedy as well as dynamic programming based methods [74]. Using dynamic programming approach, which can provide a solution in pseudo-polynomial time.

The dynamic programming approach for approximating an optimal subset of services Φ can be computed by formulating a recursive relation. Adapting the general solution of

KP to our needs, we define a recursive function $gain(i, m)$ to be the maximum value that can be obtained with $Cost \leq m$ by considering services up to i , which is given below:

$$gain(i, m) = \begin{cases} gain(i-1, m) & \text{if } Cost_i > m, \\ \max[gain(i-1, m), (gain(i-1, m - Cost_i) + \tilde{g}_i)] & \text{if } Cost_i \leq m \end{cases}$$

The above formula is iteratively used to construct a table $gain(i, m), 1 \leq i \leq r, 1 \leq m \leq Cost_{spec}$ for approaching a solution to the service selection problem. A cell $gain(i, m)$ in this table roughly approximates the fused gain for better user experience based on service 1 to i with the cost limit m . The fused gain of services 1 to $i-1$ with a cost constraint m is referenced here by the cell $gain(i-1, m)$ of the table and is computed using the approach explained in Section 5.3.1.

The initial conditions to be set for the recursive relation are as follows:

$$gain(1, m) = \begin{cases} 0 & \text{if } Cost_i > m, \\ \tilde{g}_1 & \text{if } Cost_i \leq m \end{cases}$$

We approximate the overall gain by recursively computing $gain(r, Cost_{spec})$. To do this efficiently we construct a gain table to store the values of previous computations. The subset of services is then computed by backtracking through the table.

5.3.5 Overall Subset of Services

The overall subset Φ of services using our approach would be either a single service or a set of services, which will be suggested or executed at a certain context depending on the level of automation in the environment. However, based on a context change (such as, new user identification), the system will re-compute the subset and provide it to the user in due course.

5.4 Experiments with Service Selection

The proposed service selection approach described here is integrated in the overall framework that we will describe in the next chapter. In this chapter, we just show how to

compute gain of context-aware services and utilize their cost to select the best set of services for a user in a particular context.

5.4.1 Sample Implementation

We have implemented the proposed methodology in the context of a smart environment. In our earlier work [55], we have developed an infrastructure and a smart mirror interface for accessing various information and appliance services in a smart lab environment. We have successfully shown how the different types of services can be accessed together by the authorized user in the environment. However, in [55], we have not adopted any formal service selection mechanism, which justifies the work in this chapter.

The prototype is developed using the C# programming language and finally it has been wrapped in a web service so that it can be utilized by our framework. For the capturing of a particular context, we adopted the mechanism described in Chapter 3.

5.4.2 Gain Computation

For experiments with the prototype system, we have first asked some users to provide their media preference values and we have stored them in the AMP corresponding to the user. Some users have opted not to provide any information regarding their preference, while others volunteered to do so in order to see how the system reacts. We have allowed five users to access the system in the smart environment for a week and collected their interaction history data with their permission. Please note, while collecting interaction data, the proposed method was also applied to dynamically update their AMP scores, which were used to compute gain and select services.

In order to explain gain computation, we first show how the user's contextual media preferences change and accordingly how the AMP scores are updated. When the user interacts with the system, their interaction history is kept in the interaction history repository. Based on the interaction data, the AMP scores are updated using Eq. (5.2) and (5.3). In Figure 5.6, we only show few instances of AMP updates for clarity.

It is clear from Figure 5.6 that the scores in AMP are updated after every interaction. These scores reflect the changing preference of a user in the different metadata attributes

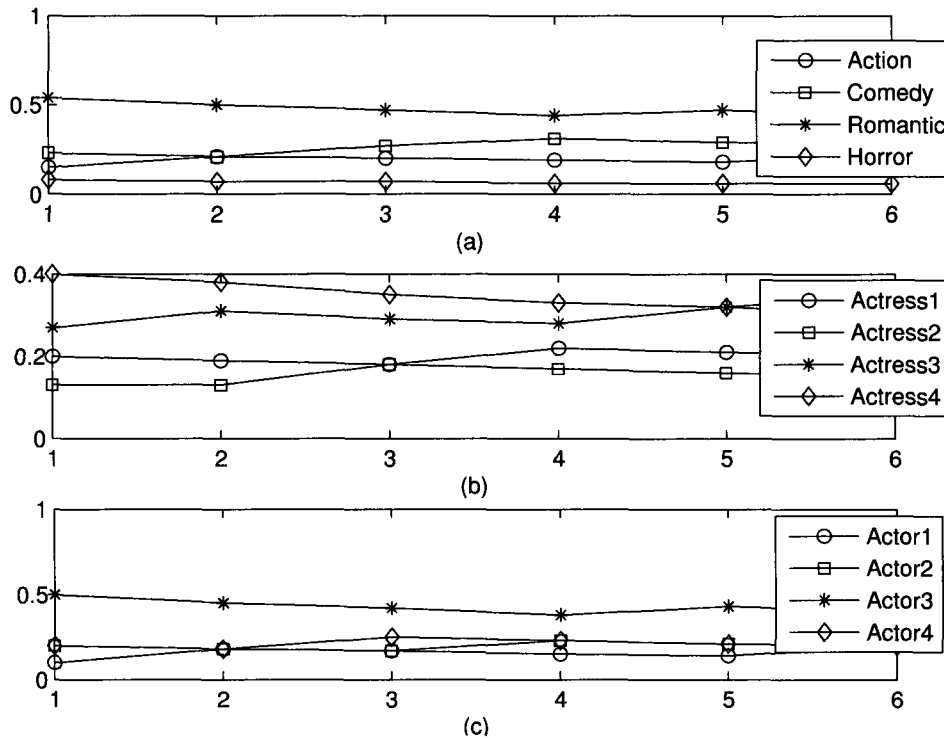


Figure 5.6: Changes in AMP scores after movie service usage by a user in a particular context. Here, (a) refers to the changes in genre, (b) changes in actress and (c) changes in actor attributes.

of the media services. For example, at the second interaction, the user selected an action movie with actress2 and actor1. Therefore, the AMP scores corresponding to these attribute values are incremented at this interaction while the scores of the other attribute values are decremented. Note, although in this figure we only show the scores updates of movie attributes, we applied our proposed mechanism to update AMP scores that correspond to the attributes of other media services, such as music and news.

The preference update process is followed by the gain computation process. As we mentioned, the gain of a service is based on AMP scores and the reputation of that particular service. In our model, we compute the preference scores, while the reputation score is assumed (e.g. we assumed the reputation of four movie services as .60, .45, .70, .65, respectively). We use Eq. (5.5) to obtain the gain of a service dynamically, where the value of α is assumed to be 0.5 to assign equal weight to the AMP-based gain and

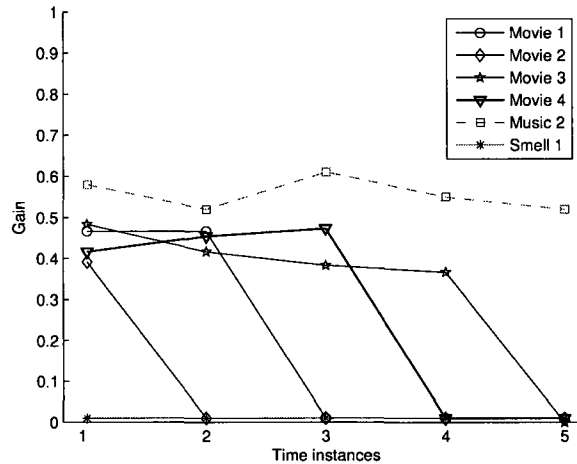


Figure 5.7: Example of a case showing the evolution of gain of different types of media services in a particular context.

the media service reputation. A different weight score can be assigned to α depending on the importance the user or the system wants to put on AMP or the reputation. In Figure 5.7, we show how the gain value changes in the case of different types of services over a period of similar contextual situations.

As shown in Figure 5.7, the gain of the media services of a particular type (e.g. movie) has evolved since its first usage. For example, movie4 has been seen by a user at time instant t_3 and hence its gain has been set to a negligible value referring to less interest in that service at a subsequent instance. On the contrary, for some media types (e.g. music) our model does not enforce its gain to be declined even after its usage. This is why the bars representing the gain of music2 show usual changes in scores. Another case is shown in Figure 5.7 for a particular type of media service (e.g. smell), which has negligible gain as there is no attribute related to that media present in the AMP.

5.4.3 Gain-cost based Service Selection

We now show how the dynamically computed service gain at a particular context is used to select relevant services for a user. As described earlier, our model uses service cost along with the gain value for the selection of services. This also allows us to obtain

Table 5.2: Data set used in the experiment showing different services, the associated gain and cost.

Group	Service label	Gain (g_i)	Cost (c_i)
<i>Visual</i> (Type1)	S_1 : Service-1	0.33	2
	S_2 : Service-2	0.54	1
	S_3 : Service-3	0.67	3
	S_4 : Service-4	0.70	2
	S_5 : Service-5	0.88	3
	S_6 : Service-6	0.42	4
<i>Audio</i> (Type2)	S_7 : Service-7	0.24	3
	S_8 : Service-8	0.58	5
	S_9 : Service-9	0.30	1
<i>Audio-visual</i> (Type3)	S_{10} : Service-10	0.72	4
	S_{11} : Service-11	0.40	8
<i>Smell</i> (Type4)	S_{12} : Service-12	0.66	6
	S_{13} : Service-13	0.58	2
	S_{14} : Service-14	0.48	3

a set of services that can be considered as an optimal set. To demonstrate this, we consider a set of $n = 14$ services as listed in Table 5.2. The cost associated with each service is determined by following the procedure described in Section 5.3.2. The gain of the services is computed for a particular context using our proposed method in the experimental system setup. Using this test data, we will now show how the service selection mechanism works.

We first compute the normalized gain of the different types of services using the Eq. (5.6). Thus the normalized gain of the services becomes $\{\{0.04, 0.07, 0.09, 0.09, 0.12, 0.06\}, \{0.03, 0.08, 0.04\}, \{0.10, 0.05, 0.09\}, \{0.08, 0.06\}\}$. We now consider the maximum acceptable cost $C_{spec} = 9$ as a test case scenario. Using the cost data in Table 5.2 and applying the equation (5.8), we compute the normalized gain per cost ratio of Type 2 (e.g. audio) and Type 3 (e.g. audio-visual) services, which is $\{\{0.01, 0.02, 0.04\}, \{0.02, 0.01, 0.01\}\}$. The obvious selection of one single service from these two types of services is the one with the maximum ratio that is 0.04 and the corresponding service is S_9 with a gain of 0.30. Similarly, for service Type 4 (e.g. smell), we apply the greedy approach and select one service with the maximum normalized gain per cost ratio as $\max\{0.04,$

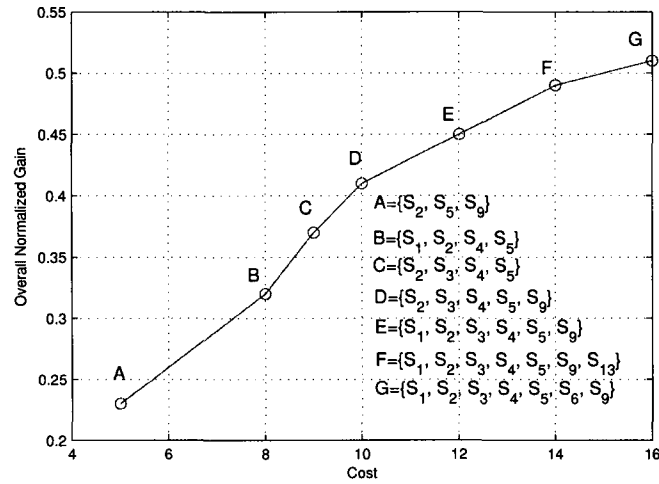


Figure 5.8: Optimally selected subset of services that maximizes the gain, subject to different cost constraints. Here, the input to the selection algorithm is the set $\{S_1, S_2, S_3, S_4, S_5, S_6, S_9, S_{13}\}$, and the symbols A to F denote the selected optimal subsets for different cost constraints.

0.02}. We select the maximum ratio that is 0.04 and the associated service is S_{13} with a gain of 0.58. Note that the above selection also satisfies the given cost criteria.

Now, we merge the two shortlisted services S_9 and S_{13} with the list of visual services (Type 1) and apply the dynamic programming algorithm on the merged set for determining the final subset of services. With $C_{spec}=9$, we obtain the overall subset of services as $\Phi=\{S_2, S_3, S_4, S_5\}$ and the overall normalized gain as $G_\Phi=0.37$ with $C_\Phi=9$.

