FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

# Model-based Automatic IO Device Selection for Ambient Environments

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieurin**

im Rahmen des Studiums

**Medizinische Informatik**

eingereicht von

**Brigitte Kupka**
Matrikelnummer 0525298

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker
Mitwirkung: Dr.techn. Christopher Mayer

Wien, 18.01.2016

(Unterschrift Verfasserin) (Unterschrift Betreuung)

# Model-based Automatic IO Device Selection for Ambient Environments

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieurin**

in

**Medical Informatics**

by

**Brigitte Kupka**
Registration Number 0525298

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker
Assistance: Dr.techn. Christopher Mayer

Vienna, 18.01.2016

(Signature of Author) (Signature of Advisor)

# Erklärung zur Verfassung der Arbeit

Brigitte Kupka
Weizenweg 43a, 1220 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

| | |
|---|---|
| (Ort, Datum) | (Unterschrift Verfasserin) |

# Abstract

Active and Assisted Living (AAL) aims to provide older adults and persons with special needs with customized services, to ease their lives at home and raise their quality of life. For this purpose, sensor technology, pervasive and mobile computing and artificial intelligence are combined, to analyze the current situation surrounding a system and respond in the form of adaptive software.
The project AALuis - Ambient Assisted Living user interfaces deals with the task of providing different health and comfort services to users, mainly older adults in their homes, with a focus on accessible user interfaces. Interaction with these services is possible using different devices and facilitating a variety of modalities for input and output. The possibilities depend on the devices available to the user and the modalities those devices support. Goal of this thesis is the definition and implementation of a model-based selection system of the best device and modality for user interaction with an AAL service, based on various factors of influence, including a user profile, the available devices, and sensor information, like ambient noise. Due to the complexity and variability of such a system, a model-based approach is more efficient than the development of a highly specialized algorithm. Based on the dynamic nature of the process of User Interface (UI) generation in AALuis and the ongoing development of new in- and output devices, this model has to be easily adaptable to new devices and, subsequently, previously not accounted for modalities. For the selection mechanism, a Java software component, using the model for inference, has to be developed, as a preliminary step for the integration in AALuis.
This thesis first introduces the background of the technologies involved in AAL and the state of the art of sensor technology and adaptive software. Probabilistic graphical models, specifically Bayesian Networks, were chosen as a modeling technique for the solution of the selection problem. It is explained in detail before the process of developing the model is described. Subsequently, the structure and functionality of the created software components are given and the results of the model-based selection mechanism are presented and discussed.
Context information regarding the user, the available devices and their properties, as well as data acquired from sensors constitute the modeled input parameters of the selection mechanism. The output is a combination of device and modality, which is best suited given the current situation to convey the output to the user. Using scenarios, it could be shown that the results, given the described situations are plausible and respond in a comprehensible way to changes in context. The run-time of the selection mechanism, even for a large example, including a large number of possible devices, modalities and sensors, averages at 59.934 ms, and is therefore in a range that does not entail a disruptive delay in the execution of services. It can therefore be concluded that

the proposed solution is able to perform the Model-based Automatic IO Device and Modality Selection for AALuis in a satisfactory way.

# Kurzfassung

Active and Assisted Living (AAL) versucht älteren Menschen und Personen mit speziellen Bedürfnissen maßgeschneiderte Dienste anzubieten, die ihnen das Leben in ihrem Zuhause erleichtern und zu einer höheren Lebensqualität beitragen. Dazu werden Technologien aus den Bereichen Sensorik, verteilte und mobile Computersysteme und künstlicher Intelligenz miteinander kombiniert, um die aktuelle Situation einer Umgebung zu analysieren und mittels adaptiver Software darauf einzugehen.

Das Projekt AALuis - Ambient Assisted Living user interfaces bietet seinen Benutzern, vorrangig älteren Menschen die zu Hause leben, verschiedene elektronische Dienste im Gesundheitsbereich mit barrierefreien User Interfaces an. In AALuis ist es möglich, jedes User Interface für eine Interaktion mit den Services in verschiedenen Modalitäten zu erzeugen, abhängig davon, welche Geräte dem Benutzer gerade zur Verfügung stehen und welche Möglichkeiten von ihnen unterstützt werden. Dafür werden die User Interfaces zuerst abstrakt definiert und erst dann konkret umgesetzt, wenn eine zu verwendende Modalität dafür festgelegt wurde.

Das Ziel dieser Arbeit ist die Definition und Umsetzung eines modellbasierten Selektionsmechanismus für die Benutzerinteraktion mit AAL Services über AALuis. Für die Auswahl sollen zahlreiche Entscheidungskriterien, wie ein Benutzerprofil, die Verfügbarkeit von Geräten und Sensordaten, wie zum Beispiel Umgebungslärm, herangezogen werden. Aufgrund der Komplexität und Variabilität eines solchen Systems, ist ein modellbasierter Ansatz effizienter als die Entwicklung eines hoch spezialisierten Algorithmus.

Nachdem AALuis die User Interfaces unabhängig vom verwendeten Gerät für viele verschiedene Modalitäten generieren kann, soll es möglich sein, das System einfach um neue Geräte und Modalitäten zu erweitern, um auch technische Neuentwicklungen oder spezielle Bedürfnisse der Benutzer zu unterstützen. Der Selektionmechanismus wird als Java Programm, das Inferenz auf dem entwickelten Modell durchführt, umgesetzt, um die Integration in AALuis vorzubereiten.

Diese Arbeit geht zunächst auf die Grundlagen des Bereichs AAL ein und beleuchtet die vielen technologischen Facetten, die einen Einfluss darauf haben. Probabilistische graphische Modelle, genauer gesagt Bayes'sche Netzwerke, wurden zur Modellierung des Selektionsproblems gewählt. Ihre Funktionsweise wird detailliert erklärt und das entwickelte Modell beschrieben. Dadurch wird der Aufbau und Funktionsumfang der entwickelten Software erläutert und die Ergebnisse des modellbasierten Selektionsmechanismus präsentiert und diskutiert.

Als Eingabegröße für den Selektionsmechanismus dient die modellierte Kontextinformation betreffend den Benutzer, die vorhandenen Geräte und deren Eigenschaften, sowie Sensordaten die die Umgebung beschreiben. Ausgegeben wird diejenige Kombination aus Gerät und Modalität, die in der aktuellen Situation am besten geeignet ist die Ausgabeinformation zu transportieren.

In Szenarien wurde gezeigt, dass die Ergebnisse in den beschriebenen Situationen sowohl plausibel sind, als auch in einer nachvollziehbaren Art und Weise auf Veränderungen des Kontext reagieren. Die Laufzeit des Selektionsmechanismus liegt auch in einem großen Beispielszenario, bestehend aus vielen möglichen Geräten, Modalitäten und Sensoren, mit im Durchschnitt 59,934 ms in einem Bereich, der keine störende Verzögerung der Ausgabe der Services nach sich zieht. Dies lässt den Schluss zu, dass das entwickelte Modell geeignet ist die Selektion von Gerät und Modalität in AALuis zufriedenstellend zu erfüllen.

# Contents

# CHAPTER 1
# Introduction

## 1.1 Motivation

Information and Communication Technology (ICT) stands out by its diverse field of applications. It is indispensable in industry, transportation, entertainment and health care. This important position goes hand in hand with an undeniable influence on society. One of the biggest social challenges today is the inclusion of elderly and disabled people. It is important to not only provide adequate care for them, but to enable them to participate in their community and in society in general. In the past 10 years, the share of the population aged 65 years or older has increased in every EU member state, while the part of the population younger than 15 years has decreased[1]. As people get older, their skills slowly degenerate, making every day tasks more difficult. Chronic diseases are common amongst older adults and often require extensive therapy and monitoring.

Many people are in need of varying degrees of care. Some require personal attention by professional caretakers, either at home or in a facility. But a lot of older adults live at home and only require assistance in some areas of their lives. It stands to reason that the current efforts in home automation, the advances in mobile computing, sensor technology and artificial intelligence can be put to practical use in AAL.

AAL Environments use modern technology to provide services specifically tailored to elderly or disabled people and their caretakers. Those services cover emergency prevention and detection, comfort services, telecommunications, telecare and in general have the objective of promoting and maintaining physical and psychological health, and assistance in daily life. In this way, ICT can contribute to healthcare and the care of the elderly itself, as well as facilitate the accessibility of computer systems.

This paradigm entails new challenges for the designers of hard- and software, as new types of UIs are needed to make the most out of Ambient Environments. First the inclusion of many

[1]eurostat - Population structure and ageing, data extracted in June 2015 `http://ec.europa.eu/eurostat/statistics-explained/index.php/Population_structure_and_ageing`

different devices, some of them mobile, raises the question of which device to use to initiate communication with the user. A mobile device might not always be present and available to the AAL service. Also not all devices have the same capabilities, so a certain degree of flexibility is required. Secondly, the users in AAL have special needs. Traditional forms of Human Computer Interaction (HCI) may not offer full accessibility for them. Also, they cannot be expected to learn overly complicated systems, when the goal of AAL is to provide useful services to them. Context-aware user interfaces have been proposed, using the information provided by sensors and other sources to automatically adapt the software in a suitable way. For individual AAL services, the automatic generation of UIs has been proposed, either tailoring the contents or presentation of the UI to the current context or investigating the distribution of UIs to the active devices around the Ambient Environment. To make AAL systems fit for real-world applications, there is a need for frameworks that support the integration of services from different vendors, along with the consolidated use of the infrastructure that at the same time guarantees accessibility of the UIs for the users.

## 1.2 Problem Statement

AALuis (Ambient Assisted Living User Interfaces) aims at providing a framework for AAL services that relieves the developers of those services from the task of creating UIs suitable for the distributed and volatile environment found in AAL. The framework is able to flexibly generate UIs from abstract UI descriptions supporting different modalities and devices. A modality is the channel used to transport a message between a human and a computer. Text or audio are examples for output modalities, text or speech would be corresponding input modalities. For the scope of this work, modalities comprising of multiple channels, like video combining images and audio, will also be identified as one modality.
The context of an AAL system includes the user, the user's capabilities and constraints, the user's personal preferences, the available devices, and the situational context like surrounding noise, light, temperature, and ongoing activities. It is reasonable to assume that in different contexts some modalities and devices might be preferable over others. For example, text output on a small device is not appropriate for a person with visual limitations. Audio output on a device that is currently in another room as the user might not be heard.
The model-based Automatic IO Device and Modality Selection has the objective of selecting the best device and modality combination given a certain context in a step preceding the generation of a concrete UI in the AALuis framework. The context information to be considered in the selection process includes data from sensors, the AALuis user profile and the available devices and their properties. The AALuis system supports the integration of new entities effortlessly, which means that the devices available and therefore the available modalities are subject to changes. The Automatic IO Device and Modality Selection has to be able to support these changes without the need for additional programming at a later time.

## 1.3 Goals

The goal of this thesis was to find a modeling technique that is able to represent the relevant context for the AALuis Automatic IO Device and Modality Selection and support the decision making process to solve this selection problem. Beyond the selection itself, the focus was placed on the extensibility and easy maintenance of the system.
Furthermore, a software should be developed, to integrate the model-based selection mechanism into the existing AALuis framework.
The selection mechanism has to be able to cope with the uncertainty in Ambient Environments, which implies that it is not guaranteed that all context information will be available at the time of execution. Using a rule-based system was excluded from the choice of possible solutions, because rules become cumbersome to maintain in larger domains.

## 1.4 Method

The first step to solve the goals stated above was to gather information on existing devices and the supported modalities used in AAL and other areas of technology in order to create a basis of exemplary devices and modalities and to gain an understanding of the problem domain. This was accomplished by the review of literature and commercially available products. The next step was to find a method suitable to support the required decision making process using a literature review and the consultation of researchers experienced in the field. Bayesian Networks proved to be the most promising approach to meet the requirements. Utilizing SamIam, a software supporting the modeling of Bayesian Networks, a model covering the problem space of the output device and modality selection was developed in several iterations. Particular consideration had to be given to the requirement of easy expandability of the model for new devices and modalities while minimizing the learning curve to train administrators to carry out such expansions. After the model performed satisfactory in tests of different prototypical scenarios, it was integrated into a stand-alone Java application, the AALuis Output Evaluator, as a preliminary step to present the feasibility of integrating the modeled Bayesian Network into the AALuis system. A second part of the application, the AALuis Output Editor, was programmed to facilitate the adaptation of the model with a Graphical User Interface (GUI).
To validate the results, the plausibility of the output generated by the selection mechanism was verified in three different exemplary scenarios.
The AALuis Output Device and Modality Selection has to be able to produce a result without creating a noticeable delay in execution for the user. The time of computation depends on the size and structure of the Bayesian Network. To evaluate how the runtime scales with the network size, it was analyzed for three different networks.
A minimal example, containing only one device, one modality and two pieces of context information, 8 nodes in total. A medium sized network containing 52 nodes in total, and a large network, four times the size of the default network.
Analyzing the runtime of probability queries, two cases were distinguished. The first query after startup takes significantly longer, because it includes building the join tree for the inference

algorithm. For all subsequent runs this step can be skipped which is reflected in the shorter average runtime. Therefore, for all examples, the execution time of the first probability query after startup was measured 20 times. Then, the execution time of queries after the join tree was already built, was evaluated for 20 runs of each example network.

## 1.5 Outline

Chapter 2 describes the background and history of AAL and the relevant issues in creating UIs for Ambient Environments. The results of the literature research regarding the state of the art in context-aware computing, sensor technology and machine learning algorithms are given in Chapter 3. Chapter 4 describes the selected method, Bayes Networks, in depth to provide the reader with the knowledge necessary to understand the modeling process, which is detailed in Chapter 5. Chapter 6 explains the software implementation. In Chapter 7, the output of the selection mechanism in three different scenarios is shown, accompanied by the results of the run-time analysis. Chapter 8 ends this thesis with a discussion of the results, a summary of the presented work, and an outlook on further research questions.

CHAPTER 2

# Background

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”

Mark Weiser

## 2.1 Active and Assisted Living

The goal of this thesis was the development of one component of an Active and Assisted Living (AAL) framework. AAL, formerly referred to as Ambient Assisted Living, is a concept that involves a variety of disciplines. This chapter starts with a look at the research that influenced the concept of AAL, beginning with the vision of Ubiquitous Computing and follows up with some background exploring the setting of this work with Active and Assisted Living User Interfaces.

### Ubiquitous Computing

When Personal Computers slowly began to make their way into private households in the 1980s, they were large and expensive machines. Up to then, their interfaces were text-based and users had to learn their language, their commands, to use them. The development of GUIs drastically improved usability, while at the same time computers became more powerful, smaller and more affordable. Today, many people own not only a personal computer, but a myriad of mobile devices including laptops, smart phones and tablets. Thanks to the advancement in mobile technology, computers have become almost *ubiquitous*. Most of us use a number of computers throughout the day and the interaction is not restricted to a certain place, like the desk, anymore. The term Ubiquitous Computing was first coined by Mark Weiser at Xerox PARC in 1991 [83]. He and his fellow researchers imagined a future that included hundreds of interconnected computers in every home and office. They built prototypes of tiny Post-It note sized machines called „tabs“, book sized „pads“ and large „boards“ acting like blackboards. People in their office were outfitted with machines called active badges, small wearable devices that enabled computers to

adopt behavior programmed specifically for the current user's preferences or location inside the office building. Weiser and his team imagined these machines to be integrated seamlessly into every-day tasks, so that users would not even be aware of the fact that they were using computers. They envisioned interaction beyond traditional input devices, using so called natural user interfaces [28], for example pointing a tab at a board, to send content from that tab to the board. In this vision of Ubiquitous Computing, not the devices themselves would be of significance, but their seamless interaction with each other and especially with the user and the environment. Instead of the user having to adjust to the way computers are operated, their operation would have to be highly intuitive.
Since 1991, not only has the range of available and affordable devices broadened, today's networking technology meets the necessary requirements effortlessly. Home automation technology is promoting the possibility to remotely control and monitor electronics, like heating systems or air conditioning over the internet, requiring only the installation of necessary sensors and actuators. But despite the advancements in the number and variety of networked devices, the idea of computers fading into the background and unnoticeably attending us in our lives is still far from reality. Computers are indispensable for numerous tasks, but in most cases we still have to actively approach them and enter the right commands, navigating their proprietary interfaces with keyboard, mouse and touch screens. In order to recede into the background, to interact more naturally with us, they need to anticipate what a human expects of them in a certain situation and adapt their behavior accordingly and make use of innovative modes of interaction.

### Ambient Intelligence

Ambient Intelligence was introduced in 1999 by the Information Society Technology Advisory Group (ISTAG). It combines the technological progress of Ubiquitous Computing with research in sensor technology, wireless network technology, Artificial Intelligence and the field HCI in an integrated perspective. But while Ubiquitous Computing focused very much on the technological possibilities, Ambient Intelligence emphasizes on the use of this technology by the individual, in their environment, and the possible benefits to society [46]. The concept of Ubiquitous Computing laid the foundation for this technological paradigm by focusing on the development of interconnected devices that offer adaptive behavior based on location information. But why limit the capabilities of our systems to programs reacting to the position of a device and its user, when we can employ a variety of sensors and combine their data with Artificial Intelligence techniques to infer appropriate adaptive behavior that goes far beyond location-awareness? Augusto and McCullagh [8] define Ambient Intelligence as follows:

> „A digital environment that proactively, but sensibly, supports people in their daily lives.“

Ambient Intelligence emphasizes the aspect of assisting the user and approaches the topic with a broad spectrum of disciplines regarding the interrelation of humans and computers as well as the technological artifacts that are placed in the human world for this matter. Figure 2.1 shows the fields of computer science that Ambient Intelligence is based on. One of the main challenges of Ambient Intelligence technology lies in the psychosocial aspect. When Augusto claims Ambient

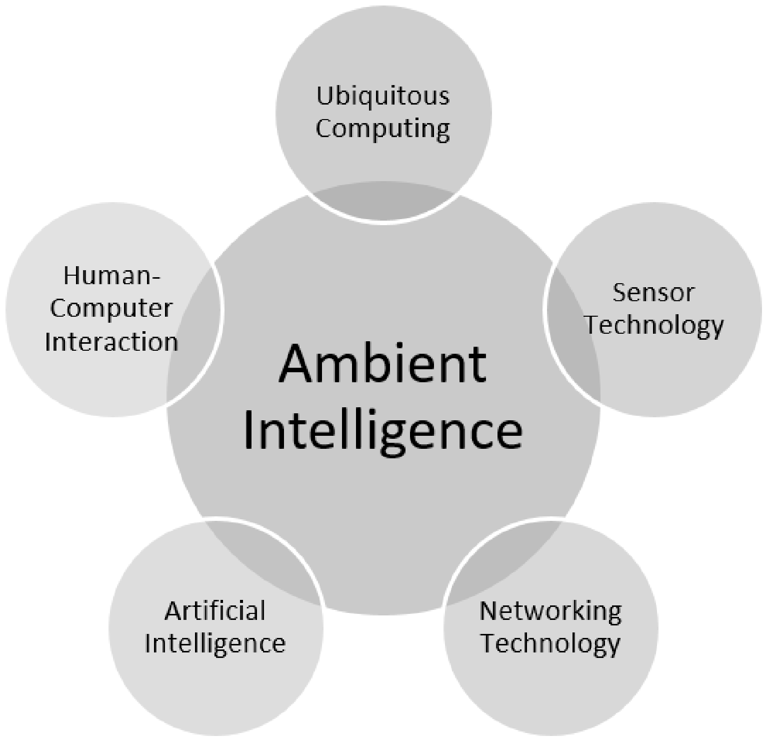


Figure 2.1: Relationship of Ambient Intelligence and other fields of study.
Reprinted from Augusto and McCullagh [8]

Intelligence should *sensibly* support people, it is clear that this cannot be achieved by an isolated effort in technological advancement.
Ambient Intelligence is not one technology or standard but a general concept for future computer systems. Several environments could be augmented by Ambient Intelligence technology. It is conceivable to enhance homes, offices, transportation or public areas with networked sensors and devices, and to use artificial intelligence to anticipate the user's needs, to offer added value and streamline their experience interacting with the *smart* environment. The smart home shall serve as an example to discuss the features of Ambient Intelligence in more detail.

**Smart homes** Smart homes are houses enhanced by Ambient Intelligence technology, which offers services tailored to the needs of the residents in their daily life. Of all the possible environments for Ambient Intelligence, the home is of especially high importance. It is a place to relax, to feel safe and secure. Many elemental activities take place inside the home. Aside from the physical structure, a home can be viewed as a collection of functional spaces, where we satisfy our basic needs [37] like regenerating, preparing food and eating, personal hygiene and family life. How well these needs can be fulfilled, has a high influence on the overall quality of life.
The automation of every day tasks, thereby achieving higher comfort, is one of the goals of many

smart home research projects. In order to turn a regular home into a smart home, it is outfitted with Ambient Intelligence technology. It complements regular consumer electronics present in most homes with sensors and actuators, networking infrastructure and a home gateway at the center. The introduced smart components can be stand-alone, like for example motion detectors installed in a room, or actuators to open and close blinds. They could also be integrated into conventional objects. One example could be power outlets which are able to detect whether a device is connected and provide the possibility to control the power supply for this device remotely.
Wearables, devices incorporated into clothing, or worn as accessory, can be used to provide a smart home with location information on multiple residents or measure vital parameters, like the heart rate [7]. Some smart homes even include robots, representing an interface for the user [87] or service robots [71].
There is a large number of projects researching which tasks can be performed efficiently by smart homes. The comfort functions mentioned above are primarily concerned with recognizing certain activities using sensor data and taking suitable automated actions on the user's behalf. Other comfort functions are essentially remote control features for processes which in the past could not be controlled remotely [1]. The preservation of resources and especially reducing energy consumption can effectively be provided by these controls [37]. Switching lights on and off, depending on the location of the residents and the tasks they currently perform, seems to be a basic component of most smart home projects [58] [43] [20]. It represents a comfort feature and has the potential to reduce energy consumption as well. It depends on sensor data to locate the users and, using Artificial Intelligence, may try to infer what activities they currently carry out and what kind of light intensity the users prefer given the presumed activity. There are numerous and manifold ideas and implementations in the areas of cooking and dining, household, maintenance, recreation, social interaction and family life which cannot all be listed here.
Notable projects include the Gator Tech Smart House of Florida University [43], including a smart closet, that suggests outfits according to the weather forecast or the smart front door, which allows keyless entry for authorized persons. The smart mailbox notifies the residents of incoming mail. The degree of usefulness of some proposals may certainly be questioned, but the experience gained from even the most flamboyant projects is still valuable.

### Active and Assisted Living

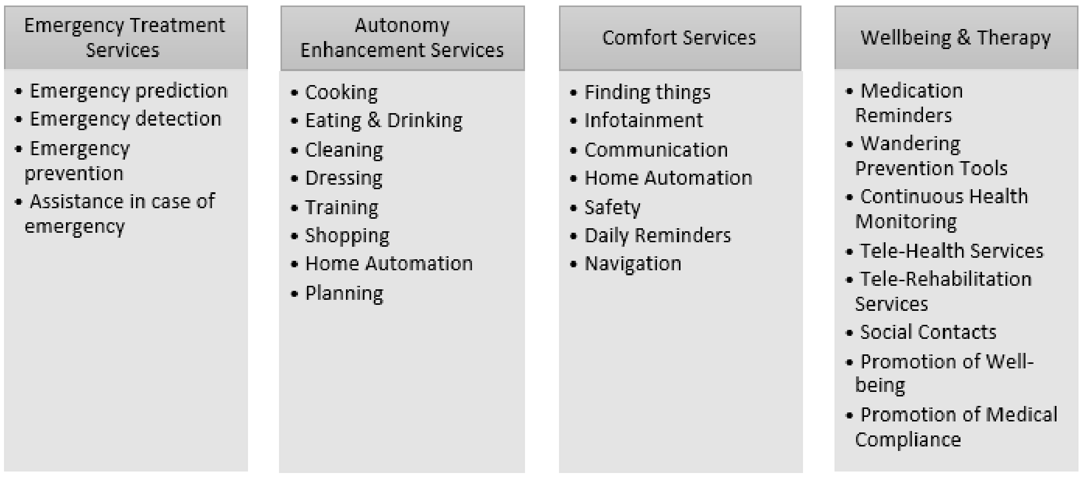
AAL employs Ambient Intelligence technology to aid elderly or disabled people living outside of institutional care in various aspects of their life. The number of elderly people in Europe's population is high and is projected to grow in the future. Living independently at home is attractive, not only for the individuals in question, but also for society in general. To an elderly person, living at home means autonomy and empowerment. A study conducted in New Zealand in 2011 has shown that when asked, pensioners placed a high importance on being able to choose their form of residency themselves [84]. To many of them, their home community is part of their identity and remaining in that place represents autonomy and independence. Institutionalized care is perceived as an undesired loss of autonomy.
For society, the cost of care on the one hand justifies an interest to maintain the health and well-being of elderly citizens. On the other hand, it remains to be seen what form of care can even

be provided in the future if the Age-Dependency-Ratio continues to rise. The Age-Dependency-Ratio is the rate of people 65 years old or older, to the number of people in working age (15-64 years), the potential care takers. In the European Union (27 countries) the ratio developed from 20,6 per 100 persons of working age in 1990, to 28,2 per 100 persons of working age in 2014 according to Eurostat[1]. Considering the cost of care mentioned before, it has to be noted that AAL technology, supposing it will mature to the consumer market, also pose a significant investment [30], which has to be balanced against the cost of institutionalized care.
How can older adults profit from AAL technology? Essentially, elderly and disabled people can benefit from smart home technology described above just as every other user. However they constitute a group of users with special characteristics, compared to the younger, tech-savvy users usually targeted by companies marketing modern technology. In the older population there is a higher prevalence of illnesses and health problems in general. Using AAL, smart homes can be tailored to support the health of its residents.
Figure 2.2 shows the so called Living Assistance Domain, grouped into four core areas, giving an indication about the span of possible AAL services. The first is emergency treatment. Since a large share of elderly people live alone, they are heavily affected by accidents and emergencies. If no other person is present to recognize the situation and intervene, even minor incidents can have serious consequences. It would be even better to prevent emergencies altogether as often as possible. AAL services can contribute much to this goal. By means of sensors they can monitor the activity inside the home and detect unusual patterns in the inhabitant's behavior early using data mining methods, as described by Jakkula et al. [47]. Fire or gas detectors are already common, but by augmenting household appliances with sensors, it could be possible to warn the residents of a number of safety hazards in advance, such as floods caused by water tabs left open [75]. Special medical sensors, some are even available as wearable devices, can be used to monitor certain vital signs, like heart rate or blood oxygen saturation [63]. The continuous collection of data also presents us with entirely new options for the creation of original parameters for measuring health status. Arcelus et al. for example measure the sit-to-stand transfer duration, the duration it takes for a person to stand up from bed [6]. If this duration suddenly gets longer, it could point to difficulties regarding the person's overall mobility, possibly long before the person would report the issue to a caretaker or doctor.
Another relevant factor for many older adults is medication. Most oft he elderly population require one or several drugs regularly. Confusions concerning medication can have fatal effects. Also, the regular intake of the medication can impact on its effectiveness. AAL systems can remind patients to take the right medication at the right time. The same reminders can be used to prompt users to perform enough physical activity. Keeping doctor‘s appointments and cultivating social contacts cannot only be facilitated by reminders. Higher accessibility can be provided by AAL systems which support telecommunications services, making those tasks easier. These examples demonstrate that comfort services from smart homes could serve the purpose of autonomy enhancements, wellbeing or therapy services when adapted to the needs of older adults, provided that the users are willing to adopt them. Heart and Kalderon found in their study [42] that while older adults do use ICT, they tend to only adopt new systems if they expect a sig-

[1] `http://ec.europa.eu/eurostat/tgm/table.do?tab=table&init=1&plugin=1&language=en&pcode=tsdde510` Accessed November 2015

nificant advantage compared to solutions they are familiar with. Especially people with health problems are found to be less inclined to use modern technology. The adaptiveness of AAL systems could alleviate this effect [14] by making them easier to use and emphasizing their key features.
A number of so called AAL services have been mentioned above to give the reader an impression of possible applications, although the term *service* remains yet to be defined. It describes the hardware, software and human actors or organizations, which interact to offer a certain benefit to the user. Most examples given above use one or more sensors and possibly actuators, multimedia and telecommunication devices on the hardware side to measure data, perform actions or contact participants. The software processes the measured data and controls the actions. There are, however, also a number of services, which by nature require the participation of persons or organizations. Care organizations may offer telecare services linked to AAL hard- and software. For example, a service monitoring the activity inside the home, could trigger an alarm if the user leaves the bedroom unusually late. A caretaker at a connected organization could receive an alarm and place a phone or video call to assess the situation. The human contact creates a personal relationship, benefiting the user. Moreover, the caretaker can perceive a more complete picture of the situation than the sensors in the home could. In this manner, technology and human resources can work together very efficiently. Many elderly people live alone, but receive regular home care. These caretakers also could be participants in an AAL system, configuring the services, maintaining data or using it to confer with other health care professionals or involved relatives.
Many such individual AAL services have been proposed in the past, but a standard on how to implement these in real homes has not yet been devised. To use multiple services efficiently at the same time, these have to have access to the devices. Ideally, they would also utilize a common component for processing sensor data. Middleware is needed to unite the smart hardware components with the AAL service software, operating on a smart home gateway. The EU project universAAL [2] for example, aims at developing such an open platform to facilitate a growing range of AAL services by ensuring interoperability. Developers, service providers, organizations and users benefit from this approach, offering systems which can be easily enriched by new services, using the existing hardware components jointly. Such a platform is also fundamental to ensure certain important non-functional requirements of AAL systems. These systems are of a highly personal nature for the user. It is for this reason that they have to meet high expectations regarding Quality of Service. Nehmer et al. [62] highlight the following nine aspects. *Availability* is of high priority in a service that supports users in health related matters. Functionality which is not available at the right time can cause serious harm when users rely on it. *Timeliness* addresses that same aspect. Many of the proposed services are time critical and have to respond in a timely manner. Emergency services are obviously of particular interest in that regard. *Interoperability* is not only important for the propagation of AAL technology in general, but also plays a relevant part in raising the quality of service. The time sensitive nature as well as the complex situation that arises from combining many different hardware and software components makes it clear that *resource efficiency* is an important factor. *Extensibility* is also noteworthy, if an AAL system is to remain sensible over a longer period of time. It needs to

[2]http://www.universaal.org/



| Emergency Treatment Services | Autonomy Enhancement Services | Comfort Services | Wellbeing & Therapy |
|---|---|---|---|
| • Emergency prediction | • Cooking | • Finding things | • Medication Reminders |
| • Emergency detection | • Eating & Drinking | • Infotainment | • Wandering Prevention Tools |
| • Emergency prevention | • Cleaning | • Communication | • Continuous Health Monitoring |
| • Assistance in case of emergency | • Dressing | • Home Automation | • Tele-Health Services |
| | • Training | • Safety | • Tele-Rehabilitation Services |
| | • Shopping | • Daily Reminders | • Social Contacts |
| | • Home Automation | • Navigation | • Promotion of Well-being |
| | • Planning | | • Promotion of Medical Compliance |

Figure 2.2: The Living Assistance Domain adapted from Rashidi and Mihailidis [70], Nehmer et al. [62] and Kleinberger et al. [51]

be able to adapt to altered life circumstances or to changing medical conditions. The examples for applications described above have shown that many of the proposed services have to perform monitoring of activities inside the home at all times. The collected data is highly personal, which is why *security* cannot be neglected to ensure that the resident's privacy is protected. Quality of service in the area of *safety* demands that software has to be specified in detail and validated, to avoid exceptions or hardware defects to cause unintended behavior. *Robustness* describes the attribute that wrong user input must not cause the system to crash. The last two aspects that Nehmer et al. define are the need for a high quality in *adaptivity* and the call for *natural, anticipatory Human-Computer Interaction*. The last two topics are issues of particular importance for this work.

## 2.2 AALuis - Ambient Assisted Living User Interfaces

The focus of AAL is on the human being. In contrast to simple home automation, its benefits unfold only through the interrelation between the home equipped with AAL technology and the person living in it. The interface between the user and the AAL system is a decisive factor for its perceived usefulness. Apart from any technological accomplishments, the UI makes the difference between a system perceived as cumbersome and intrusive or as a useful improvement. The idea of the project AALuis is to provide innovative UIs for open AAL systems. It relieves service developers of the design of the UI by assuming the task of linking the AAL system with the devices. The AALuis middleware operates from between another AAL middleware platform and the devices.
AAL services that wish to initiate a user interaction, contact the AALuis middleware and provide it with an abstract description of the User Interface in a specific format based on the Concurrent Task Tree notation [64]. An abstract UI is not specific to a device or modality and describes

one interaction included in the AAL service. In an intermediate step AALuis transforms this description to a document in the MariaXML format [65]. AALuis automatically selects a suitable combination of device and modality and creates an abstract UI description based on context and device information. Next, a concrete UI is created and sent to the selected IO device for final rendering. Input and output of one interaction do not have to be carried out on the same device, but can be distributed over devices connected to the AAL system. AALuis is based on OSGi to facilitate the integration by existing AAL middlewares and the joint use of resources like context information.

## 2.3 Human-Computer Interaction in Ambient Environments

Human-Computer Interaction was identified to be equally important to the Active and Assisted Living paradigm as the other, purely technological research fields. Given that this work addresses the creation of UIs, a few issues of this field related to AAL will be listed here.
It is a broad multidisciplinary field of computer science and psychology that deals with the design of the interaction between humans and computers. Small computers which integrate themselves seamlessly into homes not only pose challenges to the development of smaller devices and networking technology, it is also obvious that interactions working well with a stationary computer, do not necessarily do so in those new circumstances. A person using a desktop PC, a smartphone or laptop approaches the device intentionally and dedicates a certain amount of attention to it to complete a task. In Ambient Environments users cannot apply this same amount of attention to the computers. It would be stressful and defeat the purpose of Ubiquitous Computing, which started out demanding that the computers should fade into the background. For that reason Dix claims that interactions in Ambient Environments have to be kept to a minimum [28].
In AAL, the physiological and cognitive constraints of elderly or disabled users must be taken into account. Thus accessibility and usability are of particular importance. These challenges can be met with technological advancement on one hand and the study of the psychological factors of HCI on the other hand. Interaction has to happen everywhere in the Ambient Environment, which entails that it can rely less on traditional input devices and instead depend more on input that is more natural to the user, like speech recognition or gestures. The output can be made less intrusive by incorporating it into objects, in sometimes creative and innovative ways. In- and output devices can be integrated into regular household objects. Corsten et al. propose to use object tracking to convert objects to input devices by exploiting their physical properties and translate a manipulation of the user on that object to input. Two examples mentioned include using a regular pen as a pointer on a screen, including the mapping of clicking the pen to a command, and rotating a mug on the desk to simulate turning a knob to regulate light intensity or volume of a device [21]. Other approaches introduce entirely new objects to serve as interface, like the Nabaztag [3] a rabbit shaped device that uses speech synthesis for output, recognizes objects via Radio-frequency Identification (RFID) tags, and provides information unintrusively using colored LEDs. Matching the output modality to the urgency or importance of the content

[3]`http://www.nabaztag.com/`

is an important tool to prevent the Ambient Intelligence Environment from interrupting the users too frequently. Combining established and new modalities and devices to convey information while taking into account the individual constraints of especially the elderly users of AAL is a challenge for technology designers but can improve accessibility.
The design of interfaces is also decisive for the user's acceptance of systems. It is obvious that faulty or difficult to use systems are less likely to be accepted or perceived as a benefit. Emotional factors are also very important. Distributing sensors in a living environment is a delicate matter and users do not wish to be reminded of the fact that their lives are being recorded by a computer system [90]. In case an AAL service assumes a task that previously entailed contact with a human service provider, Burzagli et al. propose that the interaction should be designed to be as „human-like" as possible to improve acceptance [15]. Paro, a social robot shaped like a baby seal and designed for therapeutic purposes, has produced remarkable reactions in residents of care homes, including lowered stress levels [56]. The findings suggest that the potential influence of technological artifacts on the emotions of users is not to be underestimated. The consequences of bad design in AAL can be severe if users rely on the systems regarding their health. A user-centered design approach can help developers of AAL systems to identify areas of concern that they might not recognize on their own. It might not be enough to empathize with the users, but the individuals concerned, the potential users, can usually rely on years of experience that should be utilized in the design process.

## 2.4 Modeling

The terms model and model-based are used frequently throughout this thesis. This section will give a brief introduction on the process of modeling. [13].
A model is a representation of a process or a system. To build a model based on the observation of a real system, the modeling process combines the abstraction, idealization, simplification and aggregation of these observations. The result is a model that has a distinct purpose. It can be used to process the specific problem that was the reason to create it. It returns a result, that can in turn be interpreted and applied to the real system it represents.
The challenge of the modeling process is to create a model that is as simple and concise as possible, but includes everything that is relevant to the problem.
Models can serve as tools for the design of systems, help gain new insights or be used for simulation. Modeling is advantageous when it is not feasible to conduct experiments to gain insight. Also, when calculations are complex, a model, due to its simplicity, can be more efficient.

CHAPTER 3

# State of the Art

Adaptive user interfaces facilitate context information, which is acquired using sensors. Therefore, this chapter starts with a section on current sensor technology, and then proceeds introducing context and context models. Following the state of the art of adaptive user interfaces, it concludes with a section on machine learning and decision making for context-aware software.

## 3.1 Sensors

A sensor is a piece of hardware that transforms a physical property into an electrical signal, which can be processed as a discrete value by a computer. Sensor technology aims at reflecting reality as accurately as possible, but the value a sensor delivers is influenced by its sensitivity, dynamic range, its accuracy, resolution and possible noise [85]. A certain physical property can often be measured making use of different physical or chemical effects, determining the sensors characteristics. A sensor has to be chosen to fit the demands of the intended area of use.

Ambient Environments strongly rely on sensor technology to collect context information. The sensors can be installed around the home, worn or carried by the user, integrated into household objects working passively by collecting data, or integrated into dedicated devices to be used actively. An important distinction is made between invasive sensors, which have to be worn or used by the user consciously and non-invasive sensors operating from a distance. Especially health related sensors often require physical contact with the person, for example when measuring body temperature or blood oxygen saturation. Wearable devices, integrated into clothing or even into the fabric are being developed to achieve a subtle and unintrusive way to measure health data [70].

Many sensors used in smart homes aim at collecting data about the environment and have to be distributed in every area relevant for the proposed services. In most cases this is done after the home has been built, which makes wireless sensors more convenient to install. Such sensors are often integrated into Wireless Sensor Networks (WSNs). A WSN contains a number of dispensed sensor nodes, which can communicate with each other and a corresponding base unit.

Two different architecture types can be distinguished. In a flat architecture, sensor nodes use each other to route their data to the base unit. In a clustered architecture, specific relay nodes collect the data from their designated sensor nodes and forward it to the base station [80].
There are different standards for network communication in WSNs, defining the possible data rate, energy consumption, and the range the sensor networks can cover. IEEE 802.15.4. was developed with the aim of low cost and low energy consumption. It defines the communication on the lowest two layers of the ISO OSI model [92], the physical layer and the data link layer. These layers include the frequency used and the modulation technique. It serves as a basis for other protocols designing the details of communication on the higher levels of the model.
ZigBee of the ZigBee Alliance[1] is one example. In literature, ZigBee based Wireless Sensor Networks are often used in AAL projects, especially for fall detection [17] [27] [88], but also for monitoring of health data like Electrocardiogram (EKG) [57] or blood oxygen saturation (spO2) [82]. The main advantage of ZigBee sensor networks are the low battery consumption, low cost, and the fact that one ZigBee sensor network can support up to 65.536 nodes.
Bluetooth is a wireless standard that uses a short range frequency band. It is used in many conventional devices that are intended to pair with a laptop or smartphone, both of which commonly support Bluetooth connections. Older versions of the standard were not well suited for continuous operation in wireless applications because the energy consumption was too high. Introduction of version 4.0, Bluetooth Low Energy, rectified this situation, introducting this standard to small wearable devices. [89]. Bluetooth cannot be dismissed in Ambient Environments, because if offers easy integration of many products available off the shelf. Commercial healthcare products supporting Bluetooth include weight scales, pedometers, heart rate monitors, thermometers, blood pressure and blood glucose meters. An overview of those devices was compiled by Miguel-Bilbao et al. [23]. Recently, Bluetooth connected devices in the form of smart watches have also found their way into AAL. These devices can combine sensors like gyroscopes and optical heart rate monitors with a small display, collecting data and serving as a User Interface for the Ambient Environment at the same time [11]. Bluetooth employs a Master-Slave topology, where up to seven slaves are connected to one master and cannot communicate with each other. However, the range of Bluetooth LE is limited to only 10m [36]. While a single ZigBee node can cover a distance of around 100m, the mesh or flat topology, enhances the range significantly by routing the data of distant nodes over multiple hops to the base station [70], [35].
UltraWideBand technology aims at providing wireless communications with high data throughput at a short range of up to 10 meters. The approach can solve the problem, that the increasing number of wireless devices leads to a lack in available frequencies, by sharing frequencies of the radio spectrum [69], but cannot match the low prices of ZigBee sensors [45].
Zubiete et al. performed a review on the use of Wireless Sensor Networks in health applications. They found that many proposed AAL services used proprietary approaches for the representation and transmission of sensor data. The use of a standard for the exchange of data would improve the interoperability of sensor networks and existing and newly developed services [93].
Table 3.1 gives an overview of which parameters can be measured by sensors and their applications in AAL environments. Table 3.2 lists common medical devices using sensors which can be integrated into Ambient Environments to monitor vital signs. The most basic devices used for

[1] `http://www.zigbee.org`

| Measurement | Applications |
|---|---|
| Pressure | Pressure mats and smart tiles for location detection |
| Temperature | Body temperature, room temperature |
| Weight | Body weight |
| Light | Light intensity |
| Sound | Ambient noise level, activity recognition |
| Infrared | Motion detection |
| Ultrasonic | Location tracking, motion detection |
| Acceleration | Fall detection |
| $CO_2$ | Respiration |
| Flow | Water usage |
| RFID | Object identification |
| Magnetic Switch | Doors opening / closing |
| Video | Activity recognition |

Table 3.1: Sensor measurements and examples of their applications in Active and Assisted Living, adapated from Alam et al. [1] and Rashidi and Mihailidis [70]

| Physiological device | Measurement |
|---|---|
| Electrocardiogram | Electrical activity of the heart |
| Spirometer | Pulmonary Function Tests |
| Galvanic Skin Response | Sweating, Emotional response |
| Pulse oximeter | Pulse oxygen level |
| Sphygmomanometer | Blood pressure |
| Pulse meter | Heart rate |
| Glucometer | Blood glucose level |

Table 3.2: Physiological devices implementing sensor technology to measure vital parameters. Adapted from Alam et al. [1].

sensing in Smart Homes are binary switches, sensing magnetic contact. They can be installed on doors, cupboards and other objects and provide indirect information about the ongoing activities [26]. Similarly, pressure mats can report the presence of a heavy object or person at a specific location. On the opposing end of the spectrum of complexity are video cameras used as sensor technology for AAL. Videos offer a high density of information and could, sophisticated image processing techniques provided, be a rich source for context information, especially in combination with other sensors. The storage requirements and computing effort put this advantage into perspective [33] [26]. The high information density makes the collection of video footage potentially valuable for deriving context. At the same time this information density represents the biggest drawback of videos in AAL, because it raises serious privacy issues for the residents of a smart home as well as their employees and guests [90]. Finally, RFID must be included as a technology for sensing context information in Ambient Environments. RFID tags are small and can be attached to almost any object to associate it with a unique identifier. Combined with

RFID readers those tags can be used very efficiently for the identification of objects and thereby deriving their proximity [66].

## 3.2 Context

Ambient Environments rely on sensor technology to provide the input used to enable the software to intelligently assist the user. To that end the data provided by sensors, has to be turned into context information. The term context has been used in this text so far without clarification. The first attempts in context-aware computing focused mainly on identifying users and their location [81] [41]. Today, the term context is commonly defined much broader. Dey defines context in his work „Understanding and using context“ [24] as follows:

> “Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.”

What is notable about this definition is that it does not offer any enumeration of what is meant by *information*. Also, by including every entity that *is relevant*, it shows that what exactly constitutes as the context is dependent on the view of the user and of the person designing the application in question. In general terms, context is obviously a collection of descriptive information. In Ambient Environments, context information can include details about the user, the location, the devices present, the activities currently performed or planned, the time and the physical properties of the environment [40].
Henricksen et al. [44] distinguish between static and dynamic context information. Static context does not change or changes happen only rarely. It can be provided in files by a system administrator, the user or, considering AAL systems, by a caretaker. This often includes user profiles, the identity of the home's residents or the address. In dynamic context however, time is a defining factor. Dynamic context information changes and must therefore be updated regularly or it loses its validity. Thus, it can only be collected reasonably from sensors or other automated sources.
Chen [19] distinguishes three different levels of access for context-aware software acquiring sensor data. The first is *Direct Sensor Access*, where an application directly accesses the sensors on the hardware level and processes the acquired data according to its requirements. In Ambient Environments however, there is typically not one solitary application that can demand exclusive access to the sensors. Access to the sensors has to be managed between the services. *Middleware infrastructures* take over the process of accessing the sensors and afterwards store the inferred context information, thus enabling multiple applications to simultaneously access it. Lastly Chen mentions *Context Servers* to host the context information in distributed environments. By using such a server it is possible to remove the acquisition, storing and processing of context from the devices running the context aware applications. In Ambient Intelligence Environments, it is reasonable to remove the processing of context information from any individual device to the home gateway. This does not only solve possible conflicts when accessing the hardware, but also ensures that the sensor information can be reused by multiple services and

the system can be extended [9] [72].
From the above definition it is evident that there is a difference between sensor measurements and context information. Baldauf et al. [9] describe the process of deriving context information from sensor data and making it available to context-aware applications in the form of a conceptual framework, consisting of five layers. Encapsulating the retrieval of raw context data in the *sensor* layer makes the logic in the higher layers independent of the actual sensing hardware. The *preprocessing* layer is where sensor measurements are transformed into context information. The state of a binary sensor can be mapped easily to a piece of context information, each state of the sensor representing a clearly distinguishable state in the real world. One common example would be a door, which can be open or closed. For more complicated situations, the measurements of sensors have to be processed, aggregated or combined with each other. To determine if one out of a number of predefined contextual situations is taking place, automatic reasoning on the basis of sensor data can be used. Amoretti et al. [3] propose using a camera to infer the posture of a human, classifying postures into standing, sitting and lying. By combining this context information derived from cameras with other context sources, activities of the person can be estimated using machine learning classification algorithms. A person standing in the kitchen, combined with the fact that the oven is turned on can together be classified as the activity „cooking“ [3]. The processing in this layer has to be able to cope with missing data and with all the errors that may origin from the sensors, like noise or malfunction. The *storage and management* layer stores the preprocessed data and offers it to the applications via an interface. To store and retrieve the context information efficiently, different context modeling techniques have been proposed.

### Context Models

The simplest way to store context information is key-value pairs [73] where the keys describing certain context attributes are assigned with a value. The larger the model, the harder it becomes to store context information this way, because the keys have to be unique and descriptive. It is also not possible to capture the relationship between attributes, or ensure that there are no inconsistencies among them [10]. The use of unstandardized keys prohibits the easy integration of the model for new services. Markup scheme based models improved the key-value approach by arranging the entries in hierarchical fashion and allow the definition of data types and ranges of valid values. In their survey of context modeling, Strang and Linnhoff-Popien [78] also identify graphical models, object-oriented models, logic based models and ontology based models. They find that object or ontology based context models are the most sophisticated solutions for ubiquitous environments, because of their support of validating the model and annotating the context information with details about the quality and thus the reliability of this information. They reflect positively on the ability to handle ambiguous and incomplete data and bringing the data up to a high level of formality, making it easier to use the context model for existing and new software environments. Ontologies define a vocabulary to represent knowledge, thereby promoting a shared understanding of a certain domain [40].

## 3.3 Adaptive User Interfaces

When context has been sensed, processed and stored, context-aware software can use it to adapt its behavior or state during run-time. The goal of adaptive software is an improvement to the interaction for the user. Manually configuring settings is the traditional approach to support adaptiveness in computer software. But the more options there are for configuration, the more cumbersome this process gets and users are likely to give up on adjusting the settings. Moreover, Langley also notes that it is conceivable that there are options that are implicit, so that users can not consciously decide on their preference [54]. Context-aware software can relieve the user of this task by performing self-adaption. Since this work is concerned with the adaption process in a framework that supports the generation of UIs using different devices and modalities, this section will concentrate on adaptive UIs.
Evers et al. [32] identify four types of context-aware software, based on the degree of adaption. In software that includes parametric adaption, attributes are identified which are adjusted according to certain parameters of context information. One example is the screen of a mobile phone adjusting its brightness to the ambient light surrounding the device. Compositional adaption occurs in software that is structured in independent modules. The modules can be organized in different setups or combinations, offering functionality specific to the context. Deployment adaption is a term found in distributed environments where modules can be substituted by elements of other entities of the domain. Last, there is adaption by service integration, where whole services are joined together in context-aware configurations. According to this categorization, the Automatic IO Device and Modality Selection for AALuis is an example of parametric adaption, with a high number of input parameters, the context, and two closely related output parameters, namely the best suited device and modality combination for the interaction in the current context.
Adaptive UIs change the appearance of the user interaction related to the context. To achieve that, the system must first recognize a situation that triggers an adaption. Then, the appropriate adaption to that new situation has to be computed and executed [16]. One way to obtain the necessary context information is for a service to query actively for changes in relevant context. Another approach is to implement a framework which offers functionality for services to subscribe to certain pieces of context information. These services then receive updates in case there was a change in relevant context automatically [25]. Both approaches are appropriate for the AALuis IO Device and Modality Selection, especially if we assume that an ongoing interaction should not be interrupted by a switch to another device or modality in case relevant context changes occur. When updated context information is either queried by or pushed to an application, it has to decide if it constitutes a situation that triggers an adaption, using an adaption engine [79].

If the adaption engine comes to the conclusion that an adaption should occur, the new user interface has to be computed. To realize adaptive user interfaces, interactions are decomposed into the user interface and a formal description of the underlying task. Concurrent Task Tree is a notation that breaks a task into subtasks and describes their relationship graphically in a hierarchical order [64]. Temporal operators are used to denote the temporal relationship between tasks, distinguishing between interleaving tasks, tasks that need to synchronize, one task enabling another task with or without information passing, one task deactivating another task, iterative tasks, finite tasks, optional tasks and recursive tasks. There is also a distinction between tasks concerning their performance, the user tasks, performed entirely by the user, the application task, performed entirely by the application, the interaction task, describing interactions initiated by the user, and abstract tasks, describing more complex processes not matching the other three categories. Using such a notation to describe a task helps to ensure that the concrete user interface does not result in a loss of functionality necessary for the completion of the task. In AALuis, services which wish to use the framework provide descriptions of their tasks in the Concurrent Task Tree notation. In addition, the actual content for output is provided by the service. On the basis of this information, an abstract user interface description is created, that is still not specific to a device or a modality. Only after the device and modality have been selected, the concrete user interfaces are produced and rendered [59].

Blumendorf et al. [12] developed a framework for the creation of multimodal user interfaces supporting distribution of those interfaces on various devices. User interfaces are generated based on UI models at runtime. The communication is either sent to all devices simultaneously or the user actively picks the preferred resources. An algorithm then tries to determine which other tasks are sufficiently similar to the one the user decided on and uses that same channel for their user interfaces. They use templates to describe user interfaces for every known output channel, limiting their system's ability to automatically include new devices, which was a prerequisite for the AAluis Automatic IO Device and Modality Selection.

Möller et al. [60] created a development framework (Mobile MultiModal Interaction - M3I) for the creation of multimodal applications with the aim of speeding up the development process by offering a convenient way to include context information to influence the input and output modalities of mobile applications. The M3I framework utilizes a rule-based mechanism for switching between modalities. The influencing factors and the rules operating on those factors can be implemented at the time of development, although the possibility is shown that a graphical user interface for the creation of rules is offered to the end users themselves. The rule based approach was dismissed for the Automatic IO Device and Modality Selection in AALuis because it restricts the flexibility of the selection process and requires frequent maintenance.

In Ambient Intelligence, adaptive User Interfaces are not only studied to better integrate the user interaction into the pervasive environment, but also because they can contribute significantly to making the services accessible for elderly or disabled persons. For people with severe health limitations, adaptive user interfaces can be essential to being able to use AAL services at all. While there are specialized input and output devices to support people with special needs and cognitive disabilities, which can and should also be integrated into Ambient Environments, these are typically expensive because they only cater to a small share of the market. Carbonell anticipates positive interactions between efforts to smoothly integrate computers into Ambient Environments and specifically experimenting with innovative modalities and attempts to enhance accessibility for elderly and disabled users [18]. Utilizing the context-aware approach to enhance accessibility for the users of AAL technology offers not only static adaption, but provides continuous adaption to the changes in their abilities and preferences over time [77].

Systems providing adaptive User Interfaces have to be able to handle the uncertainty inherent to environments of context-aware software. The possible malfunctions of sensors have already been mentioned but Esfahani and Malek studied the factor of uncertainty in self-adaptive software in more detail. They found, amongst other factors, that there are uncertainties which have to be addressed in every step of self-adaptive software, including ill-fitting models, unforeseen occurrences in the environment which cannot be reflected accordingly by the context model, model drift, and the user acting on his own accordance. Moreover, analyzing and modeling what the desired outcome of an adaption process is from the point of view of the user, is a difficult task [31].