We repeat the test by changing C_{spec} and obtain different subsets of services. Figure 5.8 shows the optimally selected services at varying specified costs. In this figure, the points A, B, C, D, E, F, G show the different computed optimal subsets with different overall gain G_Φ and cost C_Φ . It is also visible from this figure that in case a service is not available, the next best subset can be chosen. For example, when $C_{spec} = 8$, the optimal subset is $\Phi=\{S_1, S_2, S_4, S_5\}$ at point B and if S_2 is not available from this set, the next best optimal subset would be $\Phi=\{S_2, S_3, S_4, S_5\}$ referring to the point C in Figure 5.8.

Figure 5.9 shows a sample screen shot of the smart mirror interface with an optimally selected subset of services, where $C_{spec} = 9$ and $\Phi=\{S_2, S_3, S_4, S_5\}$. The screen shot shows that 4 visual services have been provided to the user using the mirror interface in

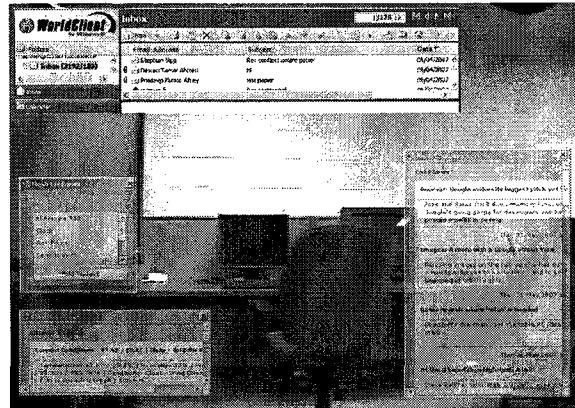


Figure 5.9: Service provided via a smart mirror interface. The optimal service selection method selected news feed, weather feed and an email client at a particular instance. In addition, the system shows the list of available services in case the user wants to select some other services manually. Please note, some of the services might not have any visual output interface. In that case, the selected services will be provided using its corresponding suitable device. In this particular case, the mirror functionality has been disabled by the user.

a particular context. Although the smart mirror itself would usually show the live video feed of the person in front of the mirror [55], unless the person turns the video off.

5.5 Summary

In this chapter, we have presented a novel approach for the selection of ambient media services in pervasive environments. We have used the concept of gain in a service in a particular context as a key factor that derives the service selection process. Our method follows a user-centered approach to incorporate the user's context, their ambient preference, interaction history and service reputation to dynamically compute gain of the available media services. We have associated the gain value with the potential service usage cost and proposed a combination of greedy and dynamic programming based solution to obtain an optimal set of services that would maximize the user's overall gain in a particular context. Initial experiments have been provided to explain our methodology.

In the next chapter, we show how the different aspects of our adaptive interaction model have been integrated in a single framework.

Chapter 6

Framework Integration

In the previous chapters, we have explained how to model and assess quality of context information in a multi-sensor environment, how to dynamically determine the level of implicit interaction in such an environment and how to select a set of context-aware services based on the dynamically adjusted levels. In each of these cases, we have proposed and adopted various models and mechanisms to address the relevant problems and performed quantitative analysis of some sort. However, an integrated view of the overall interaction framework has not yet been provided. In this chapter we aim to cover this gap. In particular, we show here the different architectural components that form the building blocks of the framework, provide detail system dynamics and elaborate on the implementation aspect.

6.1 High-level Architecture

Based on the overall goal of this thesis as stated in Chapter 1 and consequent presentation of the methodologies in Chapter 3 through Chapter 5, we draw a high-level architecture of the proposed interaction framework for pervasive environment in Figure 6.1. Note the design follows a distributed Service Oriented Architecture pattern. Every component in this architecture is responsible for a particular task. We now describe these components and show how they embed the contents described in earlier chapters.

Context Identifier: In the human-environment interaction scenario, context iden-

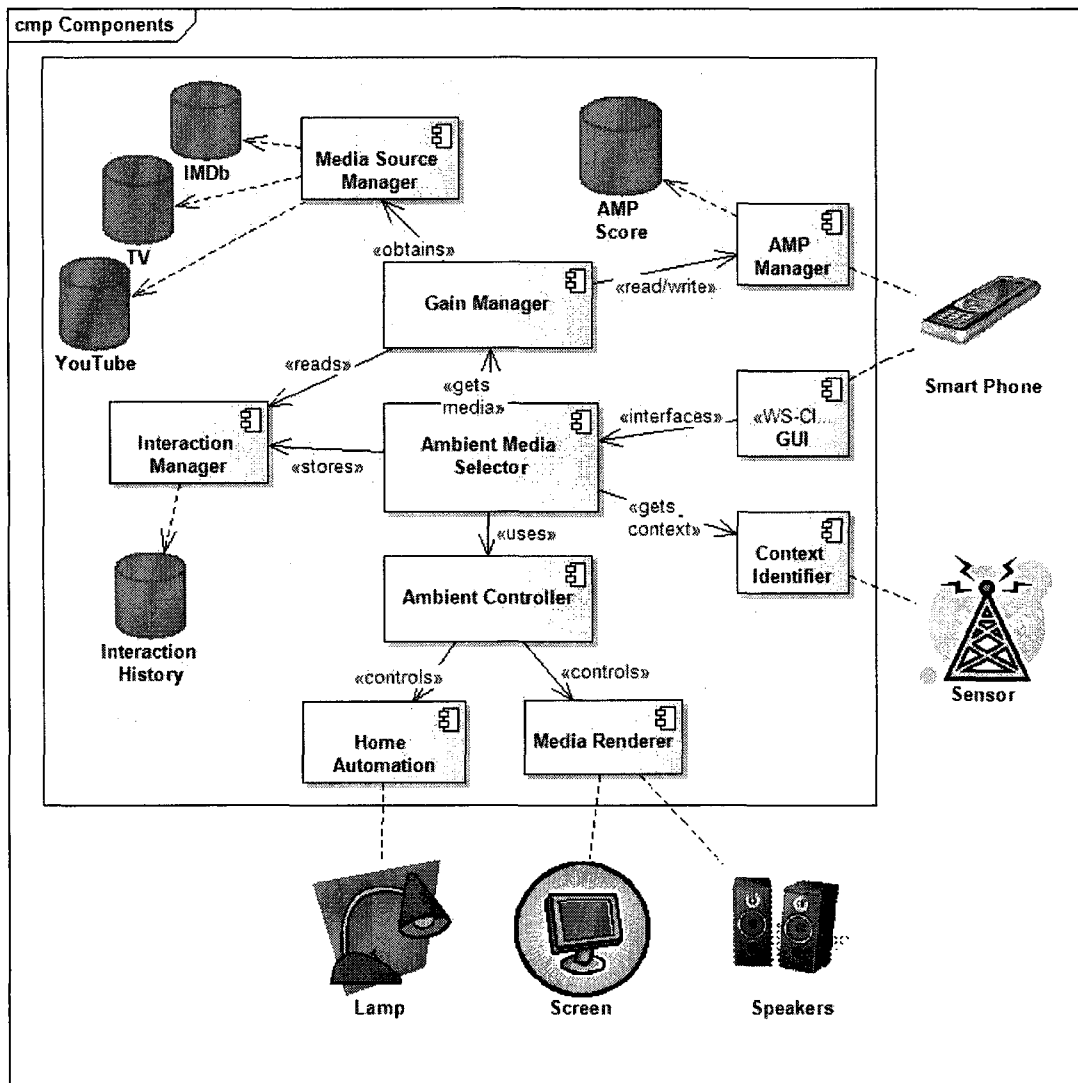


Figure 6.1: Architecture of the proposed context-aware interaction framework

tification plays a key role. A pervasive environment system inherently captures the context of the user, which is considered as input, in order to provide implicit support to the user with relevant services. However, as the sensor-driven context identification provides imprecise and often ambiguous observation, we showed in Chapter 3 a mechanism to estimate the quality of context information on-the-fly. Based on this mechanism the Context Identification component remains responsible to identify the current context as well as to compute QoI, which is then used by the corresponding module.

AMP Manager: The Ambient Media Preference (AMP) Manager is responsible to manage the AMP scores in separate repositories. Any component that needs to access or update the AMP scores must request the AMP manager to get a copy of the AMP scores. The process using which AMP scores are updated were described in Chapter 5. Note, as the proposed framework support user's mobility in the environment, we preserve the AMP scores in an XML repository that resides in the user's mobile device. This approach helps to avoid some of the privacy critiques related to personal preference handling.

Media Source Manager: The media source manager connects to many online media sources as well as discovers other available services in the environment. The components that needs to know the list of available services contacts the Media Source Manager, which returns the list along with the individual service properties.

Interaction Manager: The Interaction Manager is responsible to manage the history of user's interaction along with the timestamps. The user interactions usually contain information related to the type of services and the context of their use. The Interaction Manager provides the facility to obtain the list of interactions by any authorized component and enables updating the list with new interactions.

Home Automation Manager: This component enables the control of household smart appliance services. One of the main reason to separate this component is to hide the complexity of connecting to different devices' services and to provide a flexible means to access them when necessary.

Media Renderer: The Media Renderer component handles the issue of setting up the rendering related components, i.e. the display, speakers and their volume levels for a particular context situation.

Ambient Controller: This acts as a controller of Home Automation Manager and Media Renderer. Hence, in order to invoke a service or recommend services to the user, this component informs the Home Automation Manager and Media renderer to carry on a particular tasks.

Gain Manager: The Gain Manager when contacted performs the important task of estimating the gain of a service in a particular context as described in Chapter 5. The Gain Manager contacts the Media Source Manager and AMP Manager to calculate gain of the available services, which is passed to the Ambient Media Selector when asked.

Also the Gain Manager contacts the Interaction Manager when it needs to update the AMP scores based on the interaction history. The updated AMP scores are later used for gain estimation in the subsequent intervals.

Ambient Media Selector: The goal of an interaction between a user and an environment is to select relevant services for the user in the current context. There are two ways by which an interaction may be triggered, first, by a user through explicit request and second, by the environment on detecting a change of context. In both cases, the Ambient Media Selector component gets the current context along with QoI parameters from the Context Identifier and attempts to select the best services for the user. To accomplish this task, it also communicates with other components that will perform their respective tasks when contacted. For example, the Ambient Media Selector obtains the context and quality information from the Context Identifier and based on this information it asks the Gain Manager component to return relevant services based on the computed gain. It then determines the level of action as per the mechanism described in Chapter 4 and contacts the Ambient Controller to take appropriate actions, that is either invoke the selected services or provide suggestion to the user in their environment. It also contacts the Interaction Manager to update any new interactions.

6.2 Component Communication for Interaction

The proposed framework supports an adaptive implicit interaction scheme, which leads to mixed-initiative interaction with the pervasive intelligent environment for accessing various services of interest. It also facilitates explicit interaction with the environment with a view to provide better control to the user. Explicit interaction is initiated by the user who uses some sort of interface to explicitly ask for any service. In the implicit mode of interaction, the environment acts based on the automatic context identification. The mixed-initiative interaction occurs in the co-operation between human and the environment. We further describe these different mode of interaction in the following.

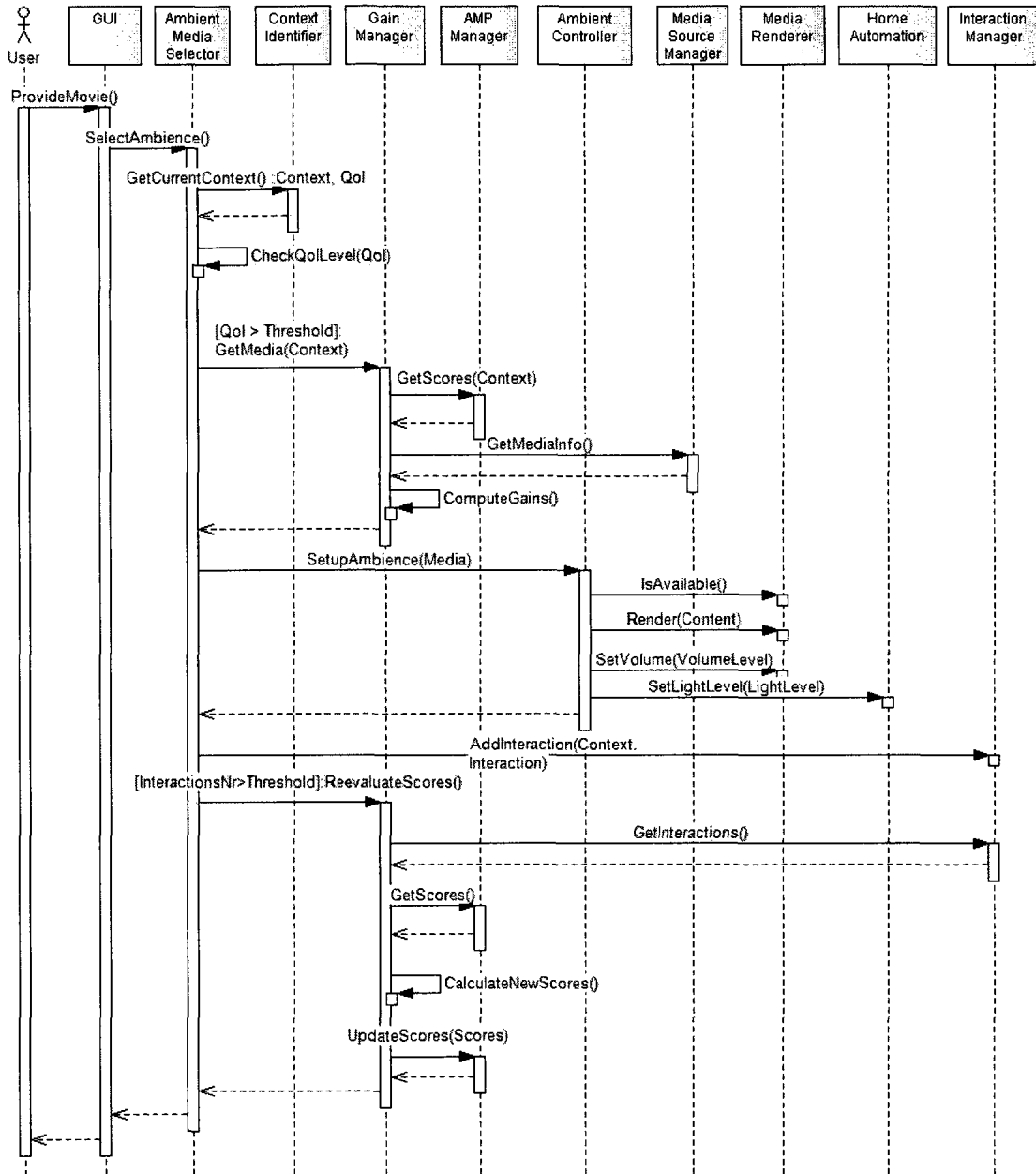


Figure 6.2: Sequence of interactions that take place among the different components in “explicit” mode of interaction

6.2.1 Explicit Interaction Support

In this mode, the user explicitly initiates an interaction by using a GUI interface. In such an explicit interaction mode, the user might only want to specify that he/she wants a particular type of service, for example, a movie. The system in response interacts with the corresponding components and selects the best movie for the user depending on the gain of the service. To serve user's request, a lot of interactions happen between the components of the architecture, which is shown using a sequence diagram in Figure 6.2. In this figure, we show that the user asks for a movie through the GUI. The GUI passes the request to the Ambient Media Selector, which communicates with other necessary components to provide the right movie that fits the current context. Note, the current context identification process includes the QoI computation and if the QoI level is high, the rest of the media selection process is carried out by the system automatically.

6.2.2 Implicit Interaction Support

In the implicit mode of interaction, the environment takes the initiative to provide service support to the user. Accordingly, the system automatically identifies context and compute QoI. Based on the QoI values, the system dynamically determines whether to activate some services, suggest some actions or display relevant information as a clue to the user. This process is shown in Figure 6.3. Also, the decision taken by the implicit interaction handler based on varying QoI leads to a mixed-initiative interaction mode, which we discuss shortly.

6.2.3 Mixed-initiative Interaction Support

Mixed-initiative interaction mode is activated as a result of dynamic adjustment of interaction based on QoI. The ultimate goal is to provide as much implicit support as possible so that user's explicit interactions can be minimized but is allowed when required. Therefore, while providing implicit interaction support to the user, if the QoI value is not high enough to automatically invoke the selected services, the system dynamically determines the next level of interaction. In such cases, the system adjusts itself to support the next best interaction level, for example, provide recommendation

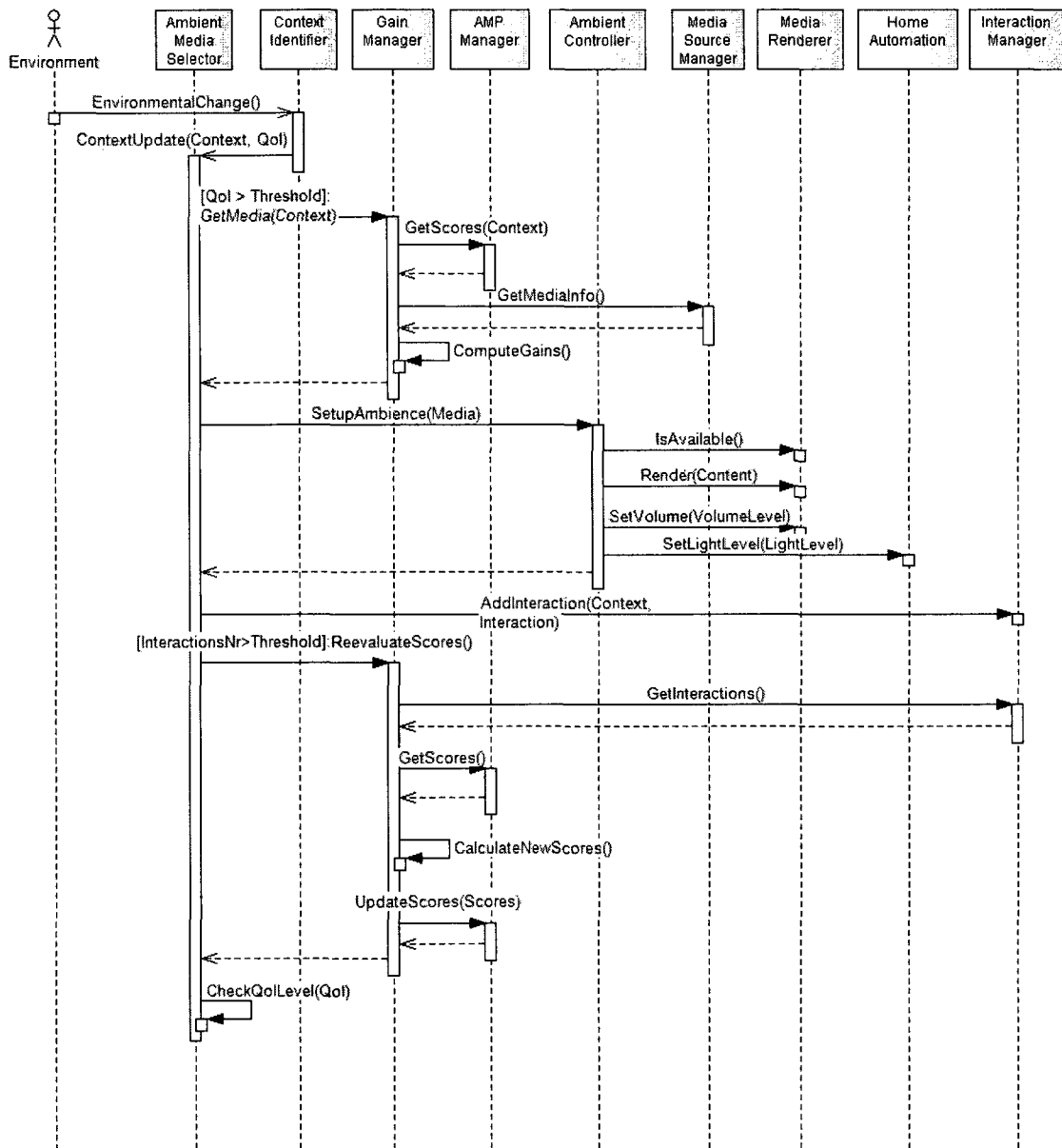


Figure 6.3: Sequence of interactions that take place among the different components in “implicit” mode of interaction

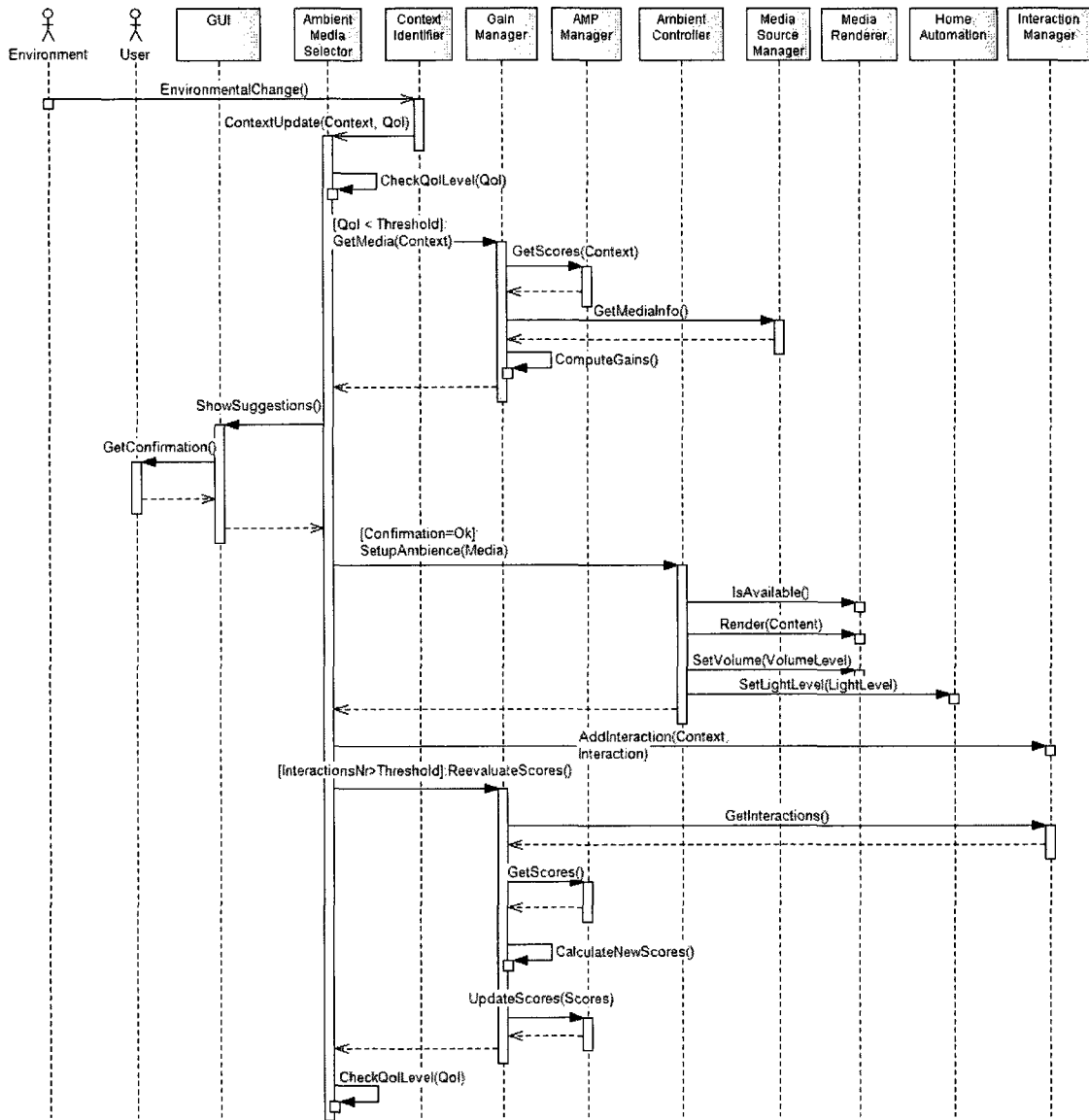


Figure 6.4: Sequence of interactions that take place among the different components in “mixed-initiative” mode of interaction

instead of execution. The adjustment of interaction to recommendation involves the user as he/she has to interact further to confirm the suggested actions by the environment. The communication that takes place among the components, user and the environment in the case of a mixed-initiative interaction is shown in Figure 6.4.

6.3 Implementation

This section describes the implementation details of the proposed framework. Consequently, Section 6.3.1 describes the experimental environment and the different devices connected to the environment. Finally, the development of different services are elaborated in Section 6.3.2.

6.3.1 Environment Setup

Figure 6.5 shows the layout of the experimental smart environment and the different devices that are in that environment. In this environment, all the rooms are equipped with X10 [4] lamps allowing the user or the system to control the light level. The environment is also equipped with several cameras, motion sensors, and RFID readers. Table 6.1 lists the different devices that are in the environment, which we use to conduct our experiment.

Besides the devices listed in Table 6.1, all the networked devices are connected to a router using Ethernet (Media Center + Kitchen PC) and alternatively using WiFi connection (Laptop, SmartPhone and Ultra Mobile PC).

6.3.2 Development of Different Component Services

In our implementation we adopted a Web Service based design approach so that different technology platform can be made interoperable. More specifically, we used .NET deployment framework tools developed in the IST Amigo Project [1] to implement the Web Services and clients, supporting WS-Discovery [43] and WS-Eventing [42] protocol. We adopted SOAP/XML Web Services as the communication middleware.