## 3.4 Machine Learning and Decision Making in Context-Aware Software

Machine learning is used when programming an algorithm manually is not possible or not practicable. Instead, the desired algorithm is extracted by a machine learning technique from example data by detecting distinctive patterns in it. When the training phase is completed, the algorithm should be applicable to new datasets. Machine learning techniques are used in recommender systems, search engines or natural language processing [29].
The existing methods can be differentiated between supervised and unsupervised learning. In supervised learning, training data is processed beforehand to serve as an example for the desired result. The learning process produces an algorithm that best maps the input data to the output data. If the training data was a representative sample, it is expected that the algorithm performs equally well during live operation with new samples. In unsupervised learning, the output is not predefined and the algorithm carves out characteristic details about the samples autonomously. The results are classification or association rules or classifiers which can be used for prediction, by using them for inference on new data. Some classifiers can also be used for descriptive purposes, because they represent knowledge about the data [2].
The complex data involved in context-aware software offers many application cases for such techniques. Wood et al. use unsupervised machine learning to predict future context, given the currently detected context in mobile environments, to save energy and computation time. At first, they cluster sensor data to discern different context situations. Then, they use a Markov chain encoding the probability of one context occurring after another one to predict the probable next context [86]. Faridi and Rahman used Hidden Markov Models to predict the next probable context derived from activity and location information obtained from sensors [34]. To use Hidden Markov Models, training data in the form of sequences of states is required. The volatile nature of an AALuis system results in a high number of possible states, for example, not all known devices are necessarily available at all times. The creation of training data would be too complex to use this technique, even without the need to include new devices or modalities at a later time.
Andreu and Angelov use a fuzzy rule-based system to derive knowledge about performed activity from wearable sensors [5]. Using this approach requires rules, mapping the input space to output classes. These rules can be derived from training data or be formulated by an expert. Aside from the problems regarding training data for the Automatic IO Device and Modality Selection in AALuis mentioned before, defining rules based on fuzzy logic, cannot guarantee that the set of rules covers the complete input space. In such a scenario, input could lead to no clas-

sification and the decision would have to fall back on defaults, which is not satisfactory. Also overfitting can occur if there are too many or too narrow rules or the person defining the rules does not have a complete understanding of the rule base as a whole.
Kabir et al. [49] developed an application for a smart home that provides context-aware services. They use an ontology to reduce raw sensor data collected by various sensors in the smart home to high level context information. Specific configurations of context that should trigger a change in the software's behavior are identified and mapped to the desired action. This data serves as training data for a k-nearest neighbors classifier, using Mahalonobis distance as metric. With new data, this kNN classifier is used to select a service appropriate to the sensed context, which the user can overrule by manually selecting different services. These manual corrections facilitate a reinforcement learning feature that adapts the behavior learned from the training data. The described approach, using a k-nearest neighbors classifier [2], is not suitable for the AALuis Automatic IO Device Selection, because it uses a metric to classify samples according to some of their features. Adding devices, modalities, sensors or user characteristics would entail repeating the feature selection process and may result in significant changes in the model so that it cannot be guaranteed that the metric will produce satisfying results.
Song and Cho [76] developed a context-adaptive user interface for the purpose of simplifying control of various home entertainment systems. They modeled the home environment as a Bayesian network to determine the devices that are likely being used in a certain situation and a behavior network to find the necessary functions that the user interface would have to include to control those devices. The user interfaces are generated from templates according to the result. The Bayesian Network used to model the device selection is a probabilistic graphical model. This method employs graph theory and probability theory to model a problem space for inference. Using a graph for representation is advantageous for human observers, because it has descriptive power but it is also a data structure that can be used efficiently by algorithms. By assigning probabilities to the possible outcomes of classification during inference it is possible to rank the results. It also entails that a result can always be obtained, even if the given context fits poorly into any class. Bayes Networks are also able to deal with missing input data, using the assigned probabilities in such cases instead. New observations of input data can be added at all times, resulting in an update of the result that can be computed efficiently. It does not rely on rules but on modeling the conditional independence of the modeled concepts. While the graph structure does not guarantee the easy extension of the model when new entities are added, interventions are only necessary at certain points. The model can be created automatically on the basis of sample data, but it can also be built by an expert in the domain who has an understanding of the modeling technique without any samples. For those reason, it was chosen as a technique to realize the Automatic IO Device and Modality Selection for AALuis.

CHAPTER 4

# Bayes Networks

## 4.1 Probabilistic Graphical Modeling

A Bayes Network is a probabilistic graphical model, more specifically it is comprised of a Directed Acyclic Graph (DAG) that can encode a large joint probability distribution in a compact fashion. An example for such a graph is shown in Figure 4.1. It allows to infer posterior probabilities of the occurrence of events when certain prior knowledge about other events is available. Bayes Networks were first introduced by Judea Pearl in 1988 [68] and are named after the English mathematician Thomas Bayes. A DAG $G = (V, E)$ consists of a finite set of vertices $V$ and a finite set of edges $E$. A DAG contains only directed edges, unidirectional connections from one node to another node. A graph is called acyclic if it does not contain any duplicate edges or loops. In a Bayes Network the nodes of the graph represent discrete random

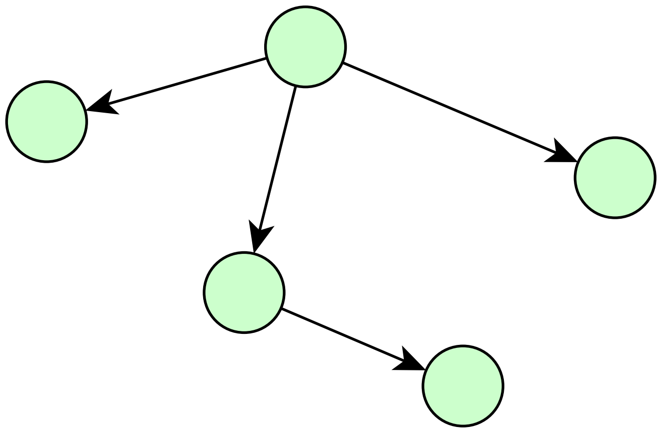


Figure 4.1: Example of a directed acyclic graph

variables and the edges denote the causal relationship, or conditional dependency, between those variables. Each node can take on a finite, mutually exclusive set of states. A probability function is assigned to each node, that returns the probability distribution of the variable represented by the node, given the values assigned to the node's parent nodes. These so called local probability distributions are stored in a table attached to the node [53].
In the following section, the statistical background needed for understanding how building a Bayes Network and inference work is given. Readers who know the basics about discrete random variables, probability distributions, conditional probability and Bayes' law may safely skip this section. Afterwards, the semantics of Bayesian Networks will be explained, followed by details about probabilistic inference on those networks. Finally, the Join Tree Algorithm, which is used for inference in the Automatic Model Based I/O Device and Modality Selection for AALuis, is described thoroughly.

## 4.2 Probability Basics

- A sample space $\Omega$ is the collection of all possible outcomes $\omega$ of an experiment
- Any subset of the sample space $\Omega$ is called an event. We denote events with capital letters.
- The sample space $\Omega$ is composed of $n$ distinct elementary events $\Omega = \{A_1, A_2, A_3, ..., A_n\}$. An elementary event contains only a single outcome. An observed event can be comprised of a subset of elementary events.
- A probability function $P(A)$ assigns a real number to every event $A \subseteq \Omega$ and has to meet the following criteria:
  1. $0 \leq P(A_i) \leq 1 \qquad$ for $1 \leq i \leq n$.
  2. $P(A_1) + P(A_2) + ... + P(A_n) = 1$.
  3. For each event $A = \{A_1, A_2, ..., A_k\}$ that consists of more than one outcome in the sample space: $P(A) = P(A_1) + P(A_2) + ... + P(A_k)$.
- $(\Omega, P)$ is called the probability space.
- Random variables are used to describe elementary events. If the random variable $X$ describes an event $A$, $X = x_i$ means that $X$ takes on the value $x_i$. The following notations are common: $P(A) = P(X = x_i) = P(x_i)$.
- If we consider two random variables $A$ and $B$, the function

$$f(a, b) = P(A = a, B = b) \tag{4.1}$$

  for every pair of values $(a, b)$ is called the joint probability distribution of $A$ and $B$. The joint probability is denoted as $P(A, B)$ or $P(A \cap B)$. Its size is given by the Cartesian product of the set of random variables.

- The conditional probability $P(A|B)$ is the probability of the event $A$, given the prior knowledge that event $B$ has occurred. Its definition is shown in Equation 4.2

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{4.2}$$

The semantic meaning of probability is subject to interpretation. Based on the classical theory of probability, two relevant interpretations were developed. The frequentist interpretation, that is widely known, has a drastically different meaning than the subjective interpretation of probability, often called Bayesian probability.

### The Classical Theory of Probability

The classical theory of probability was originally developed studying the properties of fair games of chance. If an experiment has a discrete number $n$ of mutually exclusive outcomes, the probability of a specific event $A$ is defined as:

$$P(A) = \frac{\text{\# favorable cases for the event A}}{\text{\# number of possible outcomes}}.$$

Therefore, Laplace's definition assigns a number from 0 to 1 to every event.

### The Frequentist Interpretation of Probability

A defined statistical experiment is repeated $n$ times. The number of times a specific event $A$ out of the sample space $\Omega$ (the total set of possible outcomes) occurs is the absolute frequency $F_n(a)$ of the event $A$ and the relative frequency equals

$$f_n(a) := F_n(A)/n.$$

According to the law of large numbers, for $n \to \infty$ the arithmetic mean of the absolute frequencies will converge to a certain value, which the frequentist interpretation of probabilities views as the probability of the occurrence of event $A$. The probability is therefore a property of all $n$ trials. In this context, probabilities are sometimes also called physical probabilities.

### Subjective Probability

In some situations, it is not possible to conduct an experiment multiple times in order to establish the frequencies of its outcomes. In the subjectivist interpretation of probabilities, probability is a numerical value, representing the degree of belief in the occurrence of an event given prior knowledge $E$. This definition enables us to utilize expert domain knowledge and assign probabilities that are conditioned on $E$, in favor of the above definition to events. The following rules have to be met by the numerical values assigned to events:

$$0 \leq P(A|E) \leq 1 \tag{4.3}$$

For two mutually exclusive events $A_1$ and $A_2$:

$$P(A_1 \cup A_2|E) = P(A_1|E) + P(A_2|E). \tag{4.4}$$

In case that the event $A$ has occured and the prior knowledge therefore is $A \cap E$ the following has to hold:

$$P(A \cap B|E) = P(B|A \cap E) \cdot P(A|E). \tag{4.5}$$

### Marginal Distributions

For a joint probability distribution, calculating the probability of one of the random variables, regardless of the value of the others, can be done easily by summing its values of every possible state of the other variables. For the example of two random variables, $A$ and $B$, with $B$ having $i$ different values:

$$P(A) = \sum_i P(A, B_i) \tag{4.6}$$

The term marginal distribution is used because when viewed as a table, these sums can be found in the margins after adding up all the rows and all the lines, as shown in Table 4.1.

| | B | | |
|---|---|---|---|
| A | b1 | b2 | $\sum_i (A_i, B)$ |
| $a_1$ | 0.40 | 0.20 | 0.60 |
| $a_2$ | 0.30 | 0.10 | 0.40 |
| $\sum_j (A, B_j)$ | 0.70 | 0.30 | 1 |

Table 4.1: A table of the marginal distribution of two random variables $A$ and $B$.

## 4.3 Bayes Networks

The following example illustrates the idea of Bayes Networks. The aim of this work is to find a decision making process to determine if a certain output modality is suitable for a user given the circumstances of use. For example, the user's eyesight is a relevant factor when considering if text is a suitable output modality. We assume that age has a direct influence on the eyesight of a person. The information about a user's eyesight is not directly accessible, but the user's age is. We define a probability space with two random variables age (A) and eyesight (E) along with two possible outcomes for A: $\geq$ *65 years* and <*65 years*. Eyesight will have three possible outcomes *good* (+), *impaired* (-), *severely impaired* (–). The joint distribution of these two variables lists all possible combinations of values and therefore consists of six entries. It is depicted in Table 4.2.

| A | E | $P(A,E)$ |
|---|---|---|
| *<65 years* | + | 0.492 |
| *<65 years* | - | 0.287 |
| *<65 years* | – | 0.041 |
| *≥65 years* | + | 0.036 |
| *≥65 years* | - | 0.0684 |
| *≥65 years* | – | 0.0756 |
| | | 1 |

Table 4.2: The joint distribution of Eyesight $E$ and Age $A$

The joint probability distribution assigns a probability to every possible outcome. Since the random variables of a sample space $\Omega$ are per definition exhaustive, one of the events must occur and therefore the sum of all values in the joint probability distribution equals 1. Due to this fact, the numbers in this table are very small and would be even smaller, if the number of possible outcomes is larger than in the given example. That makes the numbers hard to interpret intuitively.
A Bayes Network uses a more convenient representation for the same data, allowing a human to handle the numbers easier, by splitting the joint probability function into several parts. Remember that the definition of conditional probability given in Equation 4.2 contains the joint probability. By rearranging the definition of conditional probability, we obtain Chain Rule of Probability shown in Equation 4.8, which we can use to represent the conditional probability function for Example 1 factorized. The factorization is made up of the conditional probability $P(E|A)$ and $P(A)$.

$$P(E|A) = P(E,A)/P(A) \qquad | \cdot P(A) \tag{4.7}$$

$$P(E,A) = P(E|A) \cdot P(A) \tag{4.8}$$

A Conditional Probability Table (CPT) can be used to store the probability for every possible outcome of a variable, given the value of certain other variables, as shown in Figure 4.2. In this representation the numbers in the CPT are easier to read, because the sum of each line equals 1. A human expert could express her beliefs in the probability of events like this: "60 percent of individuals under the age of 65 years have good eyesight." In a Bayes Network both mentioned approaches to probability can be combined: the probabilities can be physical probabilities obtained from experiments or Bayesian probabilities provided by an expert in the given domain. Apart from that, the promise of a more compact representation of the joint probability function, does not seem to have been fulfilled, as the CPT associated with the node *Eyesight* in Figure 4.2 is not smaller than the whole joint distribution given in Table 4.2. After the concept of independence is explained in more detail, a slightly bigger example of a Bayesian Network will show the advantages of this modeling technique.

| $P(A)$ | |
|---|---|
| *≥65 y* | *<65 y* |
| 0.18 | 0.82 |

$$\begin{tikzcd} A \arrow[dr] & \\ & E \end{tikzcd}$$

| | $P(E\|A)$ | | |
|---|---|---|---|
| $A$ | + | - | – |
| *≥65 y* | 0.20 | 0.38 | 0.42 |
| *<65 y* | 0.60 | 0.35 | 0.50 |

Figure 4.2: Example 1 as a Bayes Network containing the variables Age (A) and Eyesight (E) with conditional probability tables next to the nodes. Some variable values are abbreviated: good (+), impaired (-), severely impaired (–)

### Independence of Events

In the first example it was assumed that the random variable *Age* had a direct influence on the random variable *Eyesight*. Knowledge about the value of *Age* influences our belief about *Eyesight*. We will rate the probability of bad eyesight higher if a person is 65 years old or older. In other words, bad eyesight is not independent of a person's age. Independence means that the occurrence of an event $a$ has no effect on the occurrence of event $b$, in that $P(b|a) = P(b)$ and we denote that fact by

$$P \perp\!\!\!\perp (a \perp b). \tag{4.9}$$

Independence is a symmetric property, which means that $a \perp b$ implies $b \perp a$. [52]

**Conditional independence** In many cases two events are not independent, when the two are considered together. However, they can be independent, when there is evidence about the occurrence of a third event. If $P(a|b, c) = P(a|c)$ holds, it means that if $c$ is already known, or if $P(b \cap c) = 0$, additional knowledge of $b$ does not change the probability of $a$. This is called conditional independency, and $a$ being conditionally independent of $b$ given $c$ in $P$ is denoted by:

$$P \perp\!\!\!\perp (a \perp b|c) \tag{4.10}$$

It is also possible to identify two or more random variables as independent. Given a distribution $P$, $A$ is independent of $B$ given $C$, if $P$ satisfies $(A = a \perp B = b|C = c)$ for all values $a \in Val(A)$, $b \in Val(B)$ and $c \in Val(C)$. This definition can also be written like this:

$$P \text{ satisfies } (A \perp B|C) \text{ if and only if } P(A, B|C) = P(A|C)P(B|C). \tag{4.11}$$

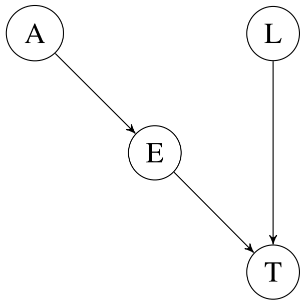


Figure 4.3: Example 2: A Bayes Network with four nodes Eyesight (E), Age (A), Text (T), Language reception (L)

If $C$ is empty, and $(A \perp B)$ holds, they are said to be marginally independent. [52]

## Independence Assumptions in Bayes Networks

Understanding independence of random variables is a basic prerequisite to explain the idea of Bayesian Networks in detail. To that end, the first Bayesian Network example from above will be extended by two nodes. Example 2 will model the influences on a random variable *Text*, that should tell us whether text is a preferable output modality for a specific user. Its domain therefore shall be binary, containing the values *yes* and *no*. The user's eyesight determines whether someone can easily read a text, as does the person's ability to understand language. The variable *Language reception* has a domain of three values: *good*, *impaired* and *severely impaired*. Combining every possible outcome from those four variables makes the joint distribution from Example 1 grow from 6 entries to 36 entries.

Figure 4.3 shows the Bayesian Network for Example 2, which graphically encodes the assumptions about independence of the variables. *Age* (A) does not directly influence the level of agreement on *Text* (T) as a good output modality, so there is no edge from *Age* (A) to *Text* (T). *Eyesight* (E) and *Language reception* (L) do have an influence, so edges were placed from those nodes to *Text* (T). To calculate the joint probability function $P(T, A, E, L)$ the definition for conditional probability introduced in Equation 4.2 is used to obtain the general Chain Rule of Probability. Extending Equation 4.8 to multiple variables results in a factorization shown in Equation 4.12.

$$P(A_1, ..., A_n) = P(A_1|A_2, ..., A_n) \cdot P(A2, ..., A_n) \tag{4.12}$$

It is obvious that the rule can iteratively be applied to the second term as well, leading to

$$P(A_2, ..., A_n) = P(A_2|A_3, ..., A_n) \cdot P(A_3, ..., A_n) \tag{4.13}$$

The result of applying this principle to all terms is depicted in Equation 4.14

$$P(A_1, ..., A_n) = P(A_1|A_2, ..., A_n) \cdot P(A_2|A_3, ..., A_n) \cdot P(A_3|A_4, ..., A_n) \cdot ... \cdot P(A_n) \tag{4.14}$$

Applying this process to the distribution of Example 2 results in the factorization shown in Equation 4.15:

$$P(T, A, E, L) = P(T|A, E, L) \cdot P(A|E, L) \cdot P(E|L) \cdot P(L) \tag{4.15}$$

This equation however does not take into account that we already defined some of our variables as being independent of each other. Consider the third term $P(E|L)$. The probability of a user having good eyesight will not change if we have information about their language reception abilities, otherwise we would have connected those nodes with an edge. Can we safely treat those two events as $P \perp\!\!\!\perp (E \perp L)$ so that $P(E|L) = P(E)$? The semantics contained in a Bayesian Network graph is to be viewed as a set of independence assertions. They are also equally reflected by the annotation of the graph with the CPT as shown in Figure 4.4 This means that by not placing an edge from *Eyesight* to *Language reception* or vice versa we have declared the two variables as independent. Formally, we write:

$$\forall A_i : \quad (A_i \perp \text{NonDescendants}_{A_i} | Pa^G_{A_i}) \tag{4.16}$$

where $Pa_G(A_i)$ denotes the parent nodes of $A_i$ in the graph $G$ [52].

But while *Eyesight* is independent of *Language reception*, according to the graph shown in Figure 4.3 it is not independent of *Age*. Simplifying $P(E|L)$ to $P(E)$ obviously does not reflect the entirety of the encoded independence assertions correctly, because the dependence on $A$ is missing. The Chain Rule for Bayes provides the right tool to correctly factorize a joint probability function modeled by a Bayesian Network.

## The Chain Rule for Bayes Networks

The Chain Rule for Bayes formalizes the factorization of a joint probability function by conditional probabilities.

$$P(A_1, ..., A_n) = \prod_{i=1}^{n} P(A_i|Pa_G(A_i)) \tag{4.17}$$

where $Pa_G(A_i)$ denotes the parent nodes of $A_i$ in the graph $G$.

Applying the Chain Rule for Bayes to Example 2 results in a simpler equation than the one obtained from applying the General Chain Rule of Probability in Equation 4.15:

$$P(T, A, E, L) = P(T|E, L) \cdot P(A) \cdot P(E|A) \cdot P(L) \tag{4.18}$$

| $P(L)$ | | |
|---|---|---|
| + | - | – |
| 0.88 | 0.1 | 0.02 |

| $P(A)$ | |
|---|---|
| ≥*65 y* | <*65 y* |
| 0.18 | 0.82 |

$$\begin{tikzcd}
A \arrow[dr] & & L \arrow[dd] \\
& E \arrow[dr] & \\
& & T
\end{tikzcd}$$

| | $P(E\|A)$ | | |
|---|---|---|---|
| $A$ | + | - | – |
| ≥*65 y* | 0.20 | 0.38 | 0.42 |
| <*65 y* | 0.60 | 0.35 | 0.05 |

| | | $P(T\|E,L)$ | |
|---|---|---|---|
| $E$ | $L$ | *yes* | *no* |
| + | + | 0.95 | 0.5 |
| | - | 0.55 | 0.45 |
| | – | 0.3 | 0.7 |
| - | + | 0.7 | 0.3 |
| | - | 0.45 | 0.55 |
| | – | 0.2 | 0.8 |
| – | + | 0.45 | 0.55 |
| | - | 0.25 | 0.75 |
| | – | 0.05 | 0.95 |

Figure 4.4: The graph for Example 2 with Conditional Probability Tables of the variables Language reception (L), Age (A), Eyesight (E) and Text (T). Variable values are partly abbreviated: good ( +), impaired (-), severly impaired (–)

## 4.4 Inference on a Bayes Network

Modeling a joint probability distribution as a Bayesian Network allows to perform inference under uncertainty. Some variables are typically observed in the real world and instantiated in the model while some remain unknown. The joint probability distribution is used to estimate the probability distribution over the not instantiated variables in a so called probability query. Additionally, the Bayesian Network can also be queried for the probability of evidence $P(e)$, the Most Probable Explanation (MPE) [61] or the Maximum A Posteriori Hypthesis (MAP) [52]. For the task of Automatic IO Device and Modality Selection in AALuis only the probability query was used, so it will be explained in the upcoming section.

### Probability Query

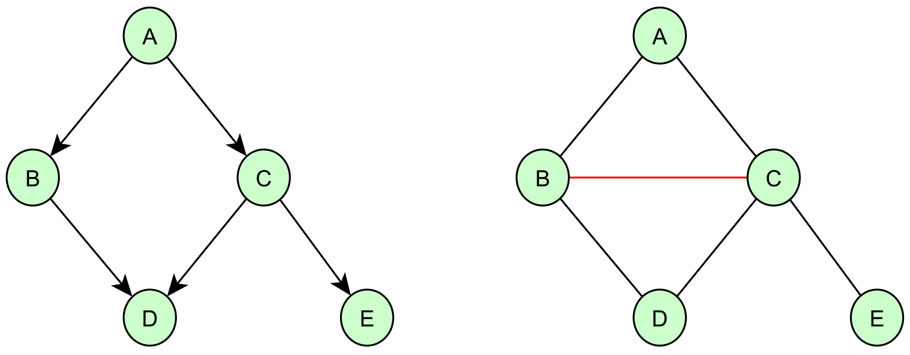
At the beginning of a probability query, we have a joint probability distribution represented in the form of a Bayesian Network. Some of the variables can be observed, which means that their values are known. This subset of random variables $E$ is called evidence, their instantiation $e$. The query variables are another subset of random variables $Y$. The probability query is used to obtain $P(Y|E = e)$, that is the posterior probability distribution of $Y$, given the prior knowledge that $E = e$. This probability is equivalent to the marginal distribution over $Y$ conditioning on $e$ [52]. But as explained before, the Bayes Network does not store the joint probability distribution of $Y$ and $E$, but the independence assertions of variables and their CPTs. Therefore, the main task of the probability query is to compute the joint probability function of the variables and marginalizing over it. Since the joint probability table can be very large, an algorithm is needed that achieves this in reasonable time and memory space. It was already shown that using the Chain Rule for Bayes from Equation 4.3 would result in the joint probability function. With a rising number of variables however the multiplication can take unreasonably long and the space required to store the complete joint probability function is in $O(d^n)$ where $d$ is the size of the largest domain and $n$ the number of variables, the maximum number of combinations of the variables' values.
To optimize this problem, many different inference algorithms have been developed. Some perform exact inference while others aim at giving an approximate result. Variable Elimination [91], Recursive Conditioning [22], Join tree or Junction tree, and the Monte Carlo Markov chain [52] are well known examples. The Junction Tree algorithm is be used in the software described in Chapter 6 and will now be explained in detail and illustrated using a small example. Details on the algorithm and the basis for the given example were obtained from Kahle et al. [50] and course study notes by Prof. Duncan F. Gillies [39] [38].

### Junction Tree Algorithm

The Junction Tree algorithm is an algorithm for Bayesian Networks that performs exact inference. It was introduced by Judea Pearl in 1982 [67] for directed causal trees. Since then it has been further developed into several variations of the general idea. Notable variants were introduced by Shenoy and Shafer who proposed to transform the directed graph to an undirected, qualitative Markov tree for the algorithm in 1986 [74]. It will be presented below. Other notable additions include the algorithm by Lauritzen-Spiegelhalter [55] from 1988 and the commercial implementation of the algorithm in the software product HUGIN[1] [4]. The Junction Tree algorithm uses techniques of graph theory to convert the Bayesian Network graph, reducing its size and facilitating inference for many Bayesian Networks. Some graphs however have a structure that cannot be significantly optimized using this algorithm. The algorithm comprises of the following steps, which will be explained in detail below:

- Moralization
- Triangulation

[1]HUGIN - Handling Uncertainty In General Inference Network



(a) A DAG

(b) The DAG after moralization.

Figure 4.5: A DAG before and after moralization.

- Building a Junction Tree
- Assigning Potentials and Initialization
- Selecting a Root Node
- Message Passing
- Marginalization

### Moralization

At the beginning the DAG is transformed to an undirected graph by connecting all parents of each node with each other with new edges and removing the directions of existing edges. Since the directed edges were used to encode dependency information in the original DAG, eliminating the directions results in a loss of independence information. However, this does not impede the capability of the algorithm to produce an exact inference result. A graph before and after moralization is shown in Figure 4.5.