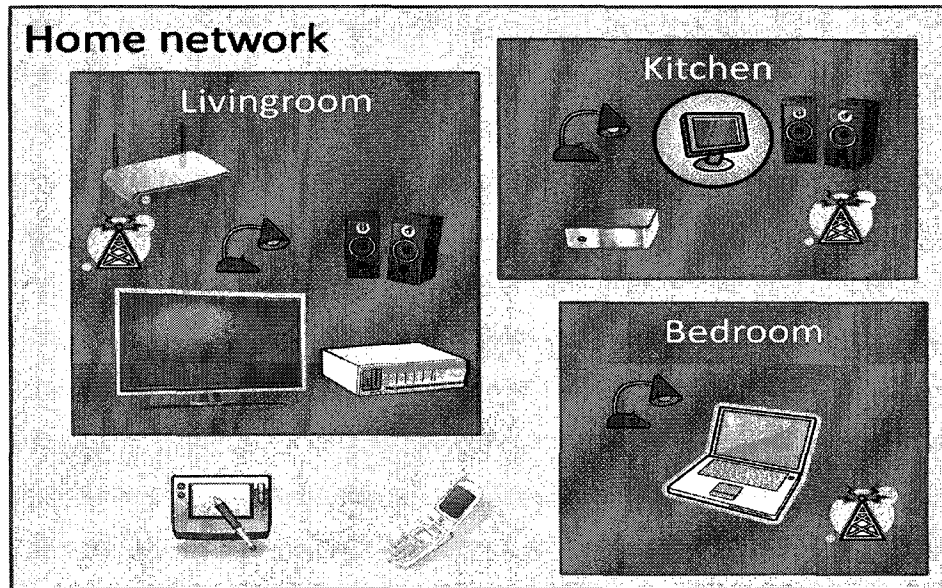


Figure 6.5: The layout of the experimental environment.

Table 6.1: The devices/resources in the experimental smart environment.

Location	Devices	Description
Living room	Media Center	Fujitsu-Siemens Scaleo E, running Windows XP Media Center Edition 2005
	Speakers	Ambient sound
	40" TV screen	Media renderer
	Cameras (2)	Context sensor
	Motion sensor	Context sensor
Bedroom	Dell Latitude D520 Laptop	Portable device
	Camera	Context sensor
	Motion sensor	Context sensor
Kitchen	19" TFT Monitor	Media renderer
	Speakers	Ambient sound
	PC	X10 home automation interface is connected to this PC
	Camera	Context sensor
	Motion sensor	Context sensor
Portable location	Samsung Q1 Ultra Mobile PC	Used as a mobile renderer
	HTC Touch Cruise Smart Phone	User interface, also contains AMP

The cornerstone of our prototype development is that all the components of the proposed framework are developed as dynamic web services that conforms to a dynamic SOA-based architecture. This also enables us to realize a loosely-coupled system, which is flexible in terms of extendability and maintenance. For example, a service's business logic can be changed separately without breaking its interface with other services.

The adoption of WS-Discovery facilitates dynamic discovery of the component services that is a basic feature for successful system integration. WS-Discovery defines a multicast discovery protocol that enables Web Service clients to locate Web Services in the network that are identified by type and/or scope attributes. The Type and Scope attributes define some metadata parameters that can be used to query a particular service. On the other hand, the WS-Eventing protocol we use describes how a client (Subscriber) can register to some events (Subscriptions) of a Web Service (Event Source). Thus, changes in the service can be notified to any client without requiring standard polling mechanism. The event delivery is accomplished using simple asynchronous messaging. This mechanism allows services to notify their clients, and become active. When a change in the service occurs, the server initiates a new communication by sending a message to the clients (Subscribers). To improve robustness, a leasing mechanism is defined so that when an event source accepts a request to create a subscription, it typically does so for a given amount of time. In our implementation, a component service is used in two different ways: as a client to other services and as a service for other clients.

Figure 6.6 also shows how the connections among the components have been implemented using Web Service paradigm. The service components offer a Web Service port as exposed interface of internal business logic, and the client components use Web Service clients as proxies of the desired service. For the sake of simplicity, we have only shown the business logic part in some of the components in this figure.

We now briefly illustrate the development of different component services of the framework as discoverable web services, which are Context Identifier WS, AMP Manager WS, Interaction Manager WS, Home Automation WS, Media Renderer WS, Media Source Manager WS, Ambient Controller WS, Gain Manager WS, Ambient Media Selector WS and Ambient Media Selector GUI WS.

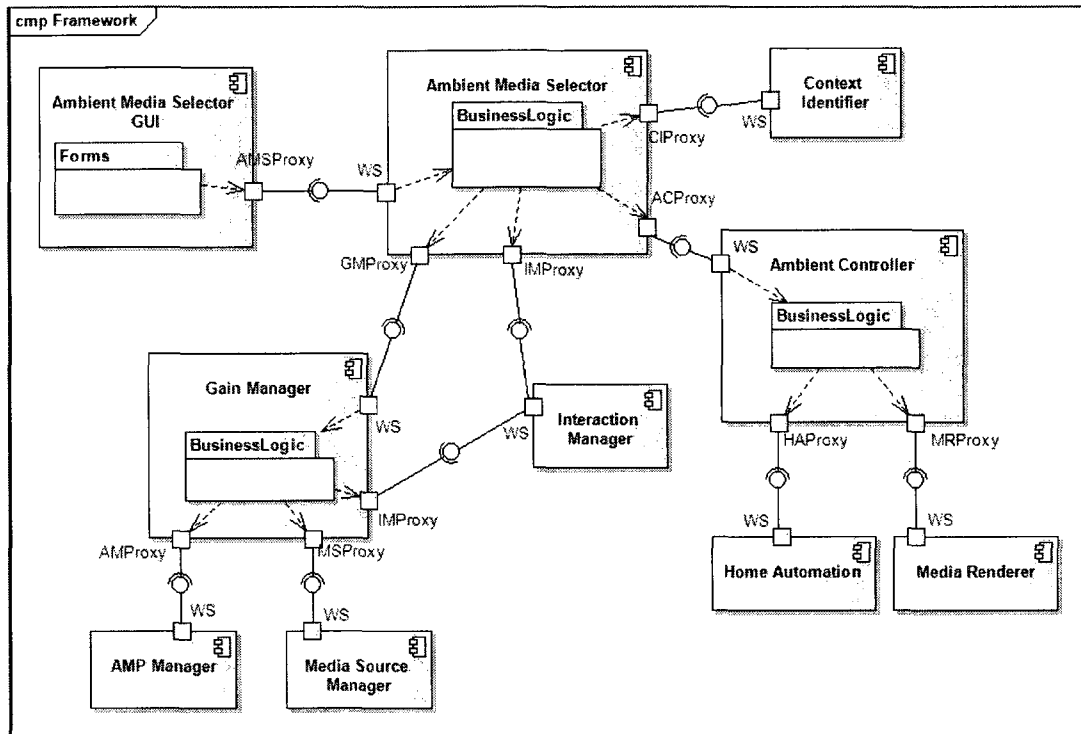


Figure 6.6: The interconnection among the different components.

6.3.2.1 Context Identifier WS

Our Context Identifier WS is developed based on the processing of camera, motion sensor events and RFID data. The processing of context information items (events) are done according to the mechanism described in Chapter 3. Primarily this service provides information such as who is around, their location and time. The identification of such information items along with QoI attribute values are propagated to the web services that subscribe to its events. Therefore, this service does not depend on the availability of other services, rather the subscriber web services depend on the events it propagates. In Figure 6.6, we see that Ambient Media Selector WS is a client for this service and receives context events whenever any change is detected in the environment.

6.3.2.2 AMP Manager WS

Like Context Identifier WS, this also acts as a separate service which facilitates the management of AMP scores in the repository. Figure 6.6 shows that the Gain Manager WS is a client of this service. Therefore, the Gain Manager WS sends SOAP/XML message to the AMP Manager WS via a proxy whenever it requires to access AMP scores. The response from AMP Manager WS is propagated as SOAP/XML message in return.

6.3.2.3 Media Source Manager WS

It is developed to integrate different media sources such as You Tube, IMDB, TVDB and any personal media repository. The Gain Manager WS access this service via a client proxy and in response it returns metadata information of the available media sources using SOAP/XML messages.

6.3.2.4 Interaction Manager WS

This service almost has the same functionality as the AMP Manager WS, except it facilitates reading the interaction history from the interaction repository and storing new interactions data into the repository. The Ambient Media Selector WS and Gain Manager WS both act as clients to this service, which is shown in Figure 6.6.

6.3.2.5 Gain Manager WS

There are several methods that are wrapped under this service and hence it performs multiple tasks. First, it updates user's AMP scores based on the interactions history record. The underlying business logic that updates such score is defined in Chapter 5. Second, it collects rating data from the web (e.g. IMDB movie database) and uses this rating (if available) with the updated AMP scores to compute gain in a media service. The gain computation is based on the approach described in Section 5.2. Therefore, the Gain Manager WS communicates with several other services. It has both the service role as well as client role. For example, it is client for AMP Manager WS, Media Source

Manager WS and Interaction Manager WS whereas it is a dependable service for Ambient Media Selector WS as depicted in Figure 6.6.

6.3.2.6 Home Automation WS

We use X10 [4] lamp modules and the ActiveHome CM11A Computer Interface to develop a ASP.NET Web Service that facilitates the lighting control in each room. Thus, the lighting level of each room can be set individually using standard Web Service protocols. Figure 6.7 shows the code snippet of the web methods in the web service for home automation. The Ambient Controller WS defines a client proxy to invoke the functionality of this service.

6.3.2.7 Media Renderer WS

The Media Renderer WS in our design has been implemented using VLC [3]), which is a highly portable multimedia player supporting various streaming protocols, audio and video formats. VLC integrates a little HTTP server that can be used for a HTTP remote control interface. We developed a ASP.NET Web Service wrapper that controls the VLC player using VLC's HTTP interface. Figure 6.8 shows the code snippet that is used by the media render web service to control the renderer. Similar to Home Automation WS, the Ambient Controller WS is a client of the Media Renderer WS.

6.3.2.8 Ambient Controller WS

This service acts as a gateway to access the Home Automation WS and Media Renderer WS. Like the Gain Manager WS, it also has both client and service role. The Ambient Media Selector WS is a client of this service.

6.3.2.9 Ambient Media Selector WS

This is the coordinator service and is a client of the Gain Manager WS, Interaction Manager WS, Ambient Controller WS and Context Identifier WS as shown in Figure 6.6. This service is considered as the entry point of ambient media service selection at home. While selecting services for the users, this WS also determines the dynamic level of

```
//Web service wrapper for X10 lighting control class
1. Declare web service header
    ([WebService(Namespace = "http://www.mcrlab.uottawa.ca/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)])

2. Declare public class X10LampWebService
of System.Web.Services.WebService
attributes: HouseCode, DeviceCode, CurrentLevel
methods: SetLightLevel of type void, SetDim of type boolean,
         SetBright of type boolean, getLightLevel of type int

3. Declare Web methods
    //Sets the light level of a given lamp
3.a Web method SetLightLevel(local parameter newLightLevel)
    Call X10 system controller:
        dllX10.SetLightLevel(this.HouseCode, this.DeviceCode, newLightLevel)

    //Dims the lightlevel to the given value
3.b Web method SetDim(local parameter dimValue)
    Call X10 system controller:
        dllX10.SetDim(this.HouseCode, this.DeviceCode, dimValue)

    //Increases the brightness of the light
3.c Web method SetBright(local parameter brightValue)
    Call X10 system controller:
        dllX10.SetBright(this.HouseCode, this.DeviceCode, brightValue);

    //Returns the current light level
3.d Web method getLightLevel()
    return this.CurrentLevel
```

Figure 6.7: Code snippet of the four web methods in the lighting control web service.

```
//Web service wrapper for media player
1. Declare web service header
    ([WebService(Namespace = "http://www.mcrlab.uottawa.ca/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)])

2. Declare public class VLCWebService of System.Web.Services.WebService

3. Declare Web methods
    //Plays the media context from the playlist
    3.a Web method Play()
        Step1--> Create HTTP request object:
            HttpWebRequest myReq = HttpWebRequest.Create
                (@"http://localhost:8080/?control=play");
        Step2--> Get the web response:
            HttpWebResponse myResp = (HttpWebResponse)myReq.GetResponse();
        Step3--> Close response object

//sets the media contents in the playlist
    3.b Web method SetMediaContent(local parameter ContentUri)
        Step1--> Create HTTP request object:
            HttpWebRequest myReq = HttpWebRequest.Create
                (@"http://localhost:8080/?control=add&mrl=" + ContentUri);
        Step2--> Get the web response:
            HttpWebResponse myResp = (HttpWebResponse)myReq.GetResponse();
        Step3--> Close response object
```

Figure 6.8: Code snippet of the two web methods in the media renderer web service.



Figure 6.9: Example output interface when the system is running in full automation mode. In this case, the selected movie has been played in a big-screen TV where the user is located and the news is shown on the computer screen.

output interaction based on QoI with a view to maximize user's satisfaction. For example, when the values of the quality attributes are high, the system automatically selects the relevant media and also notifies the user with appropriate interface for giving them more control. The system also shows the QoI attribute values and displays the current mode of interaction, which the user may choose to change. Figure 6.9 shows such an interface. Similarly, the system may adjust to a different interaction level depending on QoI values, when it may only suggest actions instead of invoking them.

6.3.2.10 Ambient Media Selector GUI WS

A Web Service client application with GUI in the user's SmartPhone has been developed that accesses the proposed framework and uses the Ambient Media Selection WS to support user with ambient media.

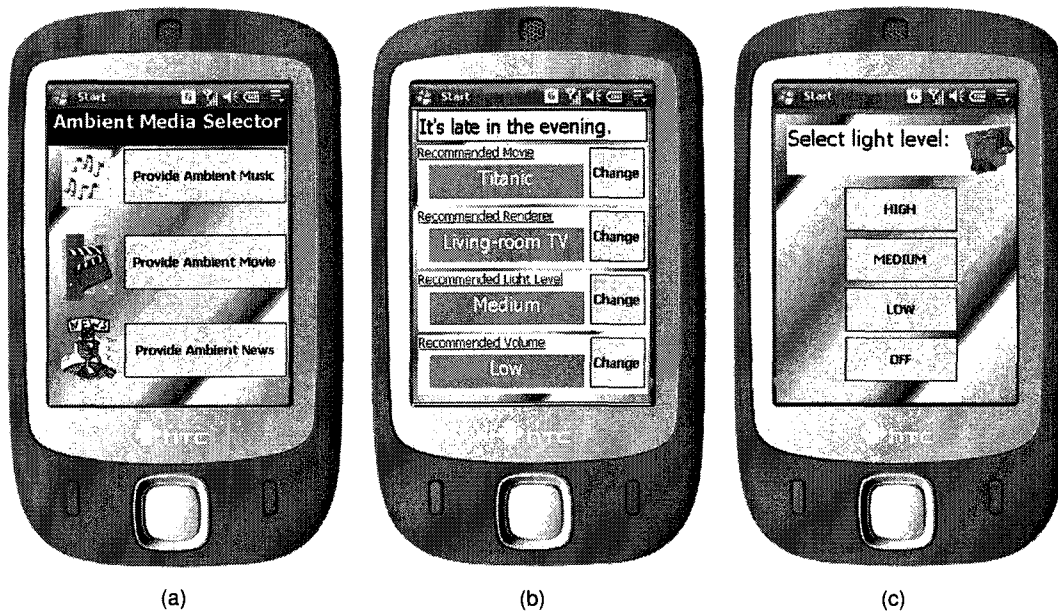


Figure 6.10: The user interface of the prototype system on a real HTC phone.

In addition to the output interface presented in Figure 6.9, we also developed sample input UI of the framework as presented in Figure 6.10. Here, the user initially sees the interface in Figure 6.10(a). In this screen the user may choose the type of media he/she wants. Once the user selects any of the options, the system automatically obtains the current context information and depending on the QoI value recommends the media of the selected type using the interface in Figure 6.10(b). There is a timing control at this stage such that if the user does not like to override the automatic selection, the system starts to deliver the media in the appropriate rendering device and sets the volume and lighting condition accordingly. Otherwise, the user may choose to override the automatic selection by changing the individual parameters. Figure 6.10(c) shows the case when the user wants to change the lighting level.

6.4 Summary

This chapter presented the design and implementation of the proposed framework and provided an integral view of the concepts and contents described in Chapter 3 to Chap-

ter 5. More specifically, we have drawn a complete view of a framework where different components are designed to do a particular task, for example, QoI estimation and gain computation. Furthermore, all the components are implemented as a discoverable web service in .NET framework for easier communication between component services. In the next chapter, we evaluate the developed framework to analyze the acceptability of the proposed interaction mechanism and justify how the overall framework increases users' satisfaction and minimizes their distrust.

Chapter 7

Evaluation

Earlier, in Chapter 3 we evaluated the QoI assessment model with sample data and in Chapter 5 we provided experimental result of the proposed gain-cost based service selection algorithm. In this chapter, we validate the integrated framework based on qualitative user evaluation. In particular, we measure 1) user acceptance as an indication of user satisfaction and 2) trust in automated system. Based on these measurement we want to validate that users's satisfaction and trust in such system will increase when the level of implicit interaction is dynamically adjusted with the changing quality of context information, which is in support with the thesis objective presented in Chapter 1.

7.1 Experiment Details

Here we provide details about environment setup, test procedure, and questionnaire used in the experiment.

7.1.1 Environment Setup and Scenario

Our experiment was carried out in a simulated smart home environment with prototype system implementation. The layout of the environment was shown earlier in Figure 6.5. The environment is equipped with several sensors, cameras, display devices and media center (as listed in Table 6.1). The sensors are used for context recognition, such as to identify who is the current user, his/her location and time of activity. Based on the home

automation technology, we facilitated the interconnection of the devices and sensors and realized different functionalities of media players as explained in Chapter 6. Instead of just using one service, we developed several services for this experiment and wrote test scripts to demonstrate the usage of those services within the context of some scenarios in a smart home environment. Example scenarios include:

1. SC-1: The system opens the door if an authorized user approaches it. However, it will depend on the correct identification of the user. Alternatively, it will be considered mis-automation if the system automatically opens the door even when it does not recognize the user with certainty.
2. SC-2: The system adjusts the lighting level of the room as per the user's preference. This situation occurs when the system identifies a user at a particular context, for example, in the living room on a weekday evening. The system in such case obtains the lighting level preference score from the ambient media preference based on the current context and sets the lighting level accordingly. However, if the context identification process results in low QoI and if such low QoI value is not considered when actuating the lighting service, it is possible for the system to mis-handle the case and set the lighting level that does not reflect the user's choice.
3. SC-3: The system selects appropriate media (movie, music or news) and renders them in location-specific media player depending on the profile and the position of the user in the room. In this scenario we consider different cases where users have the preferences to watch movie in a big screen TV and on a computer screen.
4. SC-4: When the user leaves home, the system closes the player, activate security cameras and starts monitoring the environment in usual mode.

Although the above scenarios will only demonstrate few of the functionalities among the whole spectrum of complex interactions that may actually take place in a smart space, it nevertheless presents some useful cases, for example, the selection of personalized media and render them in an appropriate player. In real environment, people often find it difficult and spend much time for searching appropriate media and information and hence automatic support in this respect will be appreciated very much.

Table 7.1: Expected relationship between UC-AM variables

Perceived Usefulness	Risk Associated	Privacy Concerns	Perceived Control	Cognitive Attitude	Affective Attitude	Intention to Use
				Positive		More
					Positive	More
More						More
More				Positive		More
	More			Negative		Less
	More					Less
	More				Negative	Less
		More		Negative		Less
		More				Less
		More			Negative	Less
			Less			Less
			Less		Negative	Less

7.1.2 Test Procedure

The scenarios stated above presents some cases of automation that assist users in a smart environment. However, due to the ambiguous context information the automatic system may perform wrong actions or sometimes carry out over-automation or mis-automation, which have an impact on user acceptance and trust towards the system. In order to evaluate such impact, we implemented two separate systems, System-A and System-B, with the above-mentioned scenarios. System-A was designed to be autonomous in realizing those scenarios; users had no explicit control over the system; and the system was supporting a predefined single-level assistance irrespective of the quality of context information. Alternatively, System-B was designed to be adaptive to the changing quality of context and provided dynamic level of control to the users. For example, due to the poor quality of context information in scenario SC-3, System-B would change the interaction level from automated action execution to action suggestion, that is to provide the user with a list of movies instead of executing a particular movie on a particular renderer. This type of dynamic interaction adjustment subsequently involves user to guide the next action to be performed by the system.

In this experiment, in order to measure user acceptance of dynamically adjusted interaction model for intelligent environment, we adopt the Ubiquitous Computing Accep-

tance Model (UC-AM) as proposed by Spiekermann [103]. She conducted an empirical research to find out the impact of control on ubiquitous pro-active system including the Intelligent Speed Adaptation System that adjusts the car speed based on the speed limit, the Intelligent Fridge that refills its shelf by proactively ordering items, and the Automatic car maintenance system that contacts garage for replacement spare parts. It is acknowledged that Ubiquitous Computing is one of the building blocks of ambient and pervasive intelligence [6]. Hence, it will be relevant to use UC-AM to analyze the level of acceptance (that leads to satisfaction) of intelligent systems with the proposed context-driven dynamic interaction model.

Using the UC-AM evaluation model, Spiekermann attempted to capture user acceptance dimension in light of intention to use with several variables of interest. These are perceived usefulness, associated risk, privacy concerns, perceived control, cognitive attitude and affective attitude. The definition used for usefulness is the degree at which the performance of a person may increase by using a system. The cognitive attitude relates to expected performance and affective attitude is usually being associated with system's usability. One of the definitions adapted for risk factor in the UC-AM model is the expectation of losses when purchasing a product or system. Privacy [98] for some people has been an issue with the ubiquitous technologies, as systems collect and use personal data to provide relevant services to the user. Finally, the perceived control is an important issue in ubiquitous and pervasive computing domain [78, 37, 21] that brought many debates as to how much control should people and machine share. The relationship between these factors can be visualized in Table 7.1. As an example, if the perceived usefulness of a system is more, the cognitive attitude towards that system will be positive and the intention to use the system will be more. The rest of the rows in Table 7.1 is self descriptive.

Therefore, as per the UC-AM model, our goal is to investigate whether the proposed dynamic interaction mechanism has positive impact on its acceptance in smart environment settings.

In order to measure trust in the adaptive interaction system we propose, we resort to the approach designed by Jian et al. [64]. In this work the authors proposed an experimental framework to empirically analyze trust between human-human, between

Table 7.2: Demographic distribution of the test participants

Test participants	No	Computer use	Age limit
Female graduate student	6	Yes	17 – 50
Male graduate student	14		
Female undergraduate student	2		
Male undergraduate student	7		

human-machine (automation) and trust in general. Their experiment consisted of word elicitation and questionnaire study coupled with paired comparison study. Based on that experiment, they have identified several factors of trust between users and automated systems, which we adapt to our needs for measuring trust in our proposed interaction model. We explain more about the trust measure policy in subsequent sections.

7.1.3 Data Collection

For our experiment, we invited forty people to participate as volunteer but overall twenty nine people finally committed to show up. Table 7.2 shows their demographic distribution.

We divided the participants into two groups namely GRP-A and GRP-B, having 14 and 15 participants each, respectively. In GRP-A, there were 4 female students and 10 male students. GRP-B consisted of 4 female students and 11 male students. Participants of GRP-A were given a description of System-A while GRP-B received a briefing about System-B to ensure that they know what to expect from the system. Therefore, before the test session actually started, the participants knew the system behavior, that is what kind of actions or inputs would lead into which system output. All the participants were required to come in our simulated home environment where the two prototype systems were deployed. The test was conducted over a period of one month as the participants' schedule was quite different.

Based on their experience during the testing session, each participant were given two questionnaire aiming to measure user acceptance and trust in System-A and System-B. Each test session with one participant lasted about 30 minutes. The first set of questions were related to several variables used in UC-AM model [103], while the second

Table 7.3: Questionnaire used for feedback collection regarding acceptance

No.	Measurement variable	Questionnaire	Measurement scale
1	Perceived usefulness	Using <System-A System-B> would enable me to accomplish other tasks more quickly	5 = strongly agree
		I would find <System-A System-B> useful	4 = agree
2	Perceived ease of use	I found <System-A System-B> easy to use	3 = undecided 2 = disagree 1 = strongly disagree
		Learning to operate <System-A System-B> will be easy	
		It will be easy for me to interact with <System-A System-B>	
3	Perceived control	I think that with <System-A System-B> I can decide any time in which automation mode I want to interact with the environment	
		Through the usage of <System-A System-B> I always know when and which actions have been performed by the system	
4	Perceived risk	I can imagine that if the environment is going to perform some unwanted automations it would do it anyway	

set of questions were related to trust measure [64] as proposed in the field of cognitive ergonomics.

For user acceptance measure, we have chosen a selected set of variables from the UC-AM model, on which we were mainly interested in. These are 1) perceived usefulness, 2) perceived ease of use, 3) perceived control and 4) perceived risk. In [103], although Spiekermann puts less emphasis on the ease of use variable based on the reasoning that actions in ubiquitous computing environment would supposedly require less interaction, we like to argue that even in an automated system users might need to provide their feedback to the system, which eventually requires such interfaces that are easy to use. The variable related to privacy concerns were omitted as we think that its importance varies from people to people and would require further investigation. We also ignore the variables cognitive attitude and affective attitude for which we can eventually get an indication from the other variables that we selected. For example, higher score for perceived usefulness would have positive impact on cognitive attitude towards a system. Again, lower perceived control would have a negative impact on affective attitude as pointed out by Spiekermann. In Table 7.3, we provide the questions for collecting users'

Table 7.4: Questionnaire used for feedback collection regarding trust

Question Group	No.	Questionnaire
Negative implication	1	The system is deceptive
	2	The system behaves in an underhanded manner
	3	I am suspicious of the system's intent, action or output
	4	I am wary of the system
	5	The system's actions will have a harmful or injurious outcome
Positive implication	6	I am confident in the system
	7	The system has integrity
	8	The system is dependable
	9	The system is reliable
	10	I can trust the system

feedback. These questions are adapted from [103] with some adjustment to suit our needs related to dynamic interaction support.