### Triangulating the Graph

The next step is to convert the moralized graph to a triangulated graph. In a triangulated graph, every cycle of length $n \geq 4$ has an edge that connects two nodes of the loop but is not part of the loop. Triangulating the graph thus means adding edges according to this definition until the definition is satisfied. In a graph $G$ there can be more than one valid way to triangulate it. The junction tree algorithm works with every possible triangulated version of $G$. A triangulated graph is also called chordal graph.

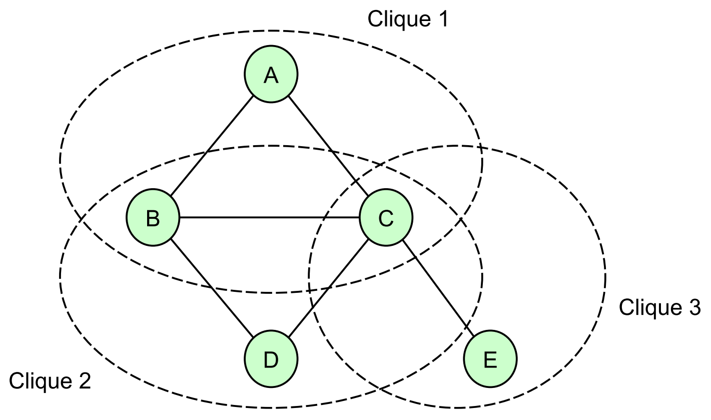


Figure 4.6: In the moralized and triangulated graph, three maximal cliques of adjacent nodes can be identified.

### Building a Junction Tree

In the next step of the algorithm, the graph is converted to a junction tree. A tree is an undirected graph that contains exactly one unique path between any two vertices and includes no loops. To create the junction tree, first a so called clique graph $H$ has to be built. A clique merges vertices of a graph that are adjacent to each other to form a clique node. Such a clique is called maximal, if no more vertices can be added to the clique. These cliques can be connected with each other to form a clique graph by labeling the edges with the nodes that equal the intersection of two connected cliques. In some variants of the algorithm, those labeled edges are also viewed as nodes and called separator nodes.
Now, on the basis of the clique graph, the junction tree can be built. A junction tree is a tree that is a subgraph[2] of the clique graph containing every node of the clique graph and satisfying the junction tree property. The junction tree property requires that for every pair of cliques $A$, $B$ with $A \cap B = S$ all cliques on the path from $A$ to $B$ contain $S$. The grouping of graph nodes to cliques is shown in Figure 4.6. Figure 4.7 depicts the finished junction tree, built on the basis of these cliques. The edges are labeled with the original graph nodes, appearing in both adjacent clique nodes.

### Assigning Potentials and Initializing

The described transformation process can result in different junction trees, which does not impact the correctness of inference. Every junction tree is called a potential representation. A potential representation consists of subsets $W_i$ of the nodes $V$, the cliques of the junction tree. The notation $W_i$ will be used to address the current clique node, $W_1$ representing the root. $W_{i+1}$

[2]A graph $H(V_H, E_H)$ is a subgraph of a graph $G(V, E)$ if the set $V_H$ is a subset of $V$ and $E_H$ is a subset of $E$.

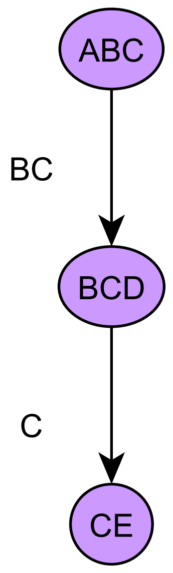


Figure 4.7: The join tree, built from the clique selection shown in Figure 4.6.

designates a child node of the current node. A potential is defined by a potential function $\Psi$ so that:

$$P(V) = \prod_i \Psi(W_i) \tag{4.19}$$

It attaches a probability to every joint state of the variables contained in the clique $W_i$ and is useful for calculating the joint probability table $P(W_i)$. We have now found a structure that compressed the nodes of the original graph to cliques and still represents the same joint probability as the graph, if the clique potentials $\Psi$ are initialized. To that end, for every clique $W_i$ the subset of nodes $X$, whose parents in the original graph also belong to the same clique, is identified. The potential function is set to:

$$\Psi(W_i) = \prod_X P(X_i|Pa(X_i)) \tag{4.20}$$

The probability of a root node of the original graph is multiplied to the potential of one clique in which it occurs and every CPT from $G$ is assigned exactly once to a potential representation. If a CPT has already been used, it is not added to the product forming another potential. Figure 4.8 shows the original graph on the left side. Its nodes are labeled with the assigned probability functions. On the right side, every probability function is assigned exactly once to a clique node containing the corresponding graph node, according to Equation 4.20. Multiplying these probability functions leads to the potential function of the clique nodes.

By initializing the clique potentials like that, they reflect the original joint distribution, because

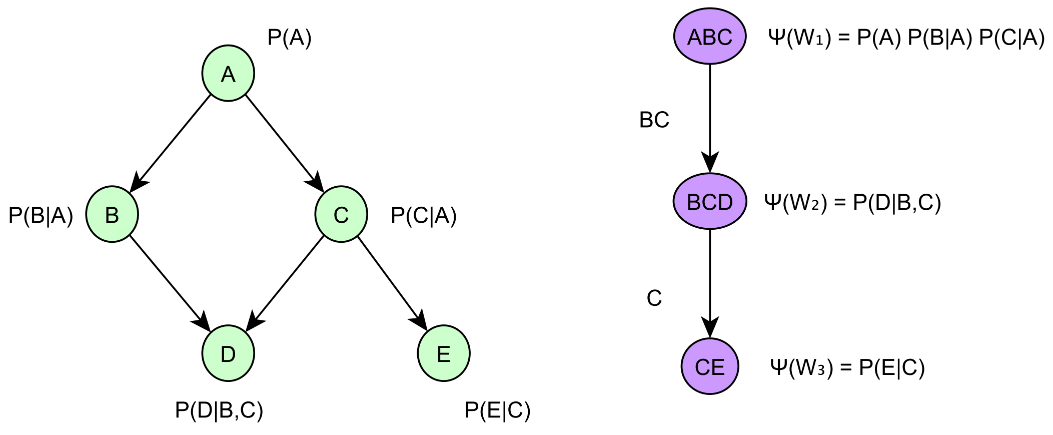


Figure 4.8: The original Bayes Network (left), next to the associated join tree. The nodes of the Bayes Network are labeled with the probabilities they represent. To assign the potentials to the cliques of the join tree, each of these probability functions is assigned exactly once to a clique (see Equation 4.20).

the potential functions represent a partitioning of the Chain Rule for Bayes (Equation 4.3):

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|Pa_G(X_i)) = \Psi(W_i) = \prod_{X} P(X_i|Pa(X_I)) \tag{4.21}$$

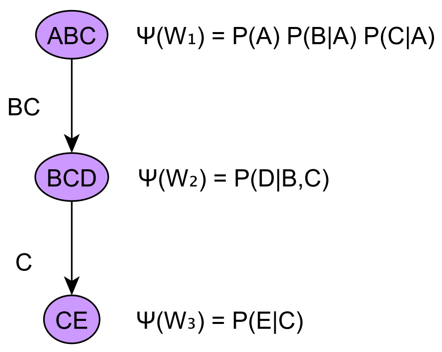
For successful inference, the goal is to find $P(W_i)$. This way, individual probabilities can be easily calculated by marginalizing one of the joint probabilities of a clique that contains the desired variable. Every clique can be described to contain variables $S$ that appear in their parent clique, while the variables $R$ do not. The joint probability distribution of one clique node can therefore be written as $P(W_i) = P(R_i, S_i)$. Remembering the Chain Rule for Bayes (Equation 4.3), the same fact can be expressed by using a conditional probability:

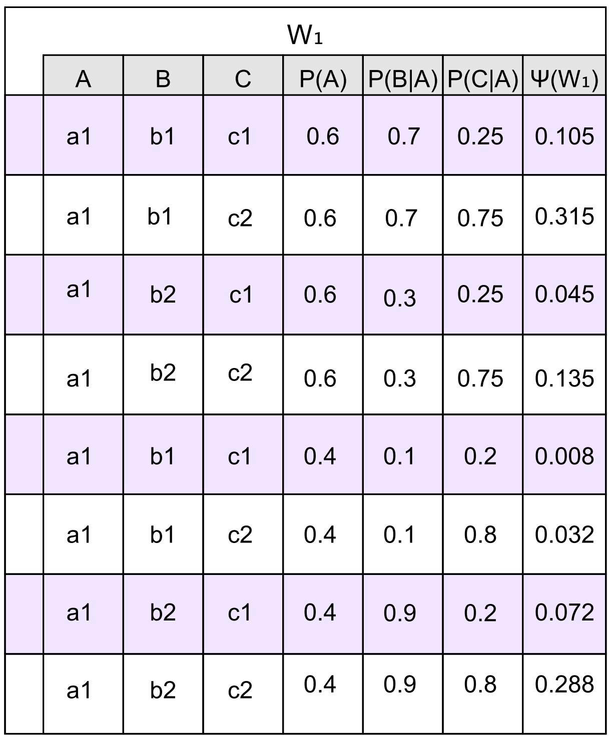
$$P(W_i) = P(R_i, S_i) = P(R_i|S_i)P(S_i) \tag{4.22}$$

This relationship will be used in the message passing part of the algorithm.

### Selecting an Arbitrary Root Node

After assigning the CPTs to clique nodes, one of the nodes has to be selected as root node. In the following step, called message passing, messages are first passed from the leaves to this root node and then back again to the leaves.
In the given example shown in Figure 4.7, the top node, labeled $ABC$ will serve as root node. Figure 4.9 shows the example junction tree. The associated table for the node $W_1$ shows each





| W1 | | | | | | | |
|---|---|---|---|---|---|---|---|
| | A | B | C | P(A) | P(B\|A) | P(C\|A) | $\Psi(W_1)$ |
| | a1 | b1 | c1 | 0.6 | 0.7 | 0.25 | 0.105 |
| | a1 | b1 | c2 | 0.6 | 0.7 | 0.75 | 0.315 |
| | a1 | b2 | c1 | 0.6 | 0.3 | 0.25 | 0.045 |
| | a1 | b2 | c2 | 0.6 | 0.3 | 0.75 | 0.135 |
| | a1 | b1 | c1 | 0.4 | 0.1 | 0.2 | 0.008 |
| | a1 | b1 | c2 | 0.4 | 0.1 | 0.8 | 0.032 |
| | a1 | b2 | c1 | 0.4 | 0.9 | 0.2 | 0.072 |
| | a1 | b2 | c2 | 0.4 | 0.9 | 0.8 | 0.288 |

Figure 4.9: An example junction tree with a table showing probabilities for each original graph node and the resulting potential function.

state the variables of the clique node can take on, together with their probabilities from the original graph. $\Psi(W_1)$ is the potential for each possible state of the clique node.

### Message Passing

The aim of message passing is to update the probabilities with the changes introduced by evidence. Thus, first observations of events have to be considered. Evidence of the state of a variable is set in exactly one of the clique nodes containing the observed variable. Setting evidence means freezing the values to the observed state. This is achieved by setting the probability table of the junction tree clique to $0$ where the value of the variable does not equal the one observed. Figure 4.10 shows the potential table of clique node $W_2$ before and after setting evidence.
Then, the message passing part of the algorithm can be performed. A message in this case, is a probability table, which is created, passed to a clique node and processed by the receiving clique node to update its probability table. Consequently the desired result, the probability of one or several of the variables, given the evidence, can be calculated. First, upward messages are passed from clique to clique, until they reach the root node. In the second pass, each node, beginning with the root node, creates messages for its children and passes them down.

Original potential table of W2

| W2 | | | |
|---|---|---|---|
| B | C | D | P(D\|B,C) = Ψ(W2) |
| b1 | c1 | d1 | 0.8 |
| b1 | c1 | d2 | 0.2 |
| b1 | c2 | d1 | 0.9 |
| b1 | c2 | d2 | 0.1 |
| b2 | c1 | d1 | 0.7 |
| b2 | c1 | d2 | 0.3 |
| b2 | c2 | d1 | 0.05 |
| b2 | c2 | d2 | 0.95 |

Potential table after
D is observed to equal d1

| W2 | | | |
|---|---|---|---|
| B | C | D | Ψ(W2) |
| b1 | c1 | d1 | 0.8 |
| b1 | c1 | d2 | **0** |
| b1 | c2 | d1 | 0.9 |
| b1 | c2 | d2 | **0** |
| b2 | c1 | d1 | 0.7 |
| b2 | c1 | d2 | **0** |
| b2 | c2 | d1 | 0.05 |
| b2 | c2 | d2 | **0** |

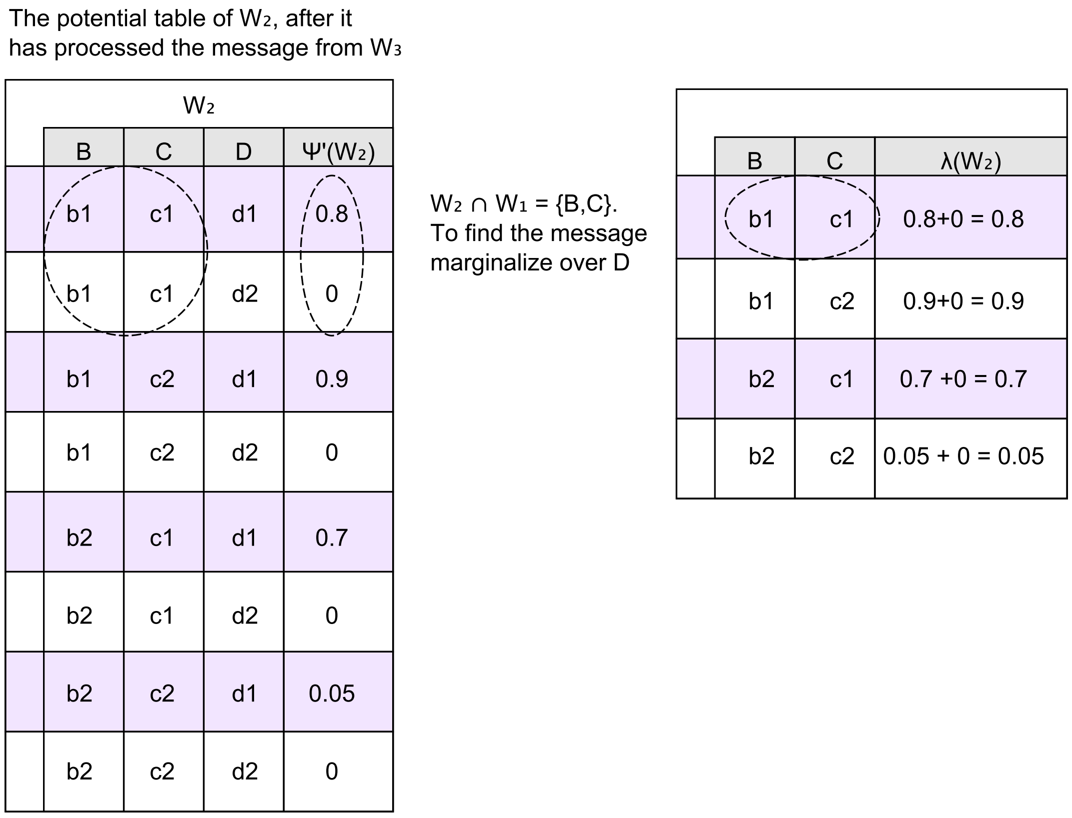
Figure 4.10: The potential table of $W_2$ before and after evidence of $d1$ was observed.

**Upward Pass** The aim of the upward pass is to find $P(R_i|S_i)$ for every clique node. A leaf, a node that has only one neighbor and is not the root node, is selected as a starting point for the message passing. To create the upward message $\lambda(W_i)$, first the table of that junction tree node is marginalized over the variables not found in the parent clique, so that it contains only variables in $S$, the variables also appearing in the parent clique node.

$$\lambda(W_i) = \sum_R \Psi(W_i) \tag{4.23}$$

The resulting table is the message $\lambda$. It is used to find $P(R_i|S_i)$ using point-wise division and then passed to the parent node.

$$P(R_i|S_i) = \Psi(W_i)/\lambda(W_i) \tag{4.24}$$


The potential table of W₂, after it has processed the message from W₃
W₂ ∩ W₁ = {B,C}.
To find the message marginalize over D

| W2 | | | |
|---|---|---|---|
| B | C | D | Ψ'(W2) |
| b1 | c1 | d1 | 0.8 |
| b1 | c1 | d2 | 0 |
| b1 | c2 | d1 | 0.9 |
| b1 | c2 | d2 | 0 |
| b2 | c1 | d1 | 0.7 |
| b2 | c1 | d2 | 0 |
| b2 | c2 | d1 | 0.05 |
| b2 | c2 | d2 | 0 |

| B | C | λ(W2) |
|---|---|---|
| b1 | c1 | 0.8+0 = 0.8 |
| b1 | c2 | 0.9+0 = 0.9 |
| b2 | c1 | 0.7 +0 = 0.7 |
| b2 | c2 | 0.05 + 0 = 0.05 |

Figure 4.11: $W_2$ after it has processed the message it received from $W_3$. To find the message for $W_1$, select the variables present in both cliques and marginalize over the others.

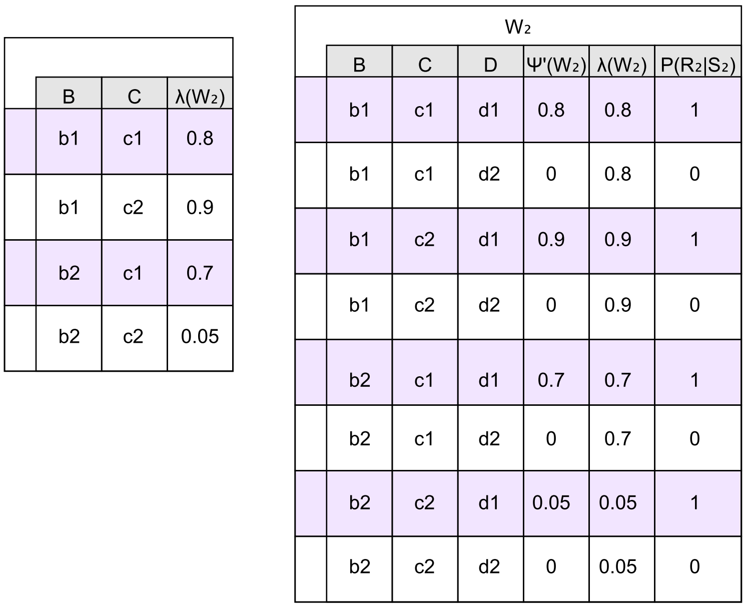
Point-wise division means that the division is performed separately for every cell that exists in both tables. Some values in the table might be 0 and therefore cause a problem acting as divisor in such a division. In those cases, the cell of the resulting table will be set to 0 by definition.

Shifting the focus to the clique node receiving the message $\lambda$ from its child, the former parent node becomes the new current node $W_i$. It multiplies its own table point-wise with $\lambda$, thereby replacing the previous $\Psi(W_i)$ by

$$\Psi'(W_i) = \Psi(W_i) \times \lambda(W_{i+1}) \tag{4.25}$$

When the node has received and processed messages from every child node, it is ready to generate its own message. The process is repeated until the root node is reached.

The upward message pass from $W_3$ to $W_2$ in the example does not change any values in $W_2$, so this step is not explicitly shown. Figure 4.11 shows the creation of the message for the second clique, $W2$. The values for the message are found by marginalization, according to Equation 4.23. In this case, $R = \{D\}$, as $D$ does not occur in the parent clique. The dotted ellipses hint at

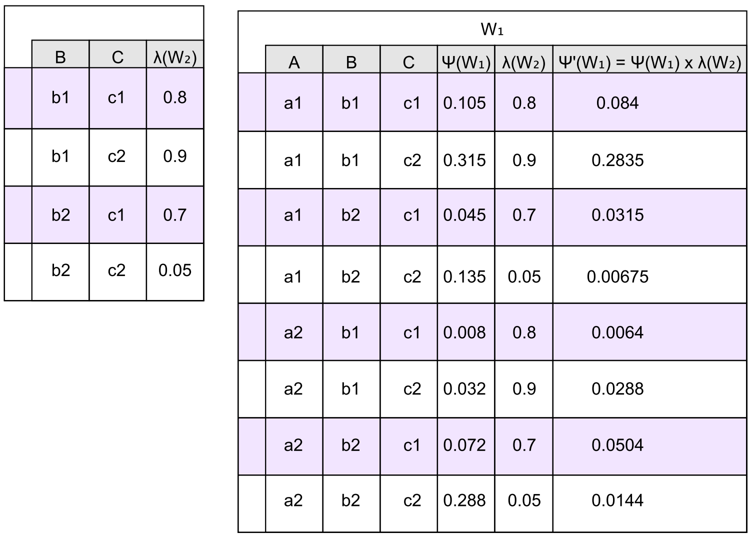


| B | C | $\lambda(W_2)$ |
|---|---|---|
| b1 | c1 | 0.8 |
| b1 | c2 | 0.9 |
| b2 | c1 | 0.7 |
| b2 | c2 | 0.05 |

$W_2$

| B | C | D | $\Psi'(W_2)$ | $\lambda(W_2)$ | $P(R_2\|S_2)$ |
|---|---|---|---|---|---|
| b1 | c1 | d1 | 0.8 | 0.8 | 1 |
| b1 | c1 | d2 | 0 | 0.8 | 0 |
| b1 | c2 | d1 | 0.9 | 0.9 | 1 |
| b1 | c2 | d2 | 0 | 0.9 | 0 |
| b2 | c1 | d1 | 0.7 | 0.7 | 1 |
| b2 | c1 | d2 | 0 | 0.7 | 0 |
| b2 | c2 | d1 | 0.05 | 0.05 | 1 |
| b2 | c2 | d2 | 0 | 0.05 | 0 |

Figure 4.12: Before the message $\lambda(W_2)$ is passed to the parent of $W_2$, it is used to find $P(R_2|S_2)$.



| B | C | $\lambda(W_2)$ |
|---|---|---|
| b1 | c1 | 0.8 |
| b1 | c2 | 0.9 |
| b2 | c1 | 0.7 |
| b2 | c2 | 0.05 |

$W_1$

| A | B | C | $\Psi(W_1)$ | $\lambda(W_2)$ | $\Psi'(W_1) = \Psi(W_1) \times \lambda(W_2)$ |
|---|---|---|---|---|---|
| a1 | b1 | c1 | 0.105 | 0.8 | 0.084 |
| a1 | b1 | c2 | 0.315 | 0.9 | 0.2835 |
| a1 | b2 | c1 | 0.045 | 0.7 | 0.0315 |
| a1 | b2 | c2 | 0.135 | 0.05 | 0.00675 |
| a2 | b1 | c1 | 0.008 | 0.8 | 0.0064 |
| a2 | b1 | c2 | 0.032 | 0.9 | 0.0288 |
| a2 | b2 | c1 | 0.072 | 0.7 | 0.0504 |
| a2 | b2 | c2 | 0.288 | 0.05 | 0.0144 |

Figure 4.13: The message $\lambda(W_2)$ is used to update the probabilities in $W_1$.

the marginalization process. The message is first used to calculate $P(R_2|S_2)$, shown in Figure 4.12. Then it is passed on to the parent clique node $W_1$. By multiplying the potential function of $W_1$ with the message, $\Psi(W_1)$ is replaced with the probabilities, updated with the knowledge of the evidence. This step is shown in Figure 4.13.

**Downward Pass** After the root node has processed all incoming messages from the upward pass and $P(R1|S1)$ has been calculated, it creates a downward message $P(S_{i+1})$ for each child by marginalizing out those variables not existent in that child. The message is passed down to the child and multiplied with the potential table, which equals $P(R_i|S_i)$ as a result of the upward pass.

$$P(W_i) = P(R_i|S_i) \times P(S_i) \tag{4.26}$$

Afterwards, $P(W_i)$ is used to calculate the message $P(S_i)$ for the child node. When every leaf node has received a downward message, all potential tables contain the values for $P(W_i)$, which allows us to find the updated probability for each variable.
Figure 4.14 shows the state of the root node $W_1$ after the upward message passing is complete. $P'(W_1)$ has been calculated by marginalizing $P(R_1|S_1)$ over all variables of the root clique and multiplying each value with that sum. Next to the table of $W_1$, the downward message $P'(S_2)$ is shown. It is calculated by marginalizing over the variables not present in $W_2$ and passed down. In $W_2$ the message is multiplied point wise with the values of $P(R_2|S_2)$ to find $P'(W_2)$. The process is repeated for $W_3$.

**Evaluating Desired Marginals**

To obtain the inference result after the message passing is completed, any potential function containing the desired variable can be marginalized on that variable to find its posterior probability. Figure 4.15 shows the table of node $W_3$ after the upward and downward message passes. To find, e.g. the updated probability of the variable $E$, the sum over all states of $E$, regardless of the state of $C$, is calculated:

$$P(e1) = 0.068136431 + 0.065931784 = 0.134068216 \tag{4.27}$$
$$P(e2) = 0.272545724 + 0.59338606 = 0.865931784 \tag{4.28}$$

Figure 4.16 shows a screenshot of the same example modeled in SamIam. When $D$ is observed to equal $d1$, the inference mechanism returns the same result for the updated probability of $E$, as shown in Figure 4.15.

One of the advantages of the Junction Tree algorithm is that the tree has to be built only once. For subsequent probability queries, triggered by new evidence, only the process message passing has to be repeated.