For the trust measure, we took the questions from [64] and adapt it to our experiment. For example, we removed a question about security and another question about familiarity, as they are not much related with the presented scenario. Table 7.4 lists the questions we used. Note that we group these trust-related questions as having negative implication and positive implication. This means a high score to a question in the former group would refer to low trust, while a high score to a question in the latter group would signify increased trust. Also note, unlike [103], the questions in [64] would need to be responded using a scale between 1-7 instead of 1-5.

7.2 Result Analysis

In this section we analyze the results in terms of user acceptance and trust measure in both System-A and System-B.

7.2.1 User Acceptance Measure

We summarize users' response to the different questions related to UC-AM model in order to better understand their perception on both System-A and System-B and qualitatively

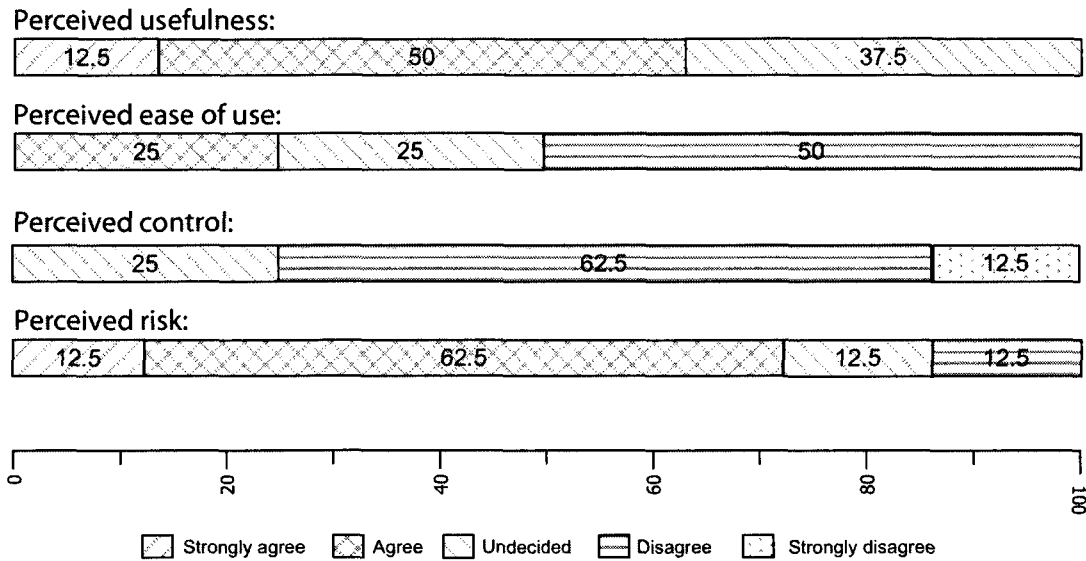


Figure 7.1: User response percentage on the evaluation of automatic interaction mechanism of System-A.

analyze the results. Figure 7.1 shows their responses in percentage scale. We notice that a significant number of users are not happy with the automatic System-A in terms of perceived ease of use, perceived control and perceived risk. However, they mostly agree that this kind of system is useful if designed properly. On the other hand, for System-B with the proposed interaction mechanism, the users were more positive. This is shown in Figure 7.2.

We also observe how the users of different age groups, 1) 30 and below and 2) 30 plus, responded in both the system. These age groups were defined for all the users of GRP-A and GRP-B. Figure 7.3 shows this analogy. It is noticeable that both the age groups responded quite similarly with respect to the different measurement variables. This could be because of the limited number of users and their background. We however observe that all these users provided lower score in terms of perceived ease of use and perceived control on System-A, which also validates our previous observation as in Figure 7.1. For System-B, we summarize the group observation in Figure 7.4. Here we see that both groups have higher perception of usefulness, ease of use and control. We also note that users of 30+ age have little more positive perception on these variables than the

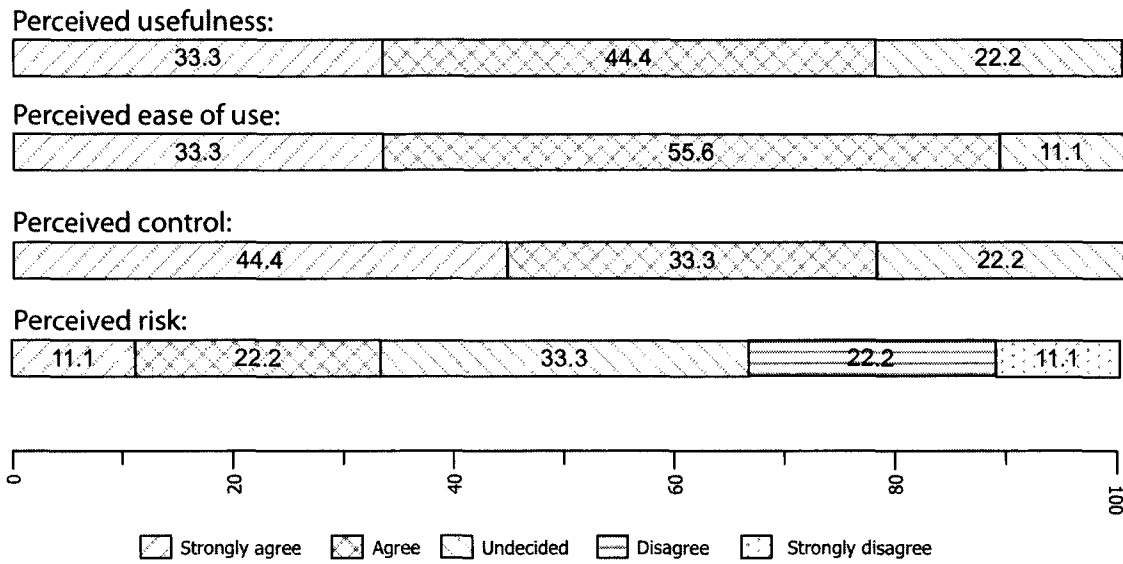


Figure 7.2: User response percentage on the evaluation of adaptive interaction mechanism of System-B.

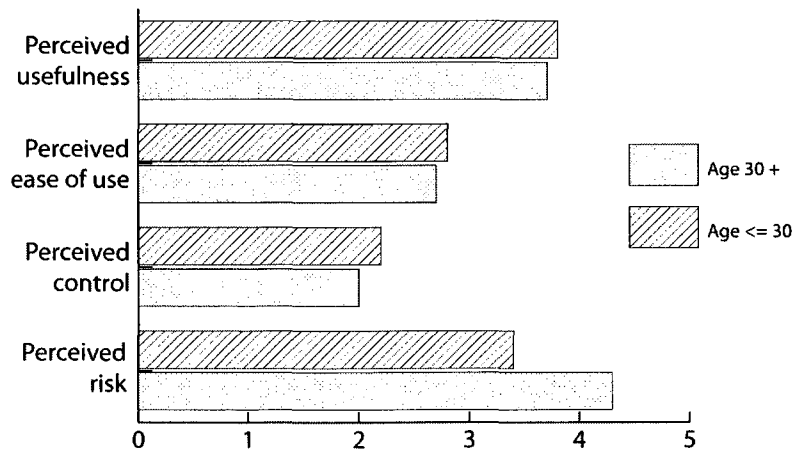


Figure 7.3: Comparison of mean responses between two age groups for System-A in terms of different UC-AM variables.

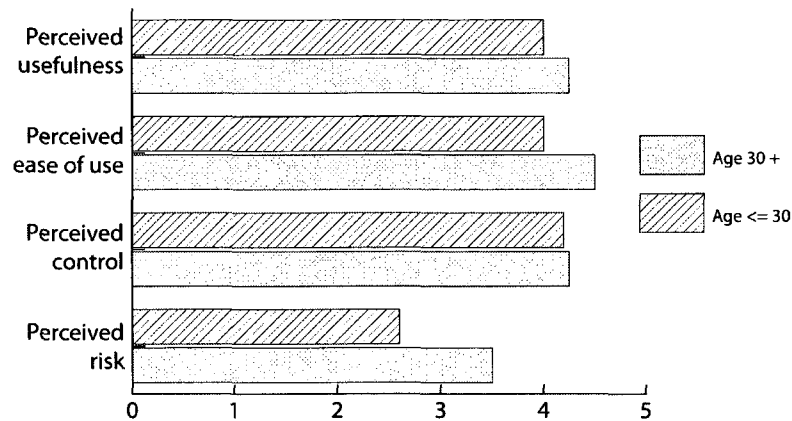


Figure 7.4: Comparison of mean responses between two age groups for System-B in terms of different UC-AM variables.

Table 7.5: User acceptance based on the overall evaluation

Interaction System	Measurement	Perceived Usefulness	Perceived Ease of Use	Perceived Control	Perceived Risk
System-A	Mean	3.75	2.75	2.13	3.75
	Variance	0.44	0.69	0.36	0.68
System-B	Mean	4.11	4.22	4.24	3.02
	Variance	0.54	0.40	0.62	1.33

other group of users.

In order to make a comparison between the two systems with respect to the different measurement variables, we present Figure 7.5 that includes the mean responses of the respective system’s users irrespective of any group. We observe that the system with adaptive interaction mechanism (System-B) have higher perceived usefulness, ease of use and control, which is encouraging and supports the objective of this thesis.

The overall scores provided by the users for both the systems are summarized in Table 7.5. The lower mean value of perceived ease of use and control for System-A reflects poor acceptance by the user. The moderate mean value of perceived usefulness shows that System-A has the potential even if the users provided lower score in other variables. This is because users want some automation that can provide some benefit to them, but they are just worried that the automation performed might be risky and they will have less

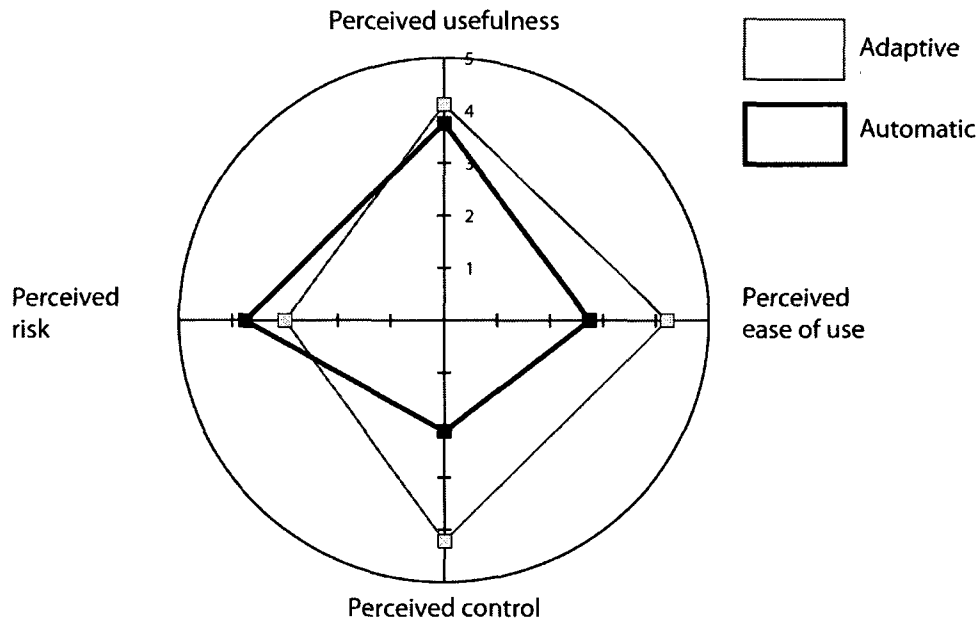


Figure 7.5: Comparison of mean responses on System-A and System-B in terms of UC-AM variables.

control over the system's functionality. We also present a similar analysis in the case of System-B. Here, the higher mean value of perceived usefulness, ease of use and control represents a strong acceptance of the adaptive interaction model. Also, note that users perceive less risk in System-B than in System-A, although both the system is attributed as moderately risky, which can be minimized with improved design.

7.2.2 Trust Measure

The responses provided by the user to the questionnaire in Table 7.4 with respect to the trust measure are summarized in Figure 7.6. In this figure we observe that mean negative feedback on System-A is high while System-B enjoys high mean positive feedback, which validates our assumption on increased trust.