$W_1$

| A | B | C | $P(W_1)$ |
|---|---|---|---|
| a1 | b1 | c1 | 0.16608997 |
| a1 | b1 | c2 | 0.56055363 |
| a1 | b2 | c1 | 0.06228374 |
| a1 | b2 | c2 | 0.013346515 |
| a2 | b1 | c1 | 0.012654474 |
| a2 | b1 | c2 | 0.056945131 |
| a2 | b2 | c1 | 0.099653979 |
| a2 | b2 | c2 | 0.028472565 |

| B | C | $P'(S_2)$ |
|---|---|---|
| b1 | c1 | 0.178744439 |
| b1 | c2 | 0.617498764 |
| b2 | c1 | 0.161937716 |
| b2 | c2 | 0.041819081 |

$W_2$

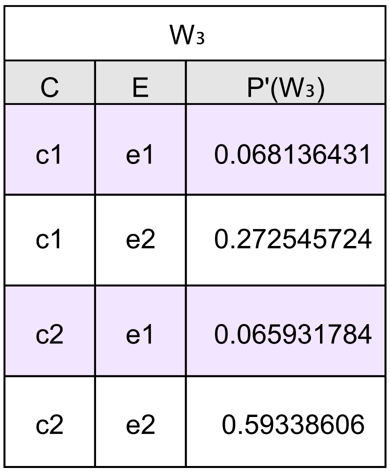
| B | C | D | $P(R_2\|S_2)$ | $P'(S_2)$ | $P(W_2)$ |
|---|---|---|---|---|---|
| b1 | c1 | d1 | 1 | 0.178744439 | 0.178744439 |
| b1 | c1 | d2 | 0 | 0.178744439 | 0 |
| b1 | c2 | d1 | 1 | 0.617498764 | 0.617498764 |
| b1 | c2 | d2 | 0 | 0.617498764 | 0 |
| b2 | c1 | d1 | 1 | 0.161937716 | 0.161937716 |
| b2 | c1 | d2 | 0 | 0.161937716 | 0 |
| b2 | c2 | d1 | 1 | 0.041819081 | 0.041819081 |
| b2 | c2 | d2 | 0 | 0.041819081 | 0 |

| C | $P'(S_3)$ |
|---|---|
| c1 | 0.340682155 |
| c2 | 0.659317845 |

$W_3$

| C | E | $P(R_3\|S_3)$ | $P'(S_3)$ | $P(W_3)$ |
|---|---|---|---|---|
| c1 | e1 | 0.2 | 0.340682155 | 0.068136431 |
| c1 | e2 | 0.8 | 0.340682155 | 0.272545724 |
| c2 | e1 | 0.1 | 0.659317845 | 0.065931784 |
| c2 | e2 | 0.9 | 0.659317845 | 0.59338606 |

Figure 4.14: The downward message $P'(S_2)$ is found by marginalizing $P(W_1)$ over $A$ and is then passed down to $W_2$. There it is multiplied with the values of $P(R_2|S_2)$ to find $P(W_2)$. The process is repeated for $W_3$.



| W3 | | |
|---|---|---|
| C | E | P'(W3) |
| c1 | e1 | 0.068136431 |
| c1 | e2 | 0.272545724 |
| c2 | e1 | 0.065931784 |
| c2 | e2 | 0.59338606 |

Figure 4.15: The potential table of clique node $W3$ after the upward and downward message passing are complete. The table can be used to find the updated probability values $P'(W3)$ for the states of $C$ and $E$ by building the sums of the corresponding rows. Equations 4.27 and 4.28 give the resulting probabilities of $e1$ and $e2$.

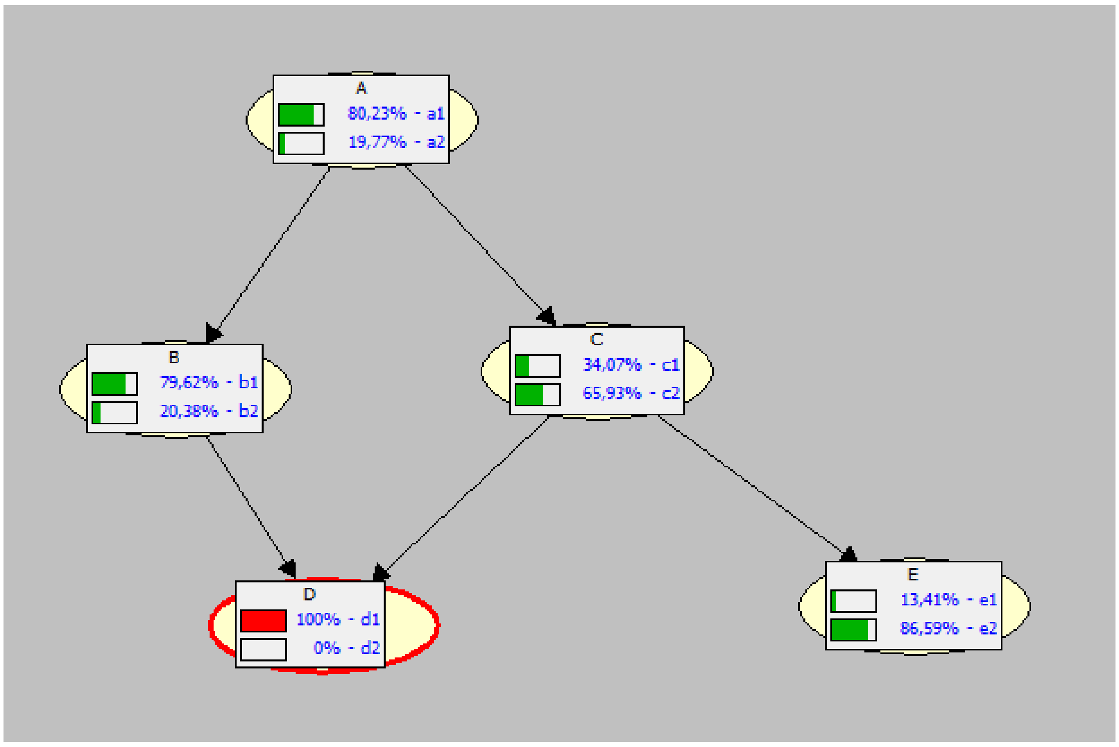


Figure 4.16: A screenshot of the example illustrated in Figures 4.5 to 4.15 modeled in SamIam. Evidence is set to $D = d1$. The nodes show the updated probability values, given this evidence.

## 4.5 SamIam

SamIam[3] stands for Sensitivity Analysis Modeling Inference And More. It is a software developed by the Automated Reasoning Group at University of California Los Angeles. The tool supports modeling of Bayesian Networks with a graphical user interface and offers an inference engine to perform queries and various analyses on the Bayes Network. It is written in Java and supports files of the Hugin and Genie file formats. It also offers an inference library API that can be used in Java Applications. The use of this API offers the option to edit a Bayes network created in the graphical SamIam tool programmatically in your own application and to perform inference queries. It is used for the modeling process as well as for the implementation of the Automatic IO Device and Modality Selection for AALuis.

[3]University of California Los Angeles, Automated Reasoning Group. Samiam - sensitivity analysis, modeling, inference and more. http://reasoning.cs.ucla.edu/samiam/.

CHAPTER 5

# Modeling

This chapter describes the process of modeling and the individual parts of the created model. All the previously introduced concepts, including the sensors of the smart home, the user profile describing the current user, and the devices available to the AALuis system are incorporated. The model is not based on a concrete household. It was created for an example setting and does not reflect every conceivable device, modality or sensor. Therefore it will be refered to as the default model. In a real application, it would be adapted according to the context of use. The default model currently comprises of 52 nodes and 71 edges, including six devices and four modalities. It was created using the software SamIam. A screenshot of the default model using SamIam is shown in Figure 5.1.
The following sections will describe how the model was developed, give details about the parts of the final model and show how it can be expanded for new devices and modalities.

## 5.1 Creating the Output Device and Modality Model

The initial modeling approach included one node for device and one for modality, the instances of each reflecting the respective choices. Each instance is assigned a probability in the Conditional Probability Table (CPT). After inference the modality and device which were assigned the highest probability equal the result, the best option given the current context. Figure 5.2 shows the two nodes as an example of this approach. Figure 5.3 depicts the CPT of the modality node of Figure 5.2, to give an impression of the size of the table, when a binary node has three binary parents.
However, this straightforward solution has several drawbacks.

- It is not evident which device supports which modality. The best device may not support the best modality and vice versa.

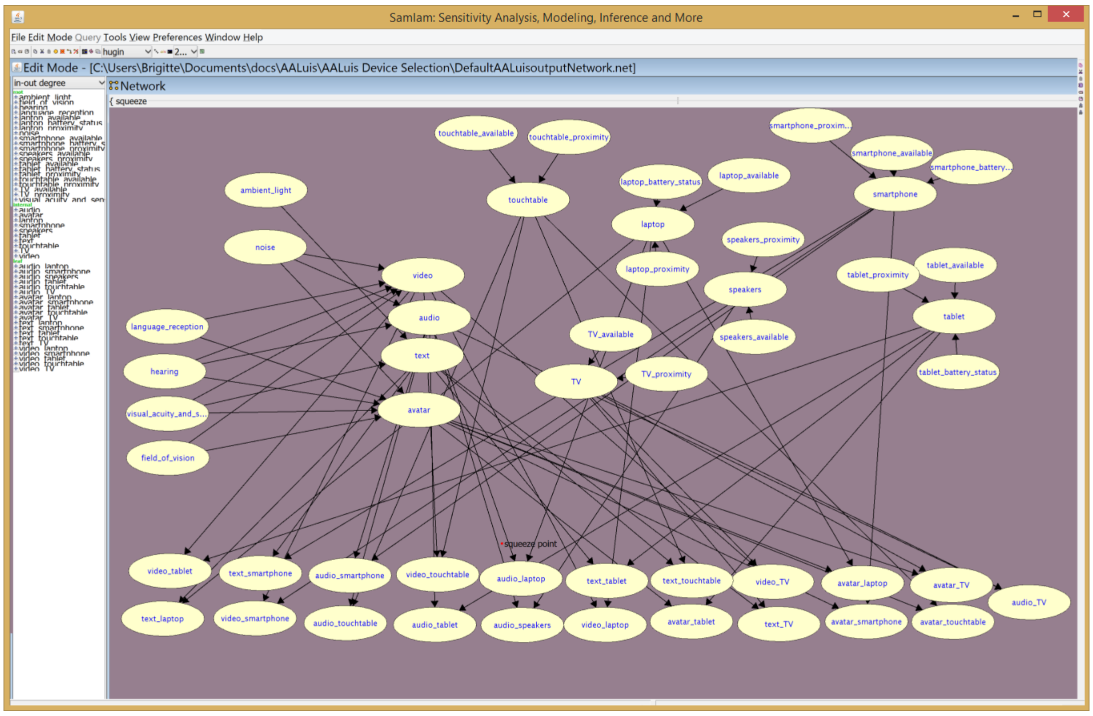


Figure 5.1: The default AALuis Output Network modeled in SamIam in an overview.

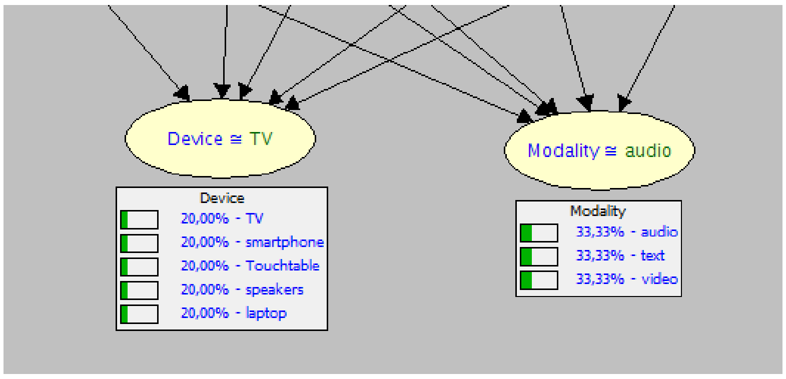
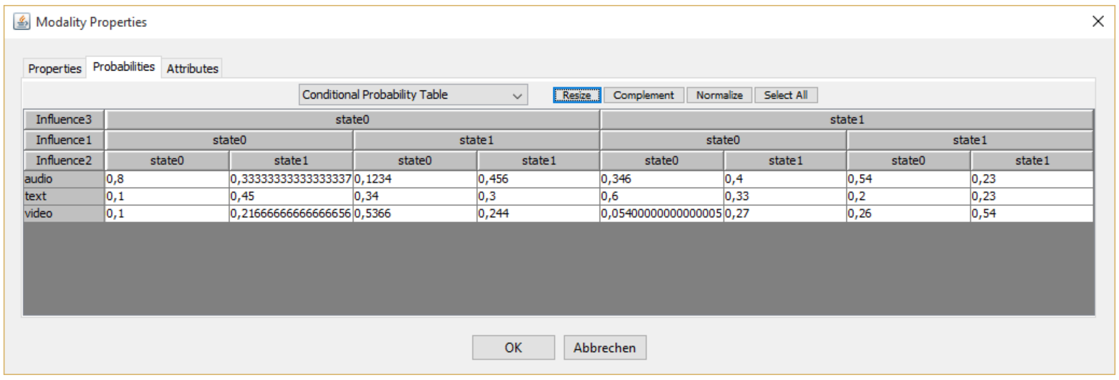


Figure 5.2: The initial modeling approach consisting of one node for device and one for modality. The known devices and modalities are featured as instances and assigned with equal probabilities.



| Influence3 | state0 | | | | state1 | | | |
|---|---|---|---|---|---|---|---|---|
| Influence1 | state0 | | state1 | | state0 | | state1 | |
| Influence2 | state0 | state1 | state0 | state1 | state0 | state1 | state0 | state1 |
| audio | 0,8 | 0,33333333333333337 | 0,1234 | 0,456 | 0,346 | 0,4 | 0,54 | 0,23 |
| text | 0,1 | 0,45 | 0,34 | 0,3 | 0,6 | 0,33 | 0,2 | 0,23 |
| video | 0,1 | 0,21666666666666656 | 0,5366 | 0,244 | 0,05400000000000005 | 0,27 | 0,26 | 0,54 |

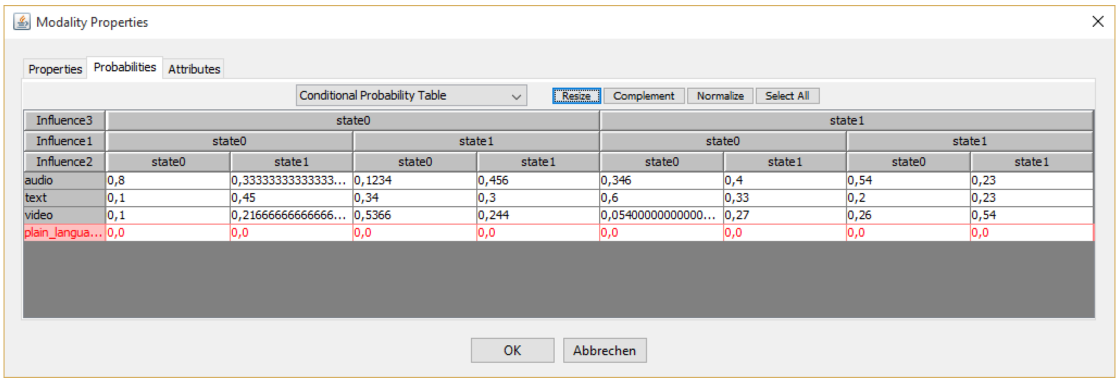
Figure 5.3: The Conditional Probability Table of the Modality node shown in Figure 5.2, including three binary influences on the modalities. 24 probability values have to be assigned and maintained in this setup.

- There is no possibility to exclude currently unavailable devices before inference. The information regarding device availability is part of the context information and should be included as evidence to narrow the results down during inference.

- The size of the CPT of such a node can grow very large. The number of entries of the CPT equals the Cartesian product of the number of instances of the node and the number of instances of all parent nodes. Both factors pose a problem. The number of instances can grow significantly when there are many devices and the parent nodes of a single device or modality node would include all the relevant context information. Additionally, a large table results in small probability values, which are difficult to maintain.

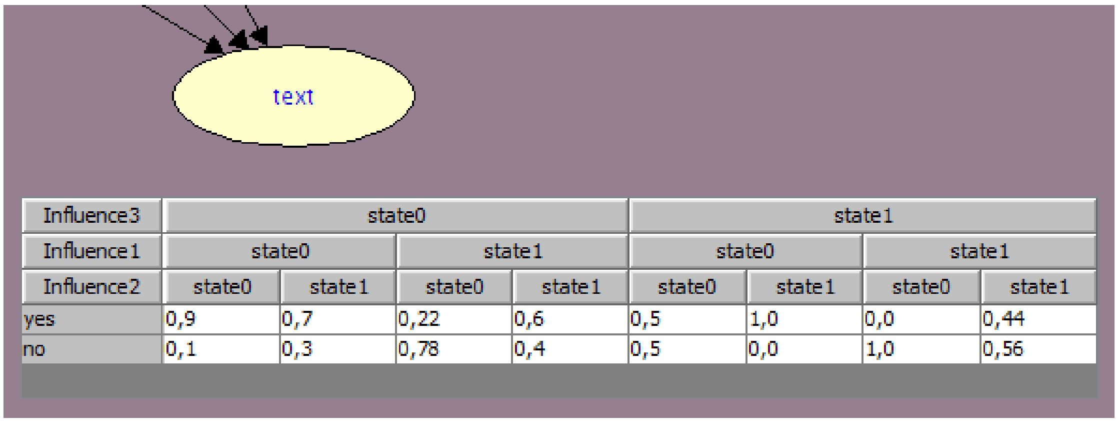
- Maintaining the integrity of the CPT is difficult. The sum of the probabilities for every column of the CPT has to equal 1. If a new device or modality is added as instance value, then every cell of the CPT has to be adapted to ensure sound probability values. This issue is evident in Figure 5.4, which shows a modality node after a new instance has been added. To assign a probability greater than 0 to the new modality, the probabilities of all other modalities have to be adjusted.

All of these issues could be addressed in the following iterations of the model.
First, the devices and modalities were extracted from the aforementioned device and modality node and individual nodes were created for each of them. That way every node would reflect the level of fitness of this device or modality given the context after inference. To reflect this, every device and modality node was assigned the instances *yes* and *no*. Figure 5.5 shows a node for the modality *text* and its CPT.
Using this technique the addition of a new modality does not entail the need for any alterations in the existing modality nodes. The restriction to the two instance values *yes* and *no* automatically results in larger, more expressive probability values, compared to the initial approach.
Using the instances *yes* and *no* to reflect the level of suitability for the modeled concept given



| Influence3 | state0 | | | | state1 | | | |
|---|---|---|---|---|---|---|---|---|
| Influence1 | state0 | | state1 | | state0 | | state1 | |
| Influence2 | state0 | state1 | state0 | state1 | state0 | state1 | state0 | state1 |
| audio | 0,8 | 0,33333333333333... | 0,1234 | 0,456 | 0,346 | 0,4 | 0,54 | 0,23 |
| text | 0,1 | 0,45 | 0,34 | 0,3 | 0,6 | 0,33 | 0,2 | 0,23 |
| video | 0,1 | 0,21666666666666... | 0,5366 | 0,244 | 0,05400000000000... | 0,27 | 0,26 | 0,54 |
| plain_langua... | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |

Figure 5.4: The Conditional Probability Table of the modality node after the addition of a new instance. The values of every column have to be adjusted.



| Influence3 | state0 | | | | state1 | | | |
|---|---|---|---|---|---|---|---|---|
| Influence1 | state0 | | state1 | | state0 | | state1 | |
| Influence2 | state0 | state1 | state0 | state1 | state0 | state1 | state0 | state1 |
| yes | 0,9 | 0,7 | 0,22 | 0,6 | 0,5 | 1,0 | 0,0 | 0,44 |
| no | 0,1 | 0,3 | 0,78 | 0,4 | 0,5 | 0,0 | 1,0 | 0,56 |

Figure 5.5: A node for the modality text and its Conditional Probability Table. The modality has two instances, yes and no and three exemplary binary influences, resulting in 18 probability values.

the context was used throughout the model.

## 5.2 Modeling Concepts

### Devices

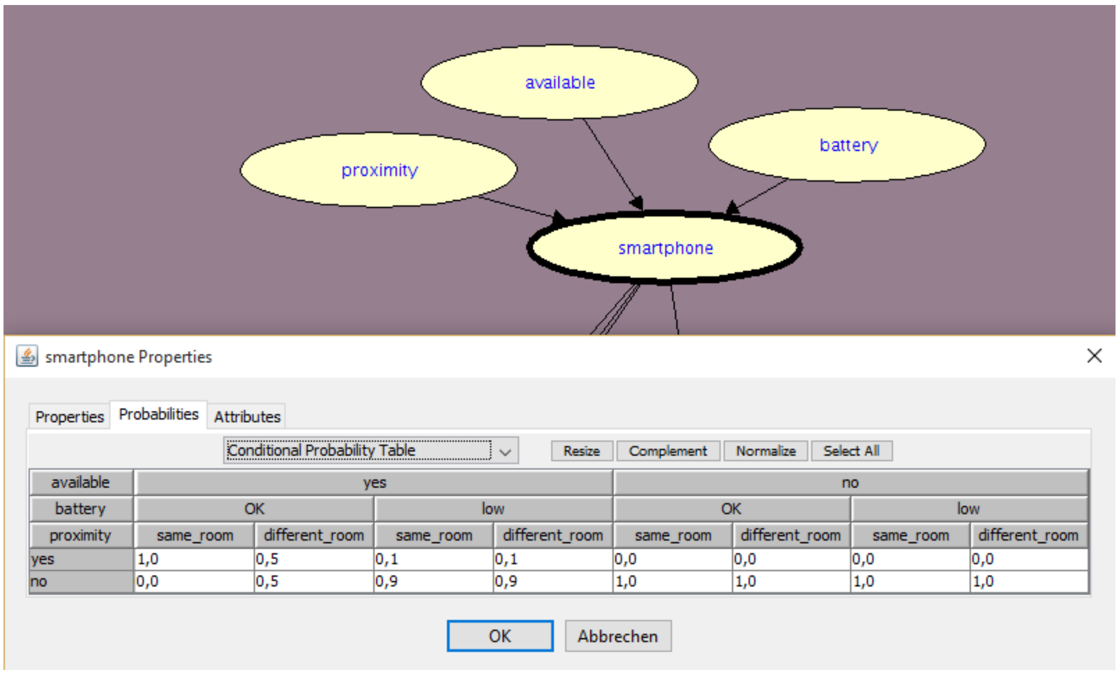
The set of nodes describing the devices will be referred to as device nodes. Semantically, the probability values assigned to these nodes reflect a measure of agreement that this device would be a good choice in the given context. Each device may have several properties influencing this assessment.

**Device Properties**

In the AALuis system, all kinds of devices can in theory be used and one aim of the Automatic IO Selection is to adapt to new devices. Some of the devices previously available, may be turned off and therefore be unavailable in the present, e.g. a smartphone that has run out of battery. When an evaluation is started, the information about the available devices is known by the AALuis system and is used as evidence for the probability query. To capture the devices' availability, a node was added for each device. The node *device_available*, referred to as availability node, has two possible values, *yes* and *no*. It acts as a switch to make sure that only available devices are included in the result.
If a device is unavailable, evidence in the corresponding availability node is set to *no*. The probabilities assigned to the device node make sure that $P(Device = yes|Available = no) = 0$. Thus, all child nodes of the device are updated with this probability during inference, excluding the device from the list of possible result combinations.
However, most devices have more relevant properties than their availability. Some devices are battery operated and are not a good option if the battery status is low. An interrupted interaction, caused by the selected device running out of battery should be avoided. The proximity of a device to the user is another property that was included for every device in the default model. In general, even if the user is fit to move around the house, it is assumed that it is more comfortable if an interaction is initiated by the AALuis system in the same room, than on a device that is currently not nearby. Other properties like resolution or screen size are also of interest and can be modeled as influences on the device node.
All device properties are modeled by nodes acting as parents to the device nodes. The resulting CPT can reflect which criteria rule out a device completely, like the availability, and which just lower their suitability, like a low battery status.
Figure 5.6 shows the nodes for the device *smartphone* with its properties. The CPT depicts how evidence that the device is not available results in fixing the probability for the instance value *no* to 100%. When the device is available, but has a low battery, it is rated as a poor choice regardless of the proximity to the device. In the case of full batteries, the user's proximity to the device is considered, but in this example both options were rated the same (50%).
In the same way, other context information influencing the fitness of the specific device may be modeled as a parent of that device.

## Modalities

In an analogous manner, each known modality is represented by a node with the instance values *yes* or *no*.
They are part of the result and therefore typically not observed. However, it is possible to exclude certain modalities from the results in general by setting the evidence in the corresponding modality node to *no*. Every piece of modeled context information that has an influence on the suitability of this modality, is connected to serve as a parent to the modality node.

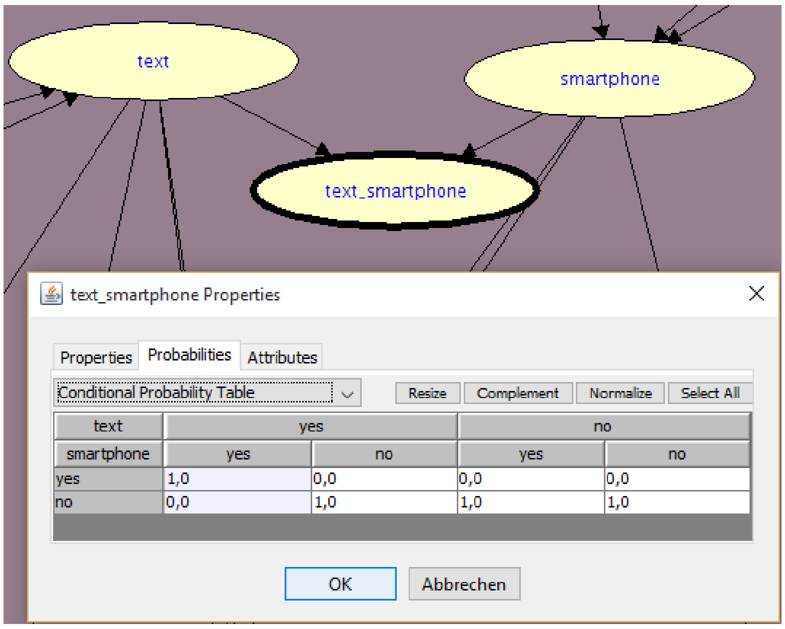


| available | yes | | | | no | | | |
|---|---|---|---|---|---|---|---|---|
| battery | OK | | low | | OK | | low | |
| proximity | same_room | different_room | same_room | different_room | same_room | different_room | same_room | different_room |
| yes | 1,0 | 0,5 | 0,1 | 0,1 | 0,0 | 0,0 | 0,0 | 0,0 |
| no | 0,0 | 0,5 | 0,9 | 0,9 | 1,0 | 1,0 | 1,0 | 1,0 |

Figure 5.6: The smartphone node with its properties battery status, availability and proximity. The Conditional Probability Table shows how the device is excluded from the results when it is unavailabile by setting the instance value *no* to 100% when there is evidence that it is not available.

### Combining Device and Modality

Modeling devices and modalities as individual nodes with the instance values *yes* and *no* reflecting their suitability, solved many of the issues regarding maintenance of the model. What remains is to model which device supports which modality.
Apart from the fact that a device has to support a modality to result in a valid combination, it is not clear if the selection of either device or modality should take precedence over the other. By selecting the best rated modality first and then choosing the most suitable device supporting it, combinations with other modalities are ruled out entirely. The same applies when selecting the best device first. To solve this predicament, while at the same time modeling which device supports which modality, an individual node was included for every possible combination of device and modality. These so called result nodes are child nodes of their respective device and modality in the Bayesian Network. Their CPTs exclude those combinations where either the device or the modality was ruled out.
Figure 5.7 shows the combination of the modality *text* and the device *smartphone*. When either the device or the modality evaluate to *no*, the result node also evaluates to *no*. Weighing the other case with 1 for *yes* for every result node means that no combination is penalized over any other. Shifting the probabilities in this column in favor of the instance *no* can be used to reflect preferences, downgrading individual combinations compared to others.



| text | yes | | no | |
|---|---|---|---|---|
| smartphone | yes | no | yes | no |
| yes | 1,0 | 0,0 | 0,0 | 0,0 |
| no | 0,0 | 1,0 | 1,0 | 1,0 |

Figure 5.7: The node representing the combination of text output on a smartphone. The Conditional Probability Table represents a switch, excluding the combination when one of the parents evaluated to *no*.

## Context Information

Context information exerts influence on the Automatic IO Device and Modality Selection. The user model provides information about the physical and cognitive abilities of the user. For the creation of this model, the user model was created on the basis of a myUI[1] user profile. The myUI schema lists a number of capabilities, which can be rated on a scale of 0 (normal) to 4 (severely impaired). The most significant entries were selected as a starting point for the creation of this default model: language reception, field of vision, visual acuity and sensitivity, and hearing.
Each entry of the user profile is represented by a node in the model. The values stored in the user model serve as evidence for the corresponding node. Again the size of the CPTs has to be considered. The five increments provided by the myUI user profile were reduced to only three values, namely *good*, *impaired* and *severely impaired*. Figure 5.8 shows the four nodes representing the user's ability that were included in the default model.
To illustrate the inclusion of context information besides the user context, two factors which influence the suitability of devices or modalities were selected: Surrounding noise and ambient light. Again, each piece of context information is modeled by a node, with instances specifying a range of possible values. To keep the model easy to maintain, only two to three instance values

[1]http://www.myui.eu/

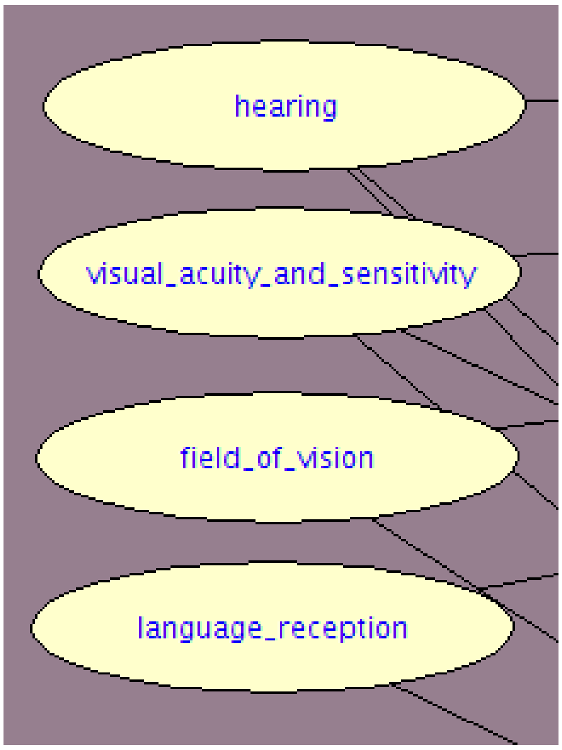
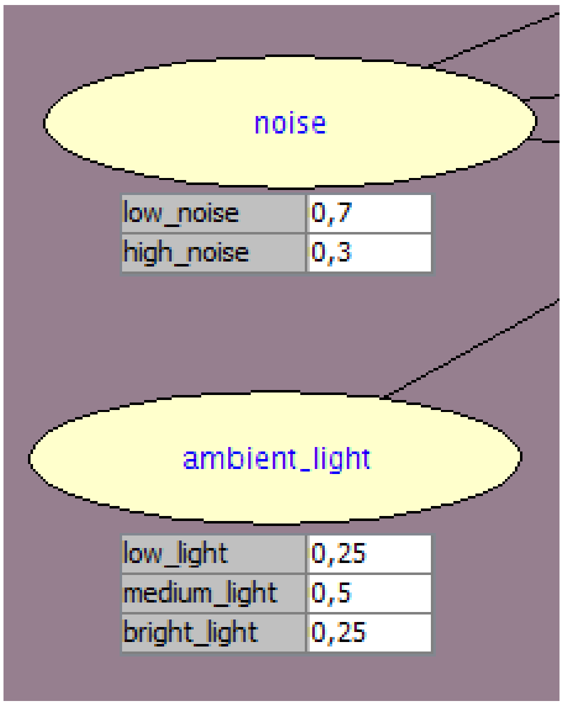


Figure 5.8: User Profile Nodes - Default Output Network

were used for each entity. Figure 5.9 shows the nodes ambient light and noise, together with their instances.

### Modeling Influences

After the purpose of the different nodes in the model has been discussed, the creation of edges between them has to be considered. The introduction on Bayesian Networks in Chapter 4 explained how an edge symbolizes a causal relationship between the connected nodes. All nodes that serve to assert evidence typically do not have any parents. That includes the context nodes and the device properties. They in turn serve as parents for every node that they have an influence on.
The central presumption in the context of this model is that every device and modality is equally suited for any task. The modeled context information has the potential to decrease their suitability. So, for every device and modality, all the negative influences are identified and connected as parents. Figure 5.10 shows the modality *audio* with its two negatively influencing factors *noise* and *hearing*.



| | |
|---|---|
| low_noise | 0,7 |
| high_noise | 0,3 |

| | |
|---|---|
| low_light | 0,25 |
| medium_light | 0,5 |
| bright_light | 0,25 |

Figure 5.9: The two modeled sensors and their instance values.

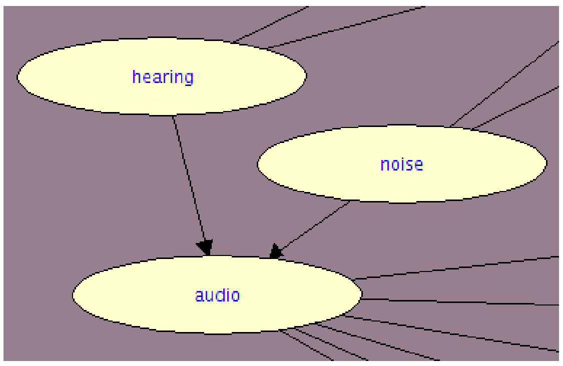
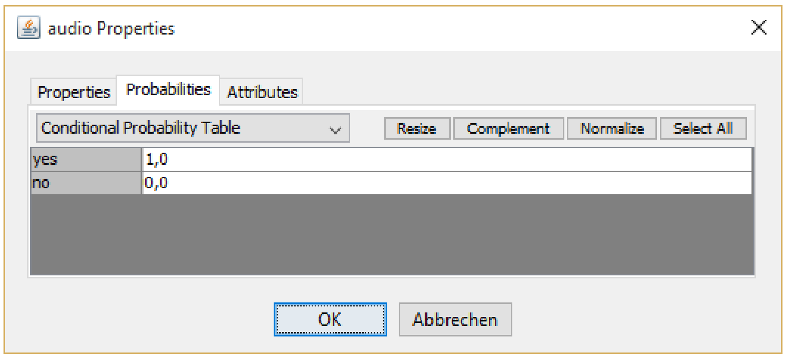


Figure 5.10: The modality audio with its two influencing factors noise and hearing.



| | |
|---|---|
| yes | 1,0 |
| no | 0,0 |

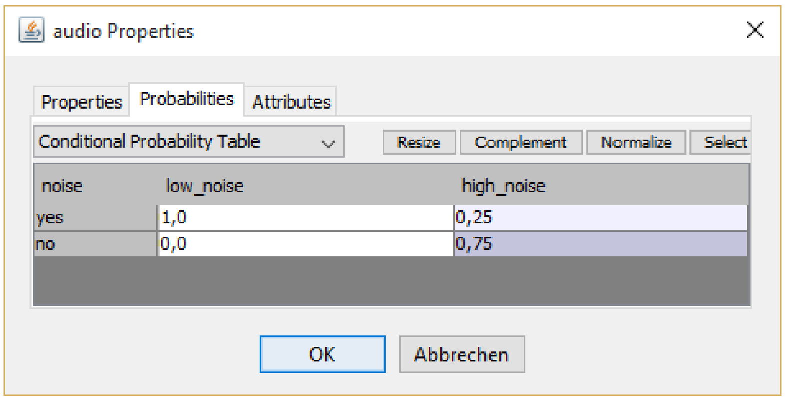
Figure 5.11: The Conditional Probability Table of the modality audio with no influences.

### Assigning Values to Conditional Probability Tables

For each node, the CPT has to be populated with probabilities. These values represent the probabilities for the outcomes of each node, given the state of its parents. Some nodes represent a concept that can be observed in the real world. For those nodes, the physical probabilities could be extracted from literature. The values for the user profile entries, reflecting the probability that the user of an AAL system has normal or impaired eyesight or hearing abilities are an example. For other nodes the probabilities cannot be observed and have to be approximated from experience or estimation.
For nodes that are generally observed, the assigned values are not very significant. When evidence is set, the values entered into the CPT do not factor in the inference at all. They only have an influence if the evidence for some reason cannot be set, for example if sensor data is unavailable. The probabilities most influential on the inference result concern the modality and device nodes.

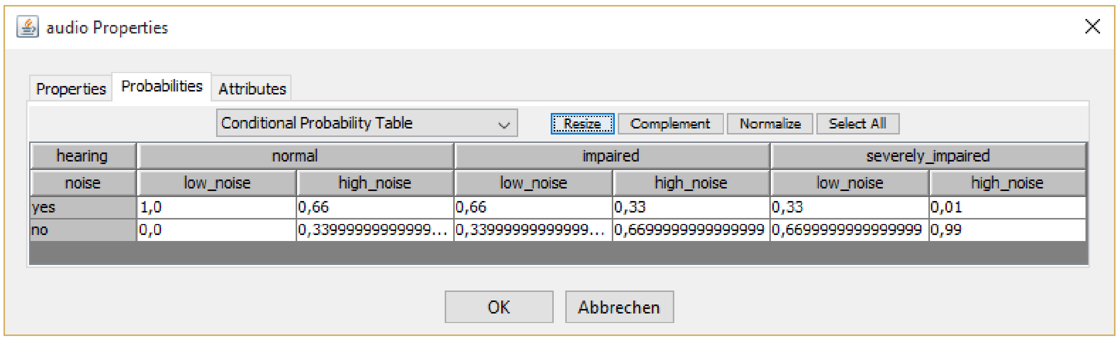
#### Assessing Multiple Negative Influences

The assignment of the probability values for devices and modalities is a crucial element of the modeling process. In this section, an example is gradually extended to describe the task. The simplest case is a modality node that has no influences. Such a CPT is shown in Figure 5.11. If it has no parents in the model, and cannot be observed, then the assigned probabilities of a node would be incorporated into the result unchanged.
Adding one node as a parent, identifying the modeled concept as an influential factor on the modality, is demonstrated in Figure 5.12. For every state the parent node can take on, the probabilities for the modality have to be adjusted. In most cases, one of the instances of the parent represent a neutral situation. When that state is observed, the modality is still as good a choice, and its *yes* instance receives a probability of 100%. In Figure 5.12 this is the instance *low noise*. The presence of the other instance of the parent, *high noise* has a negative influence on the suitability of the modality, so the *yes* instance is rated lower in that case.
The situation becomes more complex with the addition of another influence. One example of a CPT including a second influencing factors is shown in Figure 5.13. Again, it is possible to



| noise | low_noise | high_noise |
|---|---|---|
| yes | 1,0 | 0,25 |
| no | 0,0 | 0,75 |

Figure 5.12: The Conditional Probability Table of the modality audio with one influencing factor.



| hearing | normal | | impaired | | severely_impaired | |
|---|---|---|---|---|---|---|
| noise | low_noise | high_noise | low_noise | high_noise | low_noise | high_noise |
| yes | 1,0 | 0,66 | 0,66 | 0,33 | 0,33 | 0,01 |
| no | 0,0 | 0,33999999999999... | 0,33999999999999... | 0,6699999999999999 | 0,6699999999999999 | 0,99 |

Figure 5.13: The Conditional Probability Table of the modality audio with two influencing factors.

identify one state that is neutral where the probability that the modality is a good option is set to 100%. One of the states is clearly the worst, when both influencing factors evaluate to their most severe instance. In Figure 5.13 the modality was assigned with only 1% in that case for the *yes* instance. Entering the probabilities for the cases in between is difficult. A state that is objectively worse than another one should receive a lower probability. In the example of two influences, this situation can be observed when the instance of one influence stays the same, while the other one is in a state that is disadvantageous to the modality. In Figure 5.13, high noise reduces the fitness of audio output compared to low noise in every instance where the hearing abilities stay the same. At the same time it was assumed that having normal hearing capabilities during high noise situations equals having impaired hearing when there is low noise.
Deciding whether a worse situation in one influence, combined with a more advantageous state of the other influence should result in a comparable probability for the modality, is sometimes not possible. The influence of different factors cannot easily be weighed against each other.
In the creation of this model, an attempt has been made to rate such ambiguous situations homogeneously, as shown in Figure 5.13. In a real application the behavior has to be evaluated to test the soundness of this approach and refined gradually.

## 5.3 Automatic IO Selection - Using the Model for Inference

Observed context information is set as evidence in the corresponding nodes during inference. Where no evidence is set, the entered probability values are used for the evaluation. Then, the joint probability is calculated using the evidence. Afterwards the result nodes, representing the combinations of devices and modalities, can be queried for their updated probability values. The result combinations can be ordered by the probability for *yes* to obtain a ranking of the best options. This way it is also possible to present the user with, for example, the top three combinations instead of a single best option.

## 5.4 Expanding the Model

Since there are constantly new devices, new models and new types of devices being developed, the extensibility of the Bayesian Network was an important factor while designing it. To facilitate the addition of devices and modalities with the help of software, both processes could be described formally.

### Adding a Device

The process of adding a device is shown in Figure 5.14. The first step is to make sure that it does not already exist. If it is in fact new, then a node for this device is created, identified by the name of the device and including two instances *yes* and *no*. Along with that node, every device is supplemented by one node to reflect its availability. An edge has to be created from the availability node to the device node. The proximity information is included for every device as well. For the proximity node the instances *same room* and *different room* are assumed. Again an edge from the proximity node to the device node complete this part.
Subsequently if the device is battery operated, another node with the instances *OK* and *low* is added and connected with an outgoing edge to the device. All the device related nodes have to be labeled in accordance to a certain pattern: first the *device name*, followed by an underscore (_), followed by *availability*, *proximity* and *battery_status*.
Finally, the CPT has to be populated with probabilities. In columns where the availability node is presented with the instance no, the probability for the device has to be set to 100% for the instance value *no*. The probabilities for the other cases can be assigned at one's own discretion, but it should be noted, that assigning 100% to *no* in a case, leads to the complete exclusion of the device in the given situation.
Figure 5.15 shows an example. The four columns shown on the right can be disregarded completely, because the device is not available in the described cases. A full battery and the device being in the same room as the user, lead to no penalty for the device, so *yes* is assigned 100%. A low battery is considered an undesirable state, so the probability for the device is set to low values in those cases, while being in a different room with a full battery still results in 66% for the device.
After the CPT has been adjusted, all the modalities that this device supports have to be identified and a new result node for every combination included. The result nodes have to be labeled

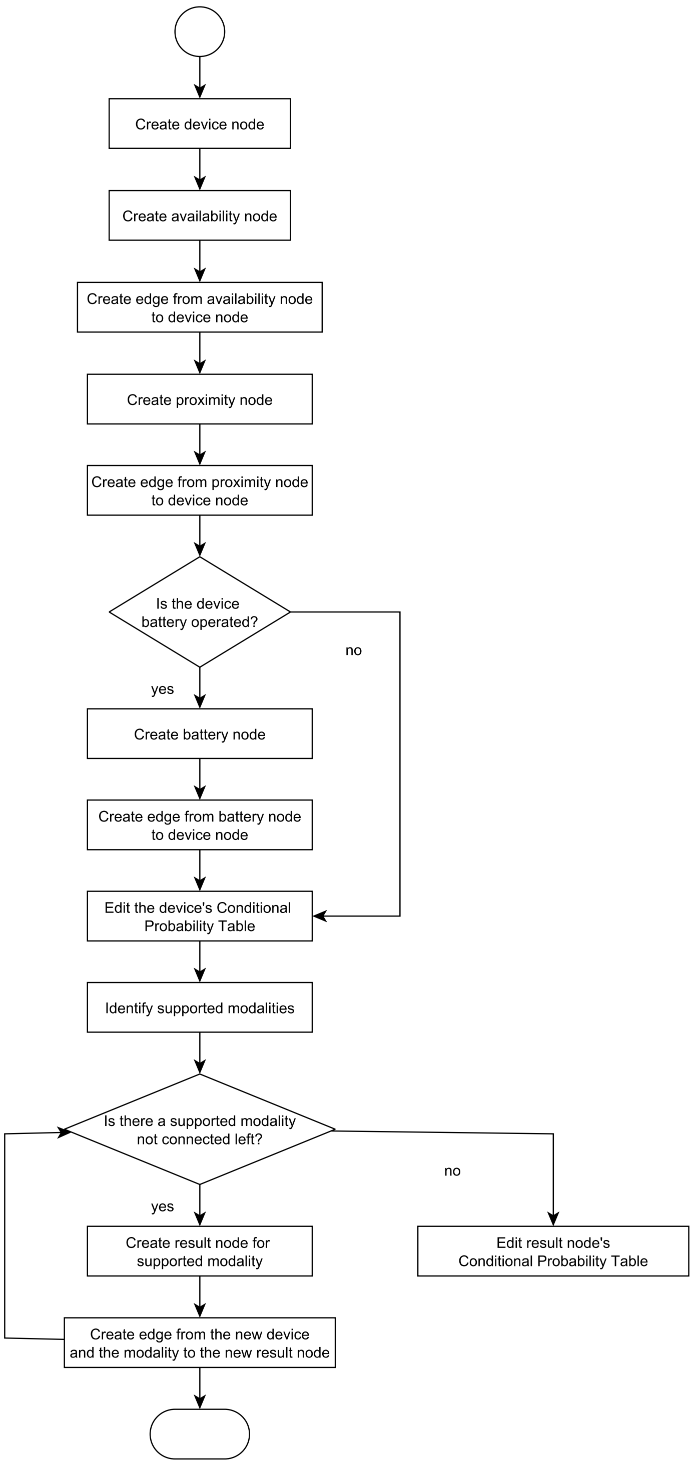
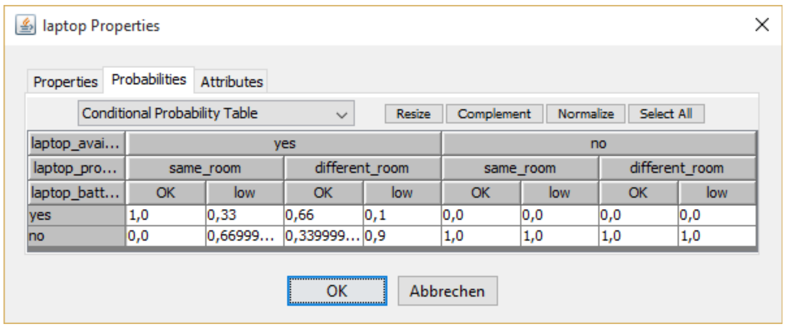

Create device node
Create availability node
Create edge from availability node to device node
Create proximity node
Create edge from proximity node to device node
Is the device battery operated?
no
yes
Create battery node
Create edge from battery node to device node
Edit the device's Conditional Probability Table
Identify supported modalities
Is there a supported modality not connected left?
no
yes
Create result node for supported modality
Edit result node's Conditional Probability Table
Create edge from the new device and the modality to the new result node

Figure 5.14: The process of adding a new device to the model



| laptop_avai... | yes | | | | no | | | |
|---|---|---|---|---|---|---|---|---|
| laptop_pro... | same_room | | different_room | | same_room | | different_room | |
| laptop_batt... | OK | low | OK | low | OK | low | OK | low |
| yes | 1,0 | 0,33 | 0,66 | 0,1 | 0,0 | 0,0 | 0,0 | 0,0 |
| no | 0,0 | 0,66999... | 0,339999... | 0,9 | 1,0 | 1,0 | 1,0 | 1,0 |

Figure 5.15: The Conditional Probability Table of a device with three parents.

according to the following pattern: the name of the modality, followed by an underscore (_) followed by the name of the device. The result node is connected by incoming edges to the device and the modality. The CPT of the result node has to match the one shown in Figure 5.7.

## Adding a Modality

The process of adding a modality is shown in the flow chart in Figure 5.16 First, a node for the new modality is created. It is assigned with the two instance values *yes* and *no*.
Next, all nodes representing context information having a negative influence on the fitness of this modality are identified. Every influence is connected to the modality node by an edge, becoming the nodes parents.
When all influences are connected, the CPT can be filled in. Just as described for the addition of new devices, each new device and modality combination is identified and a result node for each added to the model.

## Adding Context Information

When new context information representing a relevant influence on the Output Device and Modality Selection becomes available, it can be added as a node to the model. All existing modalities and devices should be reviewed to find if the addition introduced any new causal relationships. If so, an edge is created from the context node to the modality or device. Subsequently the probabilities of the affected CPT have to be adjusted. Figure **??** shows a flow chart describing this process.

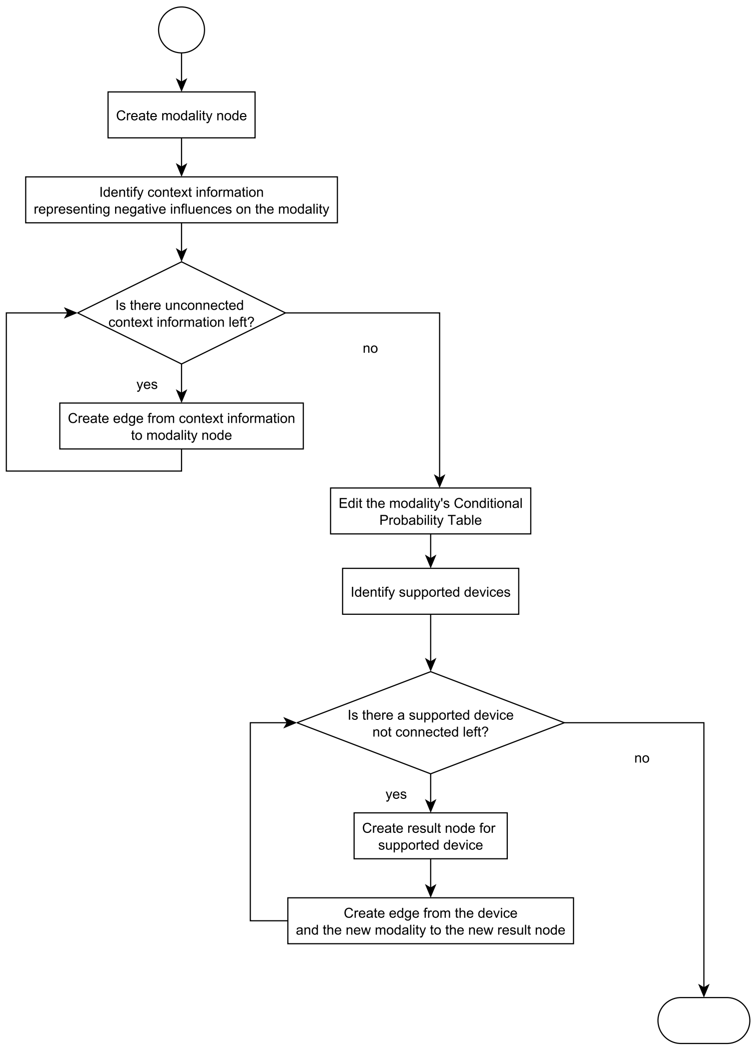

Create modality node
Identify context information representing negative influences on the modality
Is there unconnected context information left?
no
yes
Create edge from context information to modality node
Edit the modality's Conditional Probability Table
Identify supported devices
Is there a supported device not connected left?
no
yes
Create result node for supported device
Create edge from the device and the new modality to the new result node

Figure 5.16: The process of adding a modality to the model

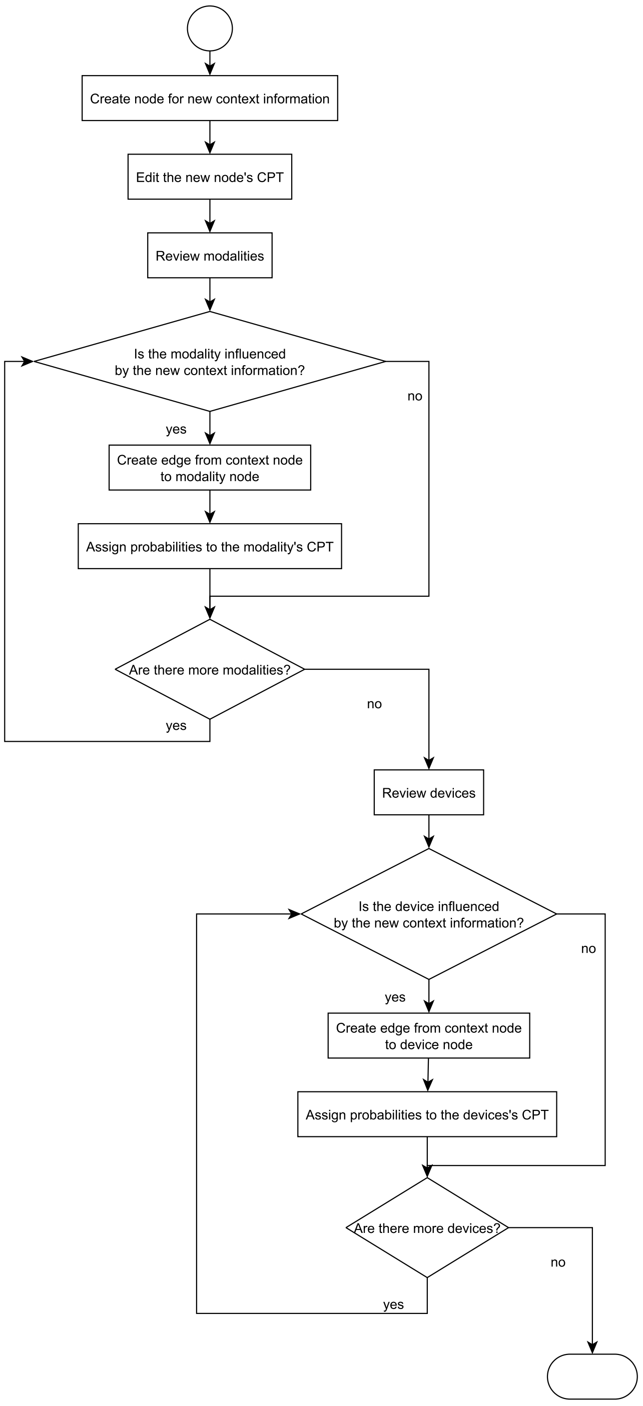

Create node for new context information
Edit the new node's CPT
Review modalities
Is the modality influenced by the new context information?
no
yes
Create edge from context node to modality node
Assign probabilities to the modality's CPT
Are there more modalities?
no
yes
Review devices
Is the device influenced by the new context information?
no
yes
Create edge from context node to device node
Assign probabilities to the devices's CPT
Are there more devices?
no
yes

Figure 5.17: The process of adding new context information to the model

CHAPTER 6

# Implementation

On the basis of the model described in the previous chapter, the inference mechanism and editing of the model according to the given guidelines were implemented in Java programs. The aim was to prepare the integration of the selection mechanism into AALuis and ensure that the inference mechanism works satisfactory outside of the modeling tool. A second part of the implementation explored the possibility to offer functions for the extension of the model to a trained user or administrator while maintaining the validity of the model. Using the SamIam Java library, the result is one class named `OutputNetworkEvaluator`, providing the possibility to load a Bayesian Network, set inference according to the context information and obtain the best device and modality selection after inference. An associated application called `OutputNetworkEditor`, offers a GUI to add devices and modalities to such a model. These software components will be described in this chapter.
Developing an application that is intended to be a part of an AAL framework involves making a few assumptions. Development was done separately from the AALuis framework. The integration of the created software is beyond the scope of this work. Thus, no user or context model were available during development. In order to be able to create this model and application, these sources were therefore replaced by files and user input. The user context was read directly from a user profile present in a file. Additional contextual influences had to be entered manually for testing purposes.