Based on the above analysis we emphasize that quality of context driven dynamically adjustable varying level of interactions has more acceptance than that of fixed level of interaction leading to increased users' satisfaction. Similar observation has been made

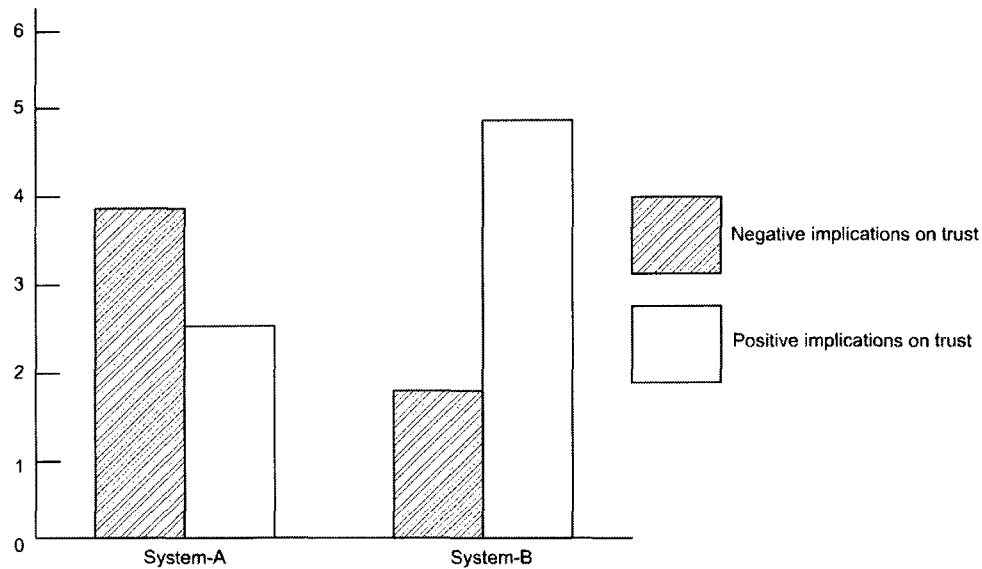


Figure 7.6: Trust measures of System-A and System-B.

in the case of trust in automated systems.

7.3 Summary

This chapter provided experimental results based on the qualitative analysis of the overall framework. In order to measure users' satisfaction we used Spiekermann's ubiquitous computing acceptance model [103], while the trust measure adopted a mechanism proposed by Jian [64]. We found out that users' acceptance of System-B is higher than System-A and at the same time they vested more trust on the proposed System-B. In other words, users' satisfaction is increased and distrust on System-B is minimized when quality of context driven dynamic interaction mechanism is adopted.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this thesis, we proposed an improved interaction mechanism suitable for interacting with pervasive environment. In particular, we addressed the problem of ambiguous context information and its use to facilitate context-aware automated interaction support. For context ambiguity, we defined a multi-dimensional QoI model consisting of certainty, accuracy/confidence and timeliness attributes. A multimodal fusion approach is used to model and assess these attributes dynamically. Based on these attributes, we developed a mapping mechanism to dynamically adjust the different levels of implicit interactions such that users' trust and satisfaction increases towards the intelligent systems. The fact that imprecise context sensing and processing often leads to mis-automation and over-automation causing dis-trust and dis-satisfaction in context-aware automated interaction, the proposed mechanism for dynamic adjustment of the level of interaction minimizes such dis-trust and maximizes users' satisfaction. We also proposed a mechanism to select a set of context-aware media services on the basis of gain-cost trade-off. This service selection mechanism is adopted when the dynamic interaction level is adjusted to service invocation or action suggestion. A framework is designed to incorporate the individual aspects of QoI, the mapping between QoI and dynamic interaction level and the selection of appropriate services in the environment. We also followed the standard SOA-based approach to develop this framework.

We conducted experiments to quantitatively evaluate the different aspects of the framework such as QoI assessment and gain-cost service selection trade-offs. In addition, the proposed framework as a whole was evaluated through qualitative analysis. Our objective of this evaluation was to verify our thesis statement, which is to design QoI-driven adaptive interaction mechanism for increased user satisfaction and trust. For this purpose, we resort to approaches from other research domains. For example, in order to measure users' satisfaction in the proposed interaction approach, we used Spiekermann's Ubiquitous Computing Acceptance Model [103] targeted for automation-related research. The level of trust in the proposed system was measured based on a study [64] in the domain of cognitive ergonomics.

Although the evaluation of the proposed framework was performed in a simulated smart home environment with several developed scenarios, we acknowledge that for real-life deployment it will require real users to be involved in such evaluation over a longer period of time, which is often very challenging and requires further investigation. Nevertheless, the result obtained based on our experiment with volunteers validates our proposal.

8.2 Future Work

Based on the work done in this thesis, several extensions may be proposed:

- **Privacy issues:** Although in our approach we maintained user's updated preference scores based on interactions history, we hardly put our concerns on its implication to privacy of individual users. Therefore, it will be interesting to further investigate this matter and obtain the insights on perceived privacy and its limits accepted by the users in pervasive and ambient intelligence environment.
- **Elderly interactions:** It is worth researching about how the elderly people interact with smart environment and how they accept such technology. Also, as the number of elderly persons are increasing world-wide, how to provide them ambient support in terms of health-care, entertainment, and living is a challenging research issue.

- **Social media recommendation:** In this thesis, we adopted a user-oriented approach to update the preference of a user by his/her interactions history over a period of time. However, it would be interesting to investigate collaborative filtering/social media recommendation approach such that a user's preference can be updated based on the preferences of other similar users.
- **Multimodal interaction:** Depending on the situation of a user in the environment, it is possible to leverage different interaction modalities such as gesture, voice, haptics and mobile phone based interaction. Although the focus of this thesis was to develop mechanisms for adapting the level of implicit interaction based on QoI, we facilitated user to perform explicit interaction with the environment via their mobile phones. However, it is an interesting and challenging research problem to address the use of different interaction modalities simultaneously.

Bibliography

- [1] Amigo. <http://www.hitech-projects.com/euprojects/amigo/>, Accessed on March 21, 2009.
- [2] Verilook SDK. http://www.neurotechnology.com/vl_sdk.html, Accessed on April 06, 2009.
- [3] VLC media player. <http://www.videolan.org/vlc/>, Accessed on January 25, 2009.
- [4] X10 (industry standard). [http://en.wikipedia.org/wiki/X10_\(industry_standard\)](http://en.wikipedia.org/wiki/X10_(industry_standard)), Accessed on January 10, 2009.
- [5] E. Aarts. Ambient intelligence: A multimedia perspective. *IEEE MultiMedia*, 11(1):12–19, Jan 2004.
- [6] E. Aarts and B. de Ruyter. New research perspectives on ambient intelligence. *Journal of Ambient Intelligence and Smart Environments*, 1(1):5–14, 2009.
- [7] G.D. Abowd and E.D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Trans. Comput.-Hum. Interact.*, 7(1):29–58, 2000.
- [8] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.
- [9] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *ACM-SIGMOD Int. Conf. Management of Data*, pages 207–216, May 1993.

- [10] R.L. Aguiar, A. Sarma, D. Bijwaard, L. Marchetti, P. Pacyna, and R. Pascotto. Pervasiveness in a competitive multi-operator environment:the daidalos project. *IEEE Communications Magazine*, 45(10):22–26, October 2007.
- [11] J.E. Allen, C.I. Guinn, and E. Horvtz. Mixed-initiative interaction. *Intelligent Systems and their Applications, IEEE*, 14(5):14–23, Sep/Oct 1999.
- [12] S. Antifakos. *Improving Interaction with Context-Aware Systems*. PhD thesis, Swiss Federal Institute of Technology Zurich, 2005.
- [13] S. Antifakos, N. Kern, B. Schiele, and A. Schwaninger. Towards improving trust in context-aware systems by displaying system confidence. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 9–14, New York, NY, USA, 2005. ACM.
- [14] L. Aroyo, P. Bellekens, M. Bjrkmann, and G. Houben. Semantic-based framework for personalised ambient media. *Multimedia Tools and Applications*, 36(1):71–87, 2008.
- [15] P.K. Atrey and A. El Saddik. Confidence evolution in multimedia systems. *IEEE Transactions on Multimedia*, 10(7):1288–1298, Nov. 2008.
- [16] P.K. Atrey, M.A. Hossain, and A. El Saddik. A method for computing the reputation of multimedia services through selection and composition. In *Fifteenth Annual IS&T/SPIE Conference on Multimedia Computing and Networking*, volume 6818, pages H1–H8, Jan. 2008.
- [17] P.K. Atrey, M. Kankanhalli, and J.B. Oommen. Goal-oriented optimal subset selection of correlated multimedia streams. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1):1–24, 2007.
- [18] P.K. Atrey, M.S. Kankanhalli, and R. Jain. Information assimilation framework for event detection in multimedia surveillance systems. *Multimedia Systems.*, 12(3):239–253, 2006.

- [19] S. Baek, H. Lee, S. Lim, and J. Huh. Managing mechanism for service compatibility and interaction issues in context-aware ubiquitous home. *IEEE Transactions on Consumer Electronics*, 51(2):524–528, May 2005.
- [20] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. Addison Wesley, New York, ACM Press, 1999.
- [21] L. Barkhuus and A. Dey. Is context-aware computing taking control away from the user? three levels of interactivity examined. In *Proceedings of the UbiComp conference, LNCS 2864*, pages 149–156, 2003.
- [22] G. Beccari, S. Caselli, and F. Zanichelli. A technique for adaptive scheduling of soft real-time tasks. *Real-Time Systems*, 30(3):187–215, 2005.
- [23] C. Bisdikian. On sensor sampling and quality of information: A starting point. In *PerCom Workshops*, pages 279–284, White Plains, NY, USA, March 2007.
- [24] F. Blache, N. Chraïet, O. Daroux, F. Evennou, T. Flury, G. Privat, and J. Viboud. Position-based interaction for indoor ambient intelligence environments. In *First European Symposium on Ambient Intelligence*, pages 192–207, 2003.
- [25] E. Blasch and S. Plano. DFIG level 5 (user refinement) issues supporting situational assessment reasoning. In *8th International Conference on Information Fusion*, volume 1, pages xxxv–xliii, Philadelphia, USA, July 2005.
- [26] S. Boutemedjet and D. Ziou. A graphical model for context-aware visual content recommendation. *IEEE Transactions on Multimedia*, 10(1):52–62, 2008.
- [27] P. Brown and G. Jones. Context-aware retrieval: Exploring a new environment for information retrieval and information filtering. *Personal Ubiquitous Computing*, 5(4):253–263, 2001.
- [28] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. Easyliving: Technologies for intelligent environments. In *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing*, pages 12–29, September 2000.

- [29] T. Buchholz, A. Kupper, and M. Schiffers. Quality of context: What it is and why we need it. In *Workshop of the HP OpenView University Association (HPOVUA)*, pages 11–18, Geneva, 2003.
- [30] P. Busetta, T. Kuflik, M. Merzi, and S. Rossi. Service delivery in smart environments by implicit organizations. In *1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 356–363, Boston, MA, USA, August 2004.
- [31] H.S. Carvalho, W.B. Heinzelman, A.L. Murphy, and C.J.N. Coelho. A general data fusion architecture. In *Sixth International Conference on Information Fusion*, volume 2, pages 1465–1472, Queensland, Australia, July 2003.
- [32] D. Chen, J. Yang, R. Malkin, and H.D. Wactlar. Detecting social interactions of the elderly in a nursing home environment. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1):6, 2007.
- [33] F.J. Cook, M. Youngblood, E.O. Heierman III, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. Mavhome: an agent-based smart home. In *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, pages 521–524, March 2003.
- [34] A.K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 5(1):4–7, 2001.
- [35] A.K. Dey, G.D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166, 2001.
- [36] A.K. Dey and J. Mankoff. Designing mediation for context-aware applications. *ACM Trans. Comput.-Hum. Interact.*, 12(1):53–80, 2005.
- [37] A.K. Dey and A. Newberger. Support for context-aware intelligibility and control. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 859–868, New York, NY, USA, 2009. ACM.

- [38] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J-C. Burgelman. Scenarios for ambient intelligence in 2010. IST advisory group final report, 2001. <ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf>.
- [39] S.A. Ehikioya. A characterization of information quality using fuzzy logic. In *18th International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 635–639, New York, USA, July 1999.
- [40] J.L. Encarnao and T. Kirste. Ambient intelligence: Towards smart appliance ensembles. In *From Integrated Publication and Information Systems to Information and Knowledge Environments*, pages 261–270. Springer, 2005.
- [41] M.R. Endsley. *Automation and Human Performance - Theory and Application*, chapter Automation and Situation Awareness, pages 163–181. Lawrence Erlbaum Associates, New Jersey, USA, 1996.
- [42] D. Box et al. Web services eventing (ws-eventing). <http://www.w3.org/Submission/WS-Eventing/>, Accessed on February 15, 2009.
- [43] J. Beatty et al. Web services dynamic discovery (ws-discovery). <http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf>, Accessed on February 10, 2009.
- [44] D.L. Hall and J. Llinas. An introduction to multisensor fusion. In *Proceedings of the IEEE: Special Issues on Data Fusion*, pages 85(1):6–23, January 1997.
- [45] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2000.
- [46] Q. Han and N. Venkatasubramanian. Timeliness-accuracy balanced collection of dynamic context data. *IEEE Transactions on Parallel and Distributed Systems*, 18(2):158–171, Feb. 2007.
- [47] T. Heider and T. Kirste. Smart environments and self-organizing appliance ensembles. In *in Mobile Computing and Ambient Intelligence*. Springer, 2005.