## 6.1 Input and Output Files

The developed application requires access to input files and uses files to save information.

### User Profile

An XML file containing the user profile in the form of key-value pairs is used to provide the user context. It was included to provide input data to test the selection mechanism. In a working AALuis environment this information would be queried from the user model.

At startup, the user profile file is accessed, the XML structure parsed and the values read are transformed to match the instances of the user profile nodes of the Bayesian Network.

### The Network File

The Bayesian Network used in the created applications was modeled using the GUI of SamIam. The model was saved in the Hugin File Format with the file extension .net (Network File). This file is accessed by the application to read the model for inference and for editing.

### Model Descriptor File

In the previous chapter, the nodes that are part of the model, were grouped into the categories device, modality, context and result node. This distinction served well to explain the modeling process. It is also relevant for the process of inference. Using the graphical tool SamIam, a user can see all existing nodes and their identifiers. To a human it is clear which nodes are intended to reflect context information, observable in the real world and which contain the result, the device and modality combinations. A computer program, however, is not able to distinguish the nodes by the meaning of their labels. Therefore, the software has to be provided with information about the semantics of the individual nodes.
To facilitate this, an additional file type was created. The so called Network Descriptor File is an XML file that arranges the names of the nodes in a structure, encoding their semantic meaning. It holds the path to the network file containing the described Bayesian Network and groups the node names into the categories device, modality, sensor, user profile and result.
It is used as input file for the evaluation and editing part of the software.

### Preferences File

Both application parts store the paths of the last used network and user profile to a preference file. The `FilePreference` implementation by David Croft[1] was used to read and write it.

## 6.2 The SamIam Java Library

The SamIam Java library provides interfaces for the modeling and inference mechanisms of SamIam to other applications. The class `HuginNet` encapsulates the Bayesian Network in an object. All operations on the model like adding nodes and edges can be performed using this object. Nodes of the model are represented by objects of the class `HuginNode`. The class offers methods to access the properties of a node, including its identifier and CPT. For inference an object of `InferenceEngine`, created with the help of the class `HuginEngineGenerator` was utilized.

[1]`http://www.davidc.net`

## 6.3 The Class OutputNetwork

The class `OutputNetwork` is the central element of both parts of the application. It holds the Bayesian Network in an object of `HuginNet`, as well as the meta information read from the network descriptor file, necessary for handling the model automatically. The meta information about the nodes is read from the XML file and the node identifiers are stored in the corresponding fields of the type `Set<String>`, called `modalityNodes`, `profileNodes`, `sensorNodes`, `deviceNodes` and `resultNodes`. Apart from the getters and setters for those fields, the methods of this class provide information about nodes and instances for the `OutputNetworkEvaluator` and `OutputNetworkEditor`.

## 6.4 OutputNetworkEvaluator

The OutputNetworkEvaluator class operates on an `OutputNetwork` object to perform inference using the Hugin variation of the join tree algorithm. To that end, it facilitates the reception and processing of evidence, performing a probability query and returning the result.
An object of `OutputNetworkEvaluator` is instantiated with an `OutputNetwork` in the class constructor. Also, the context information pertaining the user is collected right away and stored until it is required as evidence for a query. The program flow of the evaluation is depicted in Figure 6.1. After initiation, the collection of evidence and performing of inference queries can be repeated.

### Collecting Evidence

Information from the context or user model, modeled in the Bayesian Network is used as evidence for inference. The method `observe` can be called with a node name that corresponds to a piece of context information and a value. If such a node and instance exist, both Strings are added to the HashMap `observedNodesAndInstances`. If the node has been observed before, the value is updated. If the given node name represents a device, the private method `setDeviceStatus` is called.
Instead of adding the name of the device to `observedNodesAndInstances`, it is added to the HashSet `availableDevices`, if the given instance was *yes* and removed from it if the instance equals *no*. The distinction is made, because the availability of at least one device is a prerequisite for successful inference. Storing the available devices separately makes it easier to verify this when a probability query is initiated.
The method `unobserve` facilitates the removal of a piece of context information from the `observedNodesAndInstances` if it becomes unavailable in the context model.

### The Probability Query

A probability query can be initiated at all times after the `OutputNetwork` was initialized in the constructor. When the method `doProbabilityQuery` is called, first a check is performed to see if the list of available devices is empty. If that is true, no sensible result can be calculated, and an `IllegalArgumentException` is raised with the message “No device is available.”.

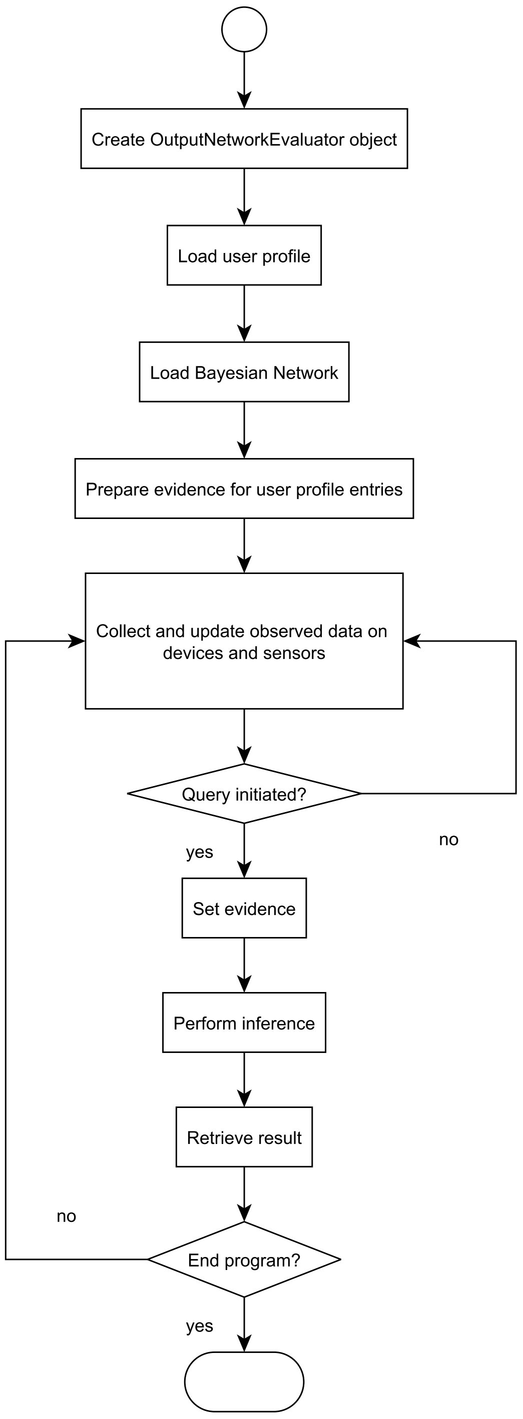

Create OutputNetworkEvaluator object
Load user profile
Load Bayesian Network
Prepare evidence for user profile entries
Collect and update observed data on devices and sensors
Query initiated?
yes
no
Set evidence
Perform inference
Retrieve result
no
End program?
yes

Figure 6.1: The Process of the OutputNetwork Evaluation

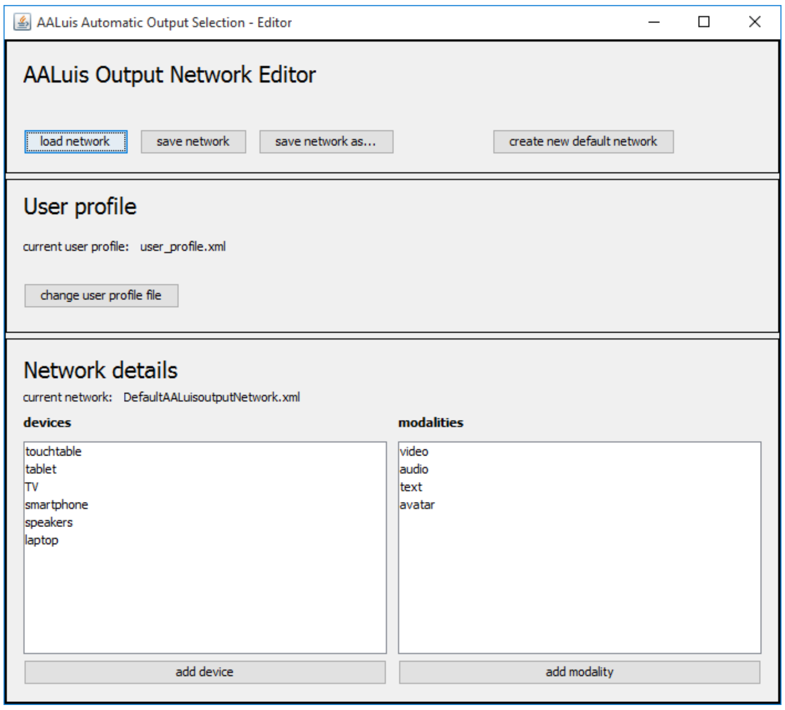
If at least one device is present, the probability query can be performed. The `HuginNet` object contained in the `OutputNetwork` has to be initialized with a new `EvidenceController`, a part of the SamIam inflib.jar library. The fields which may hold results from previous probability queries and the evidence are cleared. Then, the current status of the observations collected since the last run are assigned as new evidence in the method `setCollectedEvidence`.
First, the method `setEvidenceForDevices` iterates through the list of device names which were observed. For each one, the suffix "_available" is appended to find the node representing the availability of the device. Those nodes are included into evidence with the instance *yes* using the method `setEvidence`. To make sure that no device that was not observed to be available factors into the evaluation result inadvertently, the availability of all those devices not in the list of observed devices is explicitly set to *no*.
After this, method `setCollectedEvidence` iterates over all elements of the HashMap `observedNodesAndInstances` and calls `setEvidence` for each entry. `setEvidence` uses the HashMap `evidence` to store the instance values of the nodes in question as objects of `FiniteVariable`, together with the respective instance.
After the evidence HashMap is fully populated, an `InferenceEngine` is prepared, by specifying the parameters necessary for the Hugin inference algorithm. Then, the contents of the HashMap `evidence` are assigned to the model's evidence controller.
The method `calculateMarginalsForSetOfVariables` is called with the Set of result nodes and the created inference engine. This specifies the result nodes as the target for the evaluation result. The method performs the marginalization of the CPTs of every result node on the instance value *yes* and returns a HashMap containing the result. The result HashMap, containing the name of the result node and the probability calculated for *yes* is ordered by the probability values and the modality ranked best is assigned to the field `bestCombinationName`. The associated probability is assigned to the field `bestCombinationProbability`.
In general, the probability query always returns a result, the best device and modality combination available. However, there are cases when even the best result receives only a low probability value during inference. This could indicate a mistake in modeling, or that an error occurred, either during the collection of evidence or in the context model used. To alert the user that this could be the case, the `OutputNetworkEvaluator` uses a constant to check if the best result received a very low probability and prints a warning to the console. It was initially set to 20%, but should be adapted in real applications after an evaluation period. After a probability query has completed, the last state of collected observations remains in the HashMap observedNodesAndInstances and Set availableDevices. New observations can be added, removed or updated using the methods observe or unobserve.

## 6.5 OutputNetworkEditor

### Editing the OutputNetwork

The class `OutputNetworkEditor` offers methods to support manipulations on an Bayesian Network. Its only field is an Object of `OutputNetwork` which is initialized in the constructor.

Its methods support the processes for adding devices and modalities shown in Figures 5.16 and 5.14.
The main window of the program is shown in Figure 6.2. The button labeled *load network* opens a file chooser for the selection of a network descriptor file. All file operations are carried out by the class `OutputNetworkIO`. To load a network it parses the given XML file and returns an object of `OutputNetworkEditor`. The meta information stored in the contained `OutputNetwork` object is accessed and used to present the currently known devices and modalities to the user.



current user profile: user_profile.xml

current network: DefaultAALuisoutputNetwork.xml

devices: touchtable, tablet, TV, smartphone, speakers, laptop

modalities: video, audio, text, avatar

Figure 6.2: A default network loaded in the AALuis Output Network Editor

## Adding Devices

The steps necessary to add a device were already covered in Chapter 5. They include the addition of the device's node and the associated device properties battery status and device proximity. The methods of `OutputNetworkEditor` support this process and set the CPT of the device to a set of default probabilities. The Editor software does not support the editing of those probabilities to maintain the uniform behavior of devices in the model. If necessary they can be adapted in SamIam.
To create the default CPTs for new device nodes, there are two methods. One creates a default CPT for a battery operated device, the other one for a device without battery. Since the structure of the device nodes always equals one of those two options, the rows and columns of the CPT can be accessed directly. The SamIam Java library offers appropriate methods to address table

cells using the instance names of the parents and assign probability values to them.
In the Editor's GUI clicking the button *add device* opens a dialog window shown in Figure 6.3 that allows the user to enter all necessary information for a new device.

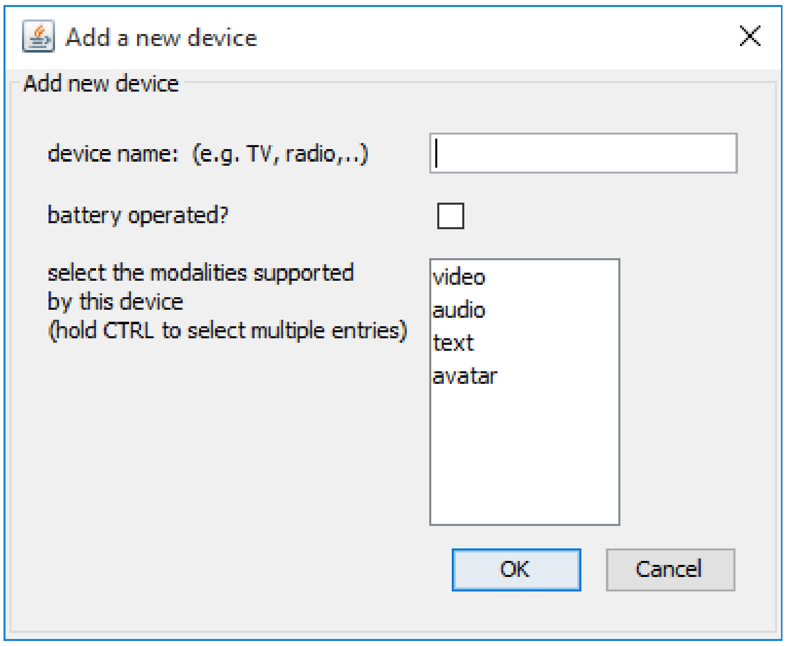
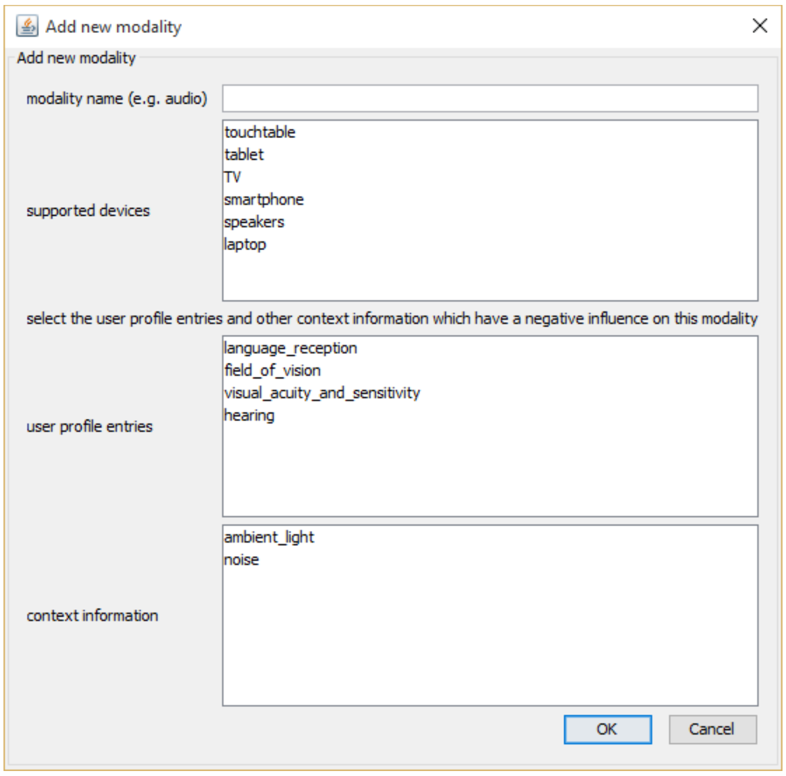

Add a new device
Add new device
device name: (e.g. TV, radio,..)
battery operated?
select the modalities supported by this device (hold CTRL to select multiple entries)
video
audio
text
avatar
OK
Cancel

Figure 6.3: Adding a device in AALuis Output Network Editor

## Adding Modalities

To add a new modality, it is necessary to add a node and identify all the context information influencing it. When the influences are connected with nodes, their effect has to be rated in the modality's CPT. The class `OutputNetworkEditor` supports the process with the method `addModality`.
The dialog that assists the user in adding a modality is shown in Figure 6.4. It groups the possible negative influences into user profile entries and other context information to keep the number of list items small. However, the influences of both categories are treated the same by the algorithm processing the entered information.
After a modality name has been specified and the supported devices and influences have been selected, all new nodes and edges have to be added to the model in order to populate the modality's CPT with probabilities. To that end, all the entered information is first collected in an object of the class `OutputNetworkModality`. The method `addModality` receives this object and creates the new modality node, the new result nodes and all the necessary edges. Then, the CPT is ready to be assigned with probabilities. Its size depends on the selected influences and the sequence of the represented states depends on the order in which the edges were created. The table's details are read from the node and used to create an interface to enable the user to adjust the probabilities with sliders.


Add new modality

Add new modality

modality name (e.g. audio)

supported devices: touchtable, tablet, TV, smartphone, speakers, laptop

select the user profile entries and other context information which have a negative influence on this modality

user profile entries: language_reception, field_of_vision, visual_acuity_and_sensitivity, hearing

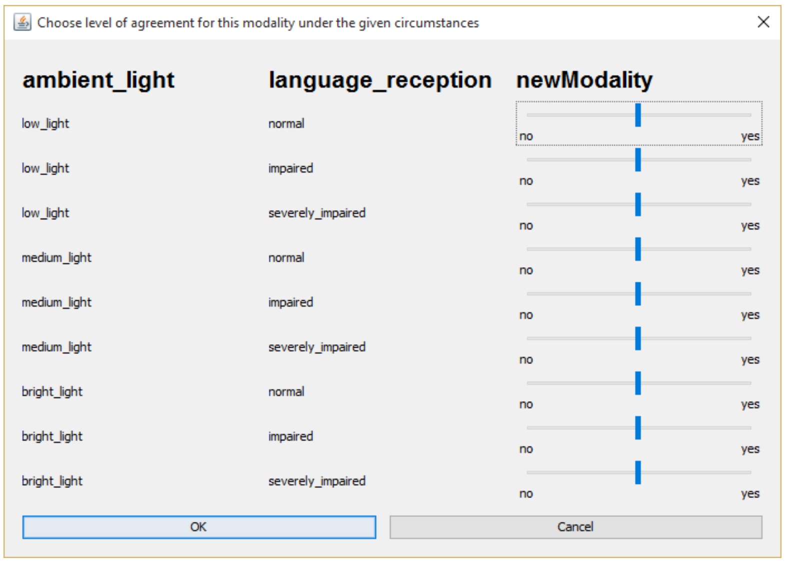
context information: ambient_light, noise

Figure 6.4: Adding a modality in AALuis Output Network Editor

### Editing the Conditional Probability Table

To dynamically create the interface to edit the CPT of a node, the variable is accessed. It returns a String representation of its table.
An example of this String for a modality with two parents is shown below:

```
ambient_light  language_reception  newModality  Value
low_light      normal              yes          0.5
low_light      normal              no           0.5
low_light      impaired            yes          0.5
low_light      impaired            no           0.5
low_light      severely_impaired   yes          0.5
low_light      severely_impaired   no           0.5
medium_light   normal              yes          0.5
medium_light   normal              no           0.5
medium_light   impaired            yes          0.5
medium_light   impaired            no           0.5
medium_light   severely_impaired   yes          0.5
medium_light   severely_impaired   no           0.5
bright_light   normal              yes          0.5
bright_light   normal              no           0.5
bright_light   impaired            yes          0.5
```


Choose level of agreement for this modality under the given circumstances

| ambient_light | language_reception | newModality |
|---|---|---|
| low_light | normal | no yes |
| low_light | impaired | no yes |
| low_light | severely_impaired | no yes |
| medium_light | normal | no yes |
| medium_light | impaired | no yes |
| medium_light | severely_impaired | no yes |
| bright_light | normal | no yes |
| bright_light | impaired | no yes |
| bright_light | severely_impaired | no yes |

Figure 6.5

```
bright_light   impaired            no          0.5
bright_light   severely_impaired   yes         0.5
bright_light   severely_impaired   no          0.5
```

It is split up to create sliders and the appropriate labels for the UI. Figure 6.5 shows an example of such an interface, created from the given String representation. The probabilities of the modality's instances yes and no are represented by a slider. This representation is convenient because together they have to equal 100%. One line in this dialog therefore represents two lines of the original table shown above. When the user confirms the selected values for the CPT using the OK button, an array of doubles is created to collect the values from the sliders. For every line, the value for the yes and no instances is derived from the slider's position and assigned to the array. This array is then passed to a method assigning the probabilities to the modality node. The assignment of all probabilities for a CPT at once using such an array of doubles is a different approach than the one used for new devices, where probabilities can be assigned to individual cells of the table.

## 6.6 Restoring a Default Network

The structure of the Bayesian Network is very important for the software to work correctly. It is a complex task to create a new one without a template. Therefore, a class `DefaultOutputNetwork` for generating the default network was included.
The method `getDefaultNetwork` can be called to retrieve an Object of `OutputNetwork` that includes the entities described in Chapter 5. The feature is embedded into the Editor software. A button loads the default network, which can then be adapted and saved accordingly.

CHAPTER 7

# Results

## 7.1 Scenarios

To evaluate the results of the proposed solution with regard to the outcome of the evaluations, three example scenarios will be presented here. The examples aim at illustrating how some context situations produce a distinct result very clearly while others are more ambiguous.
All three examples are based on the same sample household. It includes six different devices: TV, laptop, smartphone, touchtable, tablet and speakers. Together, those devices support the output modalities avatar, text, video and audio. The user is characterized by four attributes: visual acuity, hearing, field of vision, and language reception. Other modeled context information includes noise, ambient light, the proximity of a user to each device and their battery status.

### Scenario 1

The first scenario features a user that has good physical abilities, except for impaired hearing. Beyond that, the noise level in the house is high. It features a TV in a different room and a touchtable in the same room as the user. The situation and the results are depicted in Figure 7.1.
The combination of touchtable and text output is the best rated result in this case, receiving a score of 100%, followed by video output on the touchtable, receiving 78%.

### Scenario 2

The second scenario features a user that does not suffer from any physical restrictions. The measured noise level in the house is low, but the lighting conditions are unfavorable. This time, three devices are available. Again the TV is located in another room. Close to the user there are a laptop with good battery status and a smartphone with low battery. The situation is shown in Figure 7.2.
The best option in this scenario is the output of text on the laptop, receiving 100% agreement.

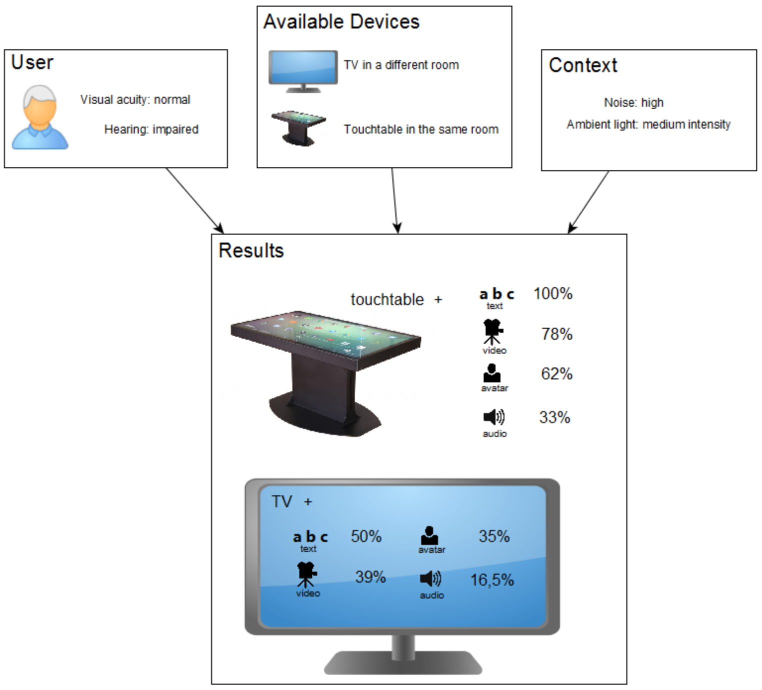


Figure 7.1: Scenario 1

The remaining modalities available for the laptop are all rated with 66%, while combinations including other devices are rated poorer overall.

### Scenario 3

Finally, the third scenario shows an overall disadvantageous situation. The user has severe physical limitations and the conditions in the house are unfavorable. The devices are either located in a different room or have a low battery status, except for the touchtable. Figure 7.3 shows this scenario and the calculated results in this case.
The results reflect the challenging situation. The touchtable stands out as the best device, with three modalities receiving a high score - video (34%), avatar (34%), and text (33%). The result for the other possible combinations receive scores ranging from 0.33% to 22.44%.