- [48] S. Helal, , W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: A programmable pervasive space. *Computer*, 38(3):50–60, 2005.
- [49] M. Hellenschmidt and T. Kirste. SodaPop: a software infrastructure supporting self-organization in intelligent environments. In *Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on*, pages 479–486, June 2004.
- [50] K. Henriksen and J. Indulska. *Pervasive Computing*, chapter Modeling Context Information in Pervasive Computing Systems, pages 79–117. Springer, 2002.
- [51] K. Henriksen and J. Indulska. Modelling and using imperfect context information. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 33–37, 2004.
- [52] E. Horvitz. Principles of mixed-initiative user interfaces. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 159–166, New York, NY, USA, 1999. ACM.
- [53] E. Horvitz. Uncertainty, action, and interaction: In pursuit of mixed-initiative computing. *IEEE Intelligent Systems*, 14:17–20, 1999.
- [54] M.A. Hossain, P.K. Atrey, and A. El Saddik. Modeling quality of information in multi-sensor surveillance systems. In *IEEE ICDE Workshop on Ambient Intelligence, Media, and Sensing (AIMS)*, pages 11–18, Istanbul, Turkey, April 2007.
- [55] M.A. Hossain, P.K. Atrey, and A. El Saddik. Smart mirror for ambient home environment. In *3rd IET International Conference on Intelligent Environments*, pages 589–596, Ulm, Germany, September 2007.
- [56] M.A. Hossain, P.K. Atrey, and A. El Saddik. Context-aware QoI computation in multi-sensor systems. In *First IEEE Workshop on Quality of Information (QoI) for Sensor Networks (QoISN 2008)*, pages 736–741, Atlanta, Georgia, USA., Sep-Oct 2008.

- [57] M.A. Hossain, P.K. Atrey, and A. El Saddik. Gain-based selection of ambient media services in pervasive environments. *Springer Journal of Mobile Networks and Applications*, 13(6):599–613, Dec. 2008.
- [58] M.A. Hossain, P.K. Atrey, and A. El Saddik. Management of ambient media preferences in distributed environments for service personalization. In *Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks*, volume 0, pages 204–209, May 2008.
- [59] M.A. Hossain, P.K. Atrey, and A. El Saddik. Modeling and assessing quality of information in multi-sensor multimedia monitoring systems. *ACM Transactions of Multimedia Computing, Communications and Applications*, 7(1), 2011. to appear.
- [60] M.A. Hossain, J. Parra, P.K. Atrey, and A. El Saddik. A framework for human-centered provisioning of ambient media services. *Multimedia Tools and Applications*, 44(3):407–431, September 2009.
- [61] M.C. Huebscher and J.A. McCann. Adaptive middleware for context-aware applications in smart-homes. In *MPAC '04: Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, pages 111–116, 2004.
- [62] K. Hughes and N. Ranganathan. A model for determining sensor confidence. In *IEEE International Conference on Robotics and Automation*, volume 2, page 136141, FL, USA, July 1993.
- [63] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.
- [64] J. Jian, A.M. Bisantz, and C.G. Drury. Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, 4(1):53–71, 2005.
- [65] C.D. Kidd, R. Orr, G.D. Abowd, C.G. Atkeson, I.A. Essa, B. MacIntyre, E. Myrnat, T.E. Starner, and W. Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Proceedings of the 2nd international workshop*

on Cooperative Buildings, volume 1670, pages 190–197, 1999. Lecture Notes in Computer Science, Springer-Verlag.

- [66] A. Klein, H. Do, M. Karnstedt, and W. Lehner. Representing data quality for streaming and static data. In *International Workshop on Ambient Intelligence, Media, and Sensing (AIMS)*, pages 3–10, Istanbul, Turkey, April 2007.
- [67] D.E. Knuth. *The art of computer programming. Vol.2: Seminumerical algorithms*. Atmospheric Chemistry & Physics, 1981.
- [68] N. Lazarevic-McManus, J. Renno, and G.A. Jones. Performance evaluation in visual surveillance using the F-measure. In *4th ACM international Workshop on Video Surveillance and Sensor Networks (VSSN)*, pages 45–52, California, USA, October 2006.
- [69] J. Lindenberg, W. Pasman, K. Kranenborg, J. Stegeman, and M.A. Neerincx. Improving service matching and selection in ubiquitous computing environments: a user study. *Personal Ubiquitous Computing*, 11(1):59–68, 2006.
- [70] S.W. Loke, S. Krishnaswamy, and T.T. Naing. Service domains for ambient services: concept and experimentation. *Mobile Networks and Applications*, 10(4):395–404, 2005.
- [71] A. Manzoor, H. Truong, and S. Dustdar. Using quality of context to resolve conflicts in context-aware systems. In K. Rothermel et al., editor, *1st International Workshop on Quality of Context(QuaCon)*, volume LNCS 5786, pages 144–155, Stuttgart, August 2009. Springer-Verlag.
- [72] A. Manzoor, H.L. Truong, and S. Dustdar. On the evaluation of quality of context. In *Proceedings of the EuroSSC*, volume 5279, pages 140–153, 2008. LNCS, Springer, Heidelberg.
- [73] V.Y. Mariano, J. Min, J.-H. Park, , R. Kasturi, D. Mihalcik, H. Li, D.S. Doermann, and T. Drayer. Performance evaluation of object detection algorithms. In *16th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 965–969, Los Alamitos, CA, USA, September 2002.

- [74] S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.
- [75] S. McBurney, M.H. Williams, N.K. Taylor, and E. Papadopoulou. Managing user preferences for personalization in a pervasive service environment. In *Proc. 3rd Advanced Intl. Conf. Telecommun.*, Mauritius, May 2007.
- [76] I. Millard, D.D. Roure, and N. Shadbolt. Contextually aware information delivery in pervasive computing environments. In *Proceedings of the International Conference on Location and Context-Awareness*, pages 189–197, May 2005.
- [77] H. Miller. The multiple dimensions of information quality. *Information Systems Management*, 13(2):79–82, 1996.
- [78] J.M.V. Misker, J. Lindenberg, and M.A. Neerincx. Users want simple control over device selection. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence*, pages 129–134, New York, NY, USA, 2005.
- [79] S. Muller-Schneiders, T. Jager, H.S. Loos, and W. Niem. Performance evaluation of a real time video surveillance systems. In *The Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 137–143, Beijing, China, October 2005.
- [80] E.F. Nakamura, A.A.F. Loureiro, and A.C. Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, 39(3):9, 2007.
- [81] J.C. Nascimento and J.S. Marques. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia*, 8(4):761–774, August 2006.
- [82] A.A. Nazari Shirehjini. A novel interaction metaphor for personal environment control: direct manipulation of physical environment based on 3d visualization. *Computers & Graphics*, 28:667–675, 2004.

- [83] A.A. Nazari Shirehjini. *Interaktion in Ambient Intelligence – Konzeption eines intuitiven Assistenten zur ganzheitlichen und konfliktfreien Interaktion in adaptiven Umgebungen*. PhD thesis, Technischen Universitt Darmstadt, 2008.
- [84] R. Parasuraman, T.B. Sheridan, and C.D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 30:286–297, May 2000.
- [85] W. Pasman. Organizing ad hoc agents for human-agent service matching. *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 278–287, 2004.
- [86] L. Peng and K.S. Candan. Confidence-driven early object elimination in quality-aware sensor workflows. In *Proceedings of the 2nd international workshop on Data management for sensor networks (DMSN '05)*, pages 45–51, New York, NY, USA, 2005. ACM.
- [87] L. Peng and K.S. Candan. Predictive early object shedding in media processing workflows. In *IEEE International Conference on Multimedia and Expo (ICME '07)*, pages 1191–1194, July 2007.
- [88] W. Qin, D. Zhang, Y. Shi, and K. Du. Combining user profiles and situation contexts for spontaneous service provision in smart assistive environments. In *5th International Conference on Ubiquitous Intelligence and Computing*, pages 187–200, Oslo, Norway, June 2008.
- [89] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294–307, March 2005.
- [90] A. Ranganathan, , J. Al-Muhtadi, and R.H. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *Pervasive Computing, IEEE*, 3(2):62–70, April-June 2004.
- [91] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

- [92] S. Sastry and S.S. Iyengar. Real-time sensor-actuator networks. *International Journal of Distributed Sensor Networks*, 1(1):17–34, 2005.
- [93] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Mobile Computing Systems and Applications Workshop*, pages 85–90, Santa Cruz, CA, USA, 1994.
- [94] T. Schlogl, C. Beleznai, M. Winter, and H. Bischof. Performance evaluation metrics for motion detection and tracking. In *17th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 519–522, California, USA, August 2004.
- [95] A. Schmidt. Implicit human computer interaction through context. *Personal and Ubiquitous Computing*, 4(2):191–199, 2000.
- [96] A. Schmidt, M. Beigl, and H.-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999.
- [97] S.A.N. Shafer, B. Brumitt, and J.J. Cadiz. Interaction issues in context-aware intelligent environments. *Hum.-Comput. Interact.*, 16(2):363–378, 2001.
- [98] K. Sheikh, M. Wegdam, and M. van Sinderen. Quality-of-context and its use for protecting privacy in context aware systems. *Journal of Software*, 3(3):83–93, March 2008.
- [99] T.B. Sheridan. *Humans and Automation: System Design and Research Issues*. Santa Monica, John Wiley & Sons, 2002.
- [100] C. Shin, A.K. Dey, and W. Woo. Mixed-initiative conflict resolution for context-aware applications. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 262–271, New York, NY, USA, 2008. ACM.
- [101] M. Siegel and H. Wu. Confidence fusion. In *IEEE International Workshop on Robot Sensing*, pages 96–99, 2004.
- [102] L. Snidaro, R. Niu, G.L. Foresti, and P.K. Varshney. Quality-based fusion of multiple video sensors for video surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(4):1044–1051, 2007.

- [103] S. Spiekermann. User control in ubiquitous computing: Design alternatives and user acceptance. Shaker Verlag, November 2008.
- [104] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 252258, Ft. Collins, CO, USA, 1999.
- [105] J. Tyan and Q.H. Mahmoud. A comprehensive service discovery solution for mobile ad hoc networks. *Mobile Networks and Applications*, 10(4):423–434, 2005.
- [106] N. Villar, G. Kortuem, K.V. Laerhoven, and A. Schmidt. The pendle: A wearable mediator for mixed initiative environment. In *The IEE International Workshop on Intelligent Environments (IE 05)*, pages 173–181, University of Essex, Colchester, UK, 2005.
- [107] J. Wang, M.S. Kankanhalli, W.-Q. Yan, and R. Jain. Performance evaluation of a real time video surveillance systems. In *ACM Workshop on Video Surveillance(IWVS)*, Berkeley, CA, USA, November 2003.
- [108] R.Y. Wang and D.M. Strong. Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–34, 1996.
- [109] B.P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, Aug. 1962.
- [110] K. Xu, M. Zhu, D. Zhang, and T. Gu. Context-aware content filtering & presentation for pervasive & mobile information systems. In *Proceedings of the 1st international conference on Ambient media and systems*, pages 1–8, Quebec, Canada, 2008.
- [111] D.J. Yates, E.M. Nahum, J.F. Kurose, and P. Shenoy. Data quality and query cost in pervasive sensing systems. *Pervasive and Mobile Computing*, 4(6):851–870, 2008.

- [112] Z. Yu, X. Zhou, D. Zhang, C. Chin, X. Wang, and J. Men. Supporting context-aware media recommendations for smart phones. *IEEE Pervasive Computing*, 5(3):68–75, 2006.
- [113] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, 29:371–388(18), December 2000.
- [114] F. Zhu, M.W. Mutka, and L.M. Ni. Service discovery in pervasive computing environments. *IEEE Pervasive Computing*, 4(4):81–90, 2005.
- [115] M. Zhu, D. Zhang, J. Zhang, and B.Y. Lim. Context-aware informative display. In *IEEE International Conference on Multimedia and Expo*, pages 324–327, July 2007.
- [116] F. Ziliani, S. Velastin, F. Porikli, L. Marcenaro, T. Kelliher, A. Cavallaro, and P. Bruneaut. Performance evaluation of event detection solutions: the creds experience. In *IEEE Conference on Advanced Video and Signal-Based Surveillance*, pages 201–206, Como, Italy, September 2005.
- [117] T. Zimmer. QoC: quality of context improving the performance of context-aware applications. In *Advances in Pervasive Computing*, pages 209–214, Dublin, Ireland, 2006.