## 7.2 Runtime Analysis

A run-time analysis was performed for three different versions of the model. Table 7.1 shows the results. The mean runtime for the first execution is depicted separately from the numbers measured for subsequent runs of the selection. For subsequent executions, the minimum and maximum runtime is given in addition to the mean and the standard deviation. The first runs take, on average, from 212.6ms for a model containing 8 nodes, to 628.8ms for the large example of

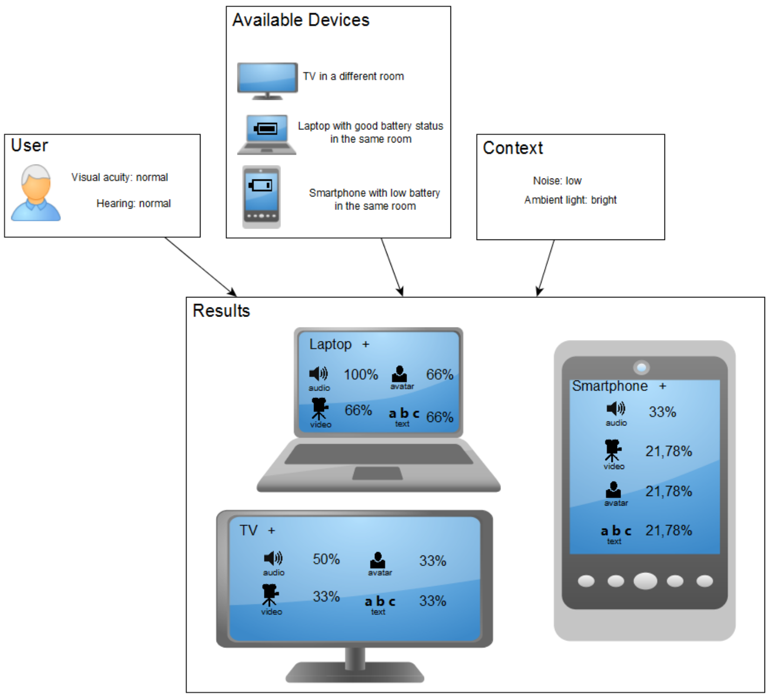


Figure 7.2: Scenario 2

| | First run | Subsequent runs | | | |
|---|---|---|---|---|---|
| Number of Nodes | Avg. Time [ms] | Avg. Time [ms] | Min. Time [ms] | Max. Time [ms] | SD [ms] |
| 8 | 212.6 | 2.206 | 2.043 | 2.741 | 0.198 |
| 52 | 362 | 19.033 | 17.679 | 23.234 | 1.243 |
| 208 | 628.8 | 59.934 | 56.875 | 66.295 | 2.617 |

Table 7.1: The results of the runtime analysis for the device and modality selection on Bayesian Networks of three different sizes

208 nodes. The execution of subsequent queries provides different results regarding the runtime. The 8-node example, on average, produces a result in 2.206ms, with a standard deviation of 0.198ms. Execution on a medium-sized example (52 nodes) takes on average 19.033ms with a standard deviation of 1.243ms. The analysis of the largest model returns an average execution time for subsequent runs of 59.934ms, with a standard deviation of 2.617ms.

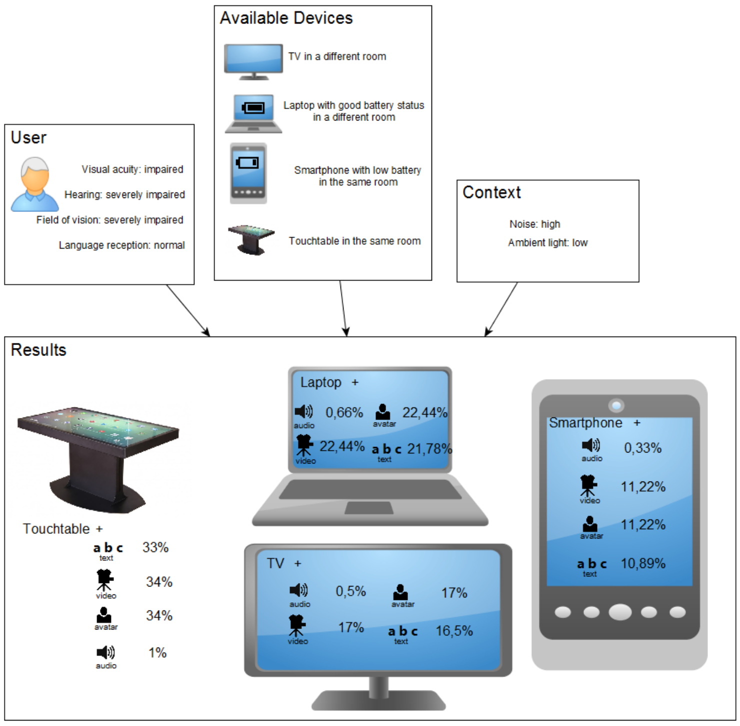


Figure 7.3: Scenario 3

CHAPTER 8

# Discussion & Conclusion

## 8.1 Discussion

### Scenario Results

In the previous chapter, three different scenarios were presented, along with the results, the AALuis Output Device and Modality Selection software produces. This section will give an interpretation for each of these scenarios.

#### Scenario 1

The situation described in this scenario, along with the results of the evaluation, was shown in Figure 7.1. The modalities audio, as well as video and avatar, which also include audio output, were penalized because of the user's impaired hearing abilities. Moreover, all combinations including the TV are downgraded because it is currently out of reach. None of the context information disfavor text output on the touchtable, which makes it the best result with 100% acceptance. Video is more agreeable than the use of an avatar, because in a video, there is a chance that the images shown contribute to the overall understanding even if the user struggles with the textual portion of the output. An avatar that produces spoken output without visual aids is therefore rated slightly lower in this context. Audio output on the TV in another room receives only a low agreement of 16.5%, as it is clearly the most disadvantageous combination given the context.

#### Scenario 2

Figure 7.2 shows the second scenario tested. Due to the bright light, all modalities using a visual component are penalized. The smartphone, due to its low battery, is at the bottom of the list of choices, while the laptop is the favored device in this situation. In this case, the results reflect no distinction between video output and the use of an avatar. According to the user context, the user would be able to process both equally well. The bright light has the same impact on them.

The device with low batteries is rated worse than the one in another room, because the use of this device could mean that the interaction is disrupted when the battery runs out. Going to the other room is an inconvenience but still the better choice in such a case.
If the bright light were not an issue, all modalities for the laptop would be rated with an agreement of 100%. Depending on the implementation, one of these options would probably be picked at random. This shows the consequences of the used model not including any personal preferences at this stage. Especially for fit users with no physical restrictions and a choice of devices, the inclusion of preferences in the model would make the results in ambiguous situations more clear and the behavior more predictable.

**Scenario 3**

Scenario 3 and the associated results were shown in Figure 7.3. In this case, all combinations are assigned probabilities of less than 50%. The best option is rated at 34% because all modalities are affected of the negative influences. While disadvantageous for the selection of device and modality, the result is in accordance with the concept that was applied for the creation of the model. This scenario describes a situation, where every possible negative influence appears. If there was a device and modality combination receiving a good result in this scenario, it would mean that it was not influenced by the modeled context. Therefore, it would always be among the top results after any probability query, regardless of the situation. The model would have to be reviewed, to make sure that no causal relationship has been overlooked .
Although low ratings for all combinations are to be expected in a worst case scenario, care must be taken if this worst case occurs frequently during real-time use. This could mean that the model might not be fit for the real context. The influencing factors on the modalities might be weighed too heavily. Reevaluating the Conditional Probability Tables (CPTs) of the modalities can improve the performance. This approach to alter the model is a good option, if the user is able to cope with their limitations better than expected. Thus, the best option would be to evaluate the behavior of the selection process in collaboration with the user.
Moreover, it may be possible to enhance conditions before initiating user interaction. In the case presented in scenario 3, changing the lighting conditions mends the situation significantly.
If this is not an option, another way to improve the evaluation results is to include devices, and hence modalities, which suit the user's needs better than the ones mentioned in this scenario. A Braille display, for instance, might be a useful addition for a user with impaired hearing and visual abilities.

The presented scenarios show that the evaluation results are plausible regarding the input. Single negative influences were shown to affect the rating of certain devices and modalities negatively, while others remain good options. Multiple influences result in a ranking that is congruent with the properties of the devices and modalities in question. The combination of all modeled negative influences results in very low ratings for all device and modality combination.

**Runtime Analysis**

In his book on Usability Engineering [48], Jakob Nielsen defines 100ms as the time up to which a user experiences the response of a computer system as instantaneous reaction. This number serves as a point of reference to evaluate the performance of the probability query.
The results of the runtime analysis depicted in Table 7.1 show that the first run takes more than 100ms, even for the minimal example. This time can be reduced by initiating the creation of the junction tree right after startup, instead of at the beginning of the first probability query.
For the subsequent runs, when the junction tree is already prepared, the average runtime ranges from 2.206 ms for the minimal network to 59.934ms for the large network of 208 nodes. This large network included 24 devices and 16 modalities, and the runtime of inference is still below 100ms. It seems unlikely this size will be exceeded by the model of a real home. Therefore, the results suggest, that execution time of the developed model and software are satisfactory, even for large households.

## 8.2 Conclusion

The goal of this thesis was to find a method to implement the context-aware selection of a suitable device and modality for the AAL middleware AALuis. Analysis of possible approaches showed that Bayesian Networks are a method satisfying the constraints of the task. In an iterative process a model was created. This model offers the possibility to reason on the available knowledge about the context and to provide a result in the form of the best combination of device and modality. It was integrated into a Java package to investigate the suitability of this approach for AALuis. The created software comprises of two parts, one offering methods to perform inference on the model and the second one featuring a GUI to facilitate the extension of the model for users or administrators. The performance of the evaluation software is satisfactory with regards to the execution time as well as the plausibility of the produced results.

## 8.3 Outlook

The implementation of the selection of output modality and device in Java using a Bayesian Network was successful. However, output is only one part of the user interaction. It remains to be seen if this approach is as suitable to select appropriate input modalities.
To validate the results of the output device and modality selection, the software should be tested in a real world ambient environment. Users belonging to the target audience of AAL systems should evaluate the results and their feedback to be incorporated into future developments.
Including the users in further steps is essential. Literature indicates that the completely automatic selection approach might be uncomfortable for users. Möller et al. [60] performed a requirement analysis for their framework supporting multimodal interaction in mobile applications with the integration of context information. Their focus group of users reported that they „would welcome a system that automatically selects modalities, but would like to retain control and be able to override the system behaviour.“.

The demand for control is understandable. Moreover, a feedback loop to enhance the Bayesian Network with data containing user's preferences may increase user acceptance.

During the creation of the model for this thesis, all modalities and devices were treated as equally able to convey information, and the nature of the interactions for which the Automatic IO Selection was performed were not specified. However, there are ideas to use devices facilitating the output of information in a discrete manner, e.g. colored LEDs or unobtrusive but distinctive sounds. Such ambient modalities cannot transmit the same density of information as a video or a text can, but they are appropriate to communicate simple facts.
To integrate such modalities into the proposed Automatic IO Device and Modality Selection, interactions could be annotated with information about which modalities are suitable for them. This information could be included into the probability queries readily, by setting evidence directly for modalities. For each interaction, evidence would be set to the instance *no* for unsupported modalities. No alterations to the Bayesian Network would be required.

APPENDIX A

# List of Abbreviations

| | | |
|---|---|---|
| **AAL** | Active and Assisted Living | iii |
| **CPT** | Conditional Probability Table | 29 |
| **DAG** | Directed Acyclic Graph | 25 |
| **EKG** | Electrocardiogram | 16 |
| **GUI** | Graphical User Interface | 3 |
| **HCI** | Human Computer Interaction | 2 |
| **ICT** | Information and Communication Technology | 1 |
| **ISTAG** | Information Society Technology Advisory Group | 6 |
| **MAP** | Maximum A Posteriori Hypthesis | 33 |
| **MPE** | Most Probable Explanation | 33 |
| **RFID** | Radio-frequency Identification | 12 |

**UI** User Interface . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iii

**WSN** Wireless Sensor Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

# Bibliography

[1] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, and Mohd Alauddin Mohd Ali. A review of smart homes. past, present, and future. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1190–1203, 2012.

[2] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[3] Michele Amoretti, Folker Wientapper, Francesco Furfari, Stefano Lenzi, and Stefano Chessa. Sensor data fusion for activity monitoring in ambient assisted living environments. In *Sensor Systems and Software*, pages 206–221. Springer, 2010.

[4] Stig K Andersen, Kristian G Olesen, Finn Verner Jensen, and Frank Jensen. Hugin-a shell for building bayesian belief universes for expert systems. In *IJCAI*, pages 1080–1085, 1989.

[5] Javier Andreu and Plamen Angelov. Real-time human activity recognition from wireless sensors using evolving fuzzy systems. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1–8. IEEE, 2010.

[6] Amaya Arcelus, Christophe L Herry, Rafik Goubran, Frank Knoefel, Heidi Sveistrup, Martin Bilodeau, et al. Determination of sit-to-stand transfer duration using bed and floor pressure sequences. *Biomedical Engineering, IEEE Transactions on*, 56(10):2485–2492, 2009.

[7] Juan Carlos Augusto. Ambient intelligence: the confluence of ubiquitous/pervasive computing and artificial intelligence. In *Intelligent Computing Everywhere*, pages 213–234. Springer, 2007.

[8] Juan Carlos Augusto and Paul McCullagh. Ambient intelligence: Concepts and applications. *Computer Science and Information Systems*, 4(1):1–27, 2007.

[9] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[10] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180, 2010.

[11] Gerald Bieber, Thomas Kirste, and Bodo Urban. Ambient interaction by smart watches. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, page 39. ACM, 2012.

[12] Marco Blumendorf, Sebastian Feuerstack, and Sahin Albayrak. Multimodal user interaction in smart environments: Delivering distributed user interfaces. In *Constructing Ambient Intelligence*, pages 113–120. Springer, 2008.

[13] Felix Breitenecker. Einführung in Modellbildung und Simulation. Lecture notes, 2015.

[14] Carmen Bruder, Lucienne Blessing, and Hartmut Wandke. Adaptive training interfaces for less-experienced, elderly users of electronic devices. *Behaviour & Information Technology*, 33(1):4–15, 2014.

[15] Laura Burzagli, Pier Luigi Emiliani, and Francesco Gabbanini. Ambient intelligence and multimodality. In *Universal Access in Human-Computer Interaction. Ambient Interaction*, pages 33–42. Springer, 2007.

[16] Gaëlle Calvary, Joëlle Coutaz, and David Thevenin. Supporting context changes for plastic user interfaces: a process and a mechanism. In *People and Computers XV Interaction without Frontiers*, pages 349–363. Springer, 2001.

[17] Eric Campo and Etienne Grangereau. Wireless fall sensor with gps location for monitoring the elderly. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 498–501. IEEE, 2008.

[18] Noëlle Carbonell. Ambient multimodality: towards advancing computer accessibility and assisted living. *Universal Access in the Information Society*, 5(1):96–104, 2006.

[19] Harry Lik Chen. *An intelligent broker architecture for pervasive context-aware systems*. PhD thesis, 2004.

[20] Diane J Cook. How smart is your home? *Science (New York, NY)*, 335(6076):1579, 2012.

[21] Christian Corsten, Ignacio Avellino, Max Möllers, and Jan Borchers. Instant user interfaces: repurposing everyday objects as input devices. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*, pages 71–80. ACM, 2013.

[22] Adnan Darwiche. Recursive conditioning. *Artificial Intelligence*, 126(1):5–41, 2001.

[23] Silvia de Miguel-Bilbao, Jorge García, M Dolores Marcos, and Victoria Ramos. Short range technologies for ambient assisted living systems in telemedicine: New healthcare environments, 2013.

[24] Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.

[25] Anind K Dey, Gregory D Abowd, and Daniel Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction*, 16(2):97–166, 2001.

[26] Dan Ding, Rory A Cooper, Paul F Pasquina, and Lavinia Fici-Pasquina. Sensor technology for smart homes. *Maturitas*, 69(2):131–136, 2011.

[27] A Dinh, D Teng, L Chen, SB Ko, Y Shi, J Basran, and V Del Bello-Hass. Data acquisition system using six degree-of-freedom inertia sensor and zigbee wireless link for fall detection and prevention. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 2353–2356. IEEE, 2008.

[28] Alan Dix. *Human-computer Interaction*. Pearson Education, 2004.

[29] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[30] Birgid Eberhardt, Uwe Fachinger, and Klaus-Dirk Henke. Better health and ambient assisted living (aal) from a global, regional and local economic perspective. *International Journal of Behavioural and Healthcare Research*, 2(2):172–191, 2010.

[31] Naeem Esfahani and Sam Malek. Uncertainty in self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II*, pages 214–238. Springer, 2013.

[32] Christoph Evers, Romy Kniewel, Kurt Geihs, and Ludger Schmidt. Achieving user participation for adaptive applications. In *Ubiquitous computing and ambient intelligence*, pages 200–207. Springer, 2012.

[33] Cardinaux Fabien, Bhowmik Deepayan, Abhayaratne Charith, S Mark, et al. Video based technology for ambient assisted living: A review of the literature. *Journal of Ambient Intelligence and Smart Environments (JAISE)*, 3(3):253–269, 2011.

[34] Ahmad Faridi and MM Hafizur Rahman. Hmm as an inference technique for context awareness. *Procedia Computer Science*, 59:454–458, 2015.

[35] Luis Fernández, José M Blasco, and José F Hernández. Wireless sensor networks in ambient intelligence. *Book Wireless Sensor Networks in Ambient Intelligence*, 2009.

[36] Emmanouil Georgakakis, Stefanos A Nikolidakis, Dimitrios D Vergados, and Christos Douligeris. An analysis of bluetooth, zigbee and bluetooth low energy and their use in wbans. In *Wireless Mobile Communication and Healthcare*, pages 168–175. Springer, 2011.

[37] AmirHosein GhaffarianHoseini, Nur Dalilah Dahlan, Umberto Berardi, Ali Ghaffarian-Hoseini, and Nastaran Makaremi. The essence of future smart houses: From embedding ict to adapting to sustainability principles. *Renewable and Sustainable Energy Reviews*, 24:593–607, 2013.

[38] Duncan F. Gillies. Intelligent data analysis and probabilistic inference lecture 10, 2015.

[39] Duncan F. Gillies. Intelligent data analysis and probabilistic inference lecture 11, 2015.

[40] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and computer applications*, 28(1):1–18, 2005.

[41] Mike Hazas, James Scott, and John Krumm. Location-aware computing comes of age. *Computer*, (2):95–97, 2004.

[42] Tsipi Heart and Efrat Kalderon. Older adults: are they ready to adopt health-related ict? *International journal of medical informatics*, 82(11):e209–e231, 2013.

[43] Sumi Helal, William Mann, Jeffrey King, Youssef Kaddoura, Erwin Jansen, et al. The gator tech smart house: A programmable pervasive space. *Computer*, 38(3):50–60, 2005.

[44] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. In *Pervasive Computing*, pages 167–180. Springer, 2002.

[45] Jennifer C Hou, Qixin Wang, Bedoor K AlShebli, Linda Ball, Stanley Birge, Marco Caccamo, Chin-Fei Cheah, Eric Gilbert, Carl Gunter, Elsa Gunter, et al. Pas: A wireless-enabled, sensor-integrated personal assistance system for independent and assisted living. In *High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, 2007. HCMDSS-MDPnP. Joint Workshop on*, pages 64–75. IEEE, 2007.

[46] ISTAG. Ambient intelligence: from vision to reality. for participation in society & business, 2003.

[47] Vikramaditya Jakkula, Diane J Cook, et al. Anomaly detection using temporal data mining in a smart home environment. *Methods of information in medicine*, 47(1):70–75, 2008.

[48] Nielsen Jakob. Usability engineering. *Fremont, California: Morgan*, 1993.

[49] M Humayun Kabir, M Robiul Hoque, and Sung-Hyun Yang. Development of a smart home context-aware application: A machine learning based approach. *learning*, 9(1), 2015.

[50] David Kahle, Terrance Savitsky, Stephen Schnelle, and Volkan Cevher. Junction tree algorithm. *STAT*, 631, 2008.

[51] Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. Ambient intelligence in assisted living: enable elderly people to handle future interfaces. In *Universal access in human-computer interaction. Ambient interaction*, pages 103–112. Springer, 2007.

[52] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[53] Timo Koski and John Noble. *Bayesian networks: an introduction*, volume 924. John Wiley & Sons, 2009.

[54] Pat Langley. Machine learning for adaptive user interfaces. In *KI-97: Advances in artificial intelligence*, pages 53–62. Springer, 1997.

[55] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

[56] Iolanda Leite, Carlos Martinho, and Ana Paiva. Social robots for long-term interaction: a survey. *International Journal of Social Robotics*, 5(2):291–308, 2013.

[57] Shing-Hong Liu, Yung-Fa Huang, and Chao-Rong Chen. The wireless holter ecg system based on zigbee. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 1816–1820. IEEE, 2014.

[58] Stephen Makonin, Lyn Bartram, and Fred Popowich. A smarter smart home: Case studies of ambient intelligence. *IEEE Pervasive Computing*, (1):58–66, 2013.

[59] Christopher Mayer, Martin Morandell, Matthias Gira, Kai Hackbarth, Martin Petzold, and Sascha Fagel. *AALuis, a user interface layer that brings device independence to users of AAL systems*. Springer, 2012.

[60] Andreas Möller, Stefan Diewald, Luis Roalter, and Matthias Kranz. Supporting mobile multimodal interaction with a rule-based framework. *arXiv preprint arXiv:1406.3225*, 2014.

[61] Richard E Neapolitan et al. *Learning bayesian networks*, volume 38. Prentice Hall Upper Saddle River, 2004.

[62] Jürgen Nehmer, Martin Becker, Arthur Karshmer, and Rosemarie Lamm. Living assistance systems: an ambient intelligence approach. In *Proceedings of the 28th international conference on Software engineering*, pages 43–50. ACM, 2006.

[63] Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan, Mary Rodgers, et al. A review of wearable sensors and systems with application in rehabilitation. *J Neuroeng Rehabil*, 9(12):1–17, 2012.

[64] Fabio Paternò, Cristiano Mancini, and Silvia Meniconi. Concurtasktrees: A diagrammatic notation for specifying task models. In *Human-Computer Interaction INTERACT ”97*, pages 362–369. Springer, 1997.

[65] Fabio Paterno, Carmen Santoro, and Lucio Davide Spano. Maria: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(4):19, 2009.

[66] Eric J Pauwels, Albert A Salah, and Romain Tavenard. Sensor networks for ambient intelligence. In *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on*, pages 13–16. IEEE, 2007.

[67] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach.

[68] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[69] Domenico Porcino and Walter Hirt. Ultra-wideband radio technology: potential and challenges ahead. *Communications Magazine, IEEE*, 41(7):66–74, 2003.

[70] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *Biomedical and Health Informatics, IEEE Journal of*, 17(3):579–590, 2013.

[71] Nicholas Roy, Gregory Baltus, Dieter Fox, Francine Gemperle, Jennifer Goetz, Tad Hirsch, Dimitris Margaritis, Michael Montemerlo, Joelle Pineau, Jamie Schulte, et al. Towards personal service robots for the elderly. In *Workshop on Interactive Robots and Entertainment (WIRE 2000)*, volume 25, page 184, 2000.

[72] Daniel Salber, Anind K Dey, and Gregory D Abowd. The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 434–441. ACM, 1999.

[73] William Noah Schilit. *A system architecture for context-aware mobile computing*. PhD thesis, Columbia University, 1995.

[74] Prakash P Shenoy and Glenn Shafer. Propagating belief functions with local computations. *IEEE Expert*, 1(3):43–52, 1986.

[75] Jeffrey Soar and Peter R Croll. Assistive technologies for the frail elderly, chronic illness sufferers and people with disabilities–a case study of the development of a smart home. 2007.

[76] In-Jee Song and Sung-Bae Cho. Bayesian and behavior networks for context-adaptive user interface in a ubiquitous home environment. *Expert Systems with Applications*, 40(5):1827–1838, 2013.

[77] Constantine Stephanidis. Human factors in ambient intelligence environments. 2012.

[78] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *Workshop Proceedings*, 2004.

[79] Thomas Vogel and Holger Giese. Model-driven engineering of self-adaptive software with eurema. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 8(4):18, 2014.

[80] Qinghua Wang and Ilangko Balasingham. *Wireless sensor networks-an introduction*. INTECH Open Access Publisher, 2010.

[81] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[82] N Watthanawisuth, T Lomas, A Wisitsoraat, and A Tuantranont. Wireless wearable pulse oximeter for health monitoring using zigbee wireless sensor network. In *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pages 575–579. IEEE, 2010.

[83] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

[84] Janine L Wiles, Annette Leibing, Nancy Guberman, Jeanne Reeve, and Ruth ES Allen. The meaning of „ageing in place“ to older people. *The gerontologist*, page gnr098, 2011.

[85] Jon S Wilson. *Sensor technology handbook*. Elsevier, 2004.

[86] Alex L Wood, Geoff V Merrett, Steve R Gunn, Bashir M Al-Hashimi, Nigel R Shadbolt, and Wendy Hall. Adaptive sampling in context-aware systems: a machine learning approach. In *Wireless Sensor Systems (WSS 2012), IET Conference on*, pages 1–5. IET, 2012.

[87] Tatsuya Yamazaki. Beyond the smart home. In *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, volume 2, pages 350–355. IEEE, 2006.

[88] Zhenhe Ye, Ying Li, Qiaoxiang Zhao, and Xue Liu. A falling detection system with wireless sensor for the elderly people based on ergnomics. *International Journal of Smart Home*, 8(1):187–196, 2014.

[89] Bin Yu, Lisheng Xu, and Yongxu Li. Bluetooth low energy (ble) based mobile electrocardiogram monitoring system. In *Information and Automation (ICIA), 2012 International Conference on*, pages 763–767. IEEE, 2012.

[90] Wolfgang L. Zagler, Paul Panek, and Marjo Rauhala. Ambient assisted living systems - the conflicts between technology, acceptance, ethics and privacy. In Arthur I. Karshmer, Jürgen Nehmer, Hartmut Raffler, and Gerhard Tröster, editors, *Assisted Living Systems - Models, Architectures and Engineering Approaches*, number 07462 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[91] Nevin Lianwen Zhang and David Poole. Exploiting causal independence in bayesian network inference. *Journal of Artificial Intelligence Research*, pages 301–328, 1996.

[92] Hubert Zimmermann. Osi reference model–the iso model of architecture for open systems interconnection. *Communications, IEEE Transactions on*, 28(4):425–432, 1980.

[93] Enrique Dorronzoro Zubiete, Luis Fernandez Luque, Ana Verónica Medina Rodríguez, and Isabel Gómez González. Review of wireless sensors networks in health applications. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 1789–1793. IEEE, 2011.