

Digital Aura: Pervasive Human-Human Interaction

Carl BRUNINX

promotor :
Prof. dr. Karin CONINX

Dankwoord

Deze thesis vormt het sluitstuk van mijn opleiding tot Master in de Informatica aan de universiteit van Hasselt. Een lange en moeilijke weg heb ik afgelegd om deze thesis tot een goed einde te brengen. In het doorlopen van dit traject werd ik bijgestaan door een aantal mensen die ik graag wil bedanken. Ten eerste bedank ik mijn promotor, prof. dr. Karin Coninx, co-promotor, dr. Kris Luyten, waar ik altijd terecht kon voor advies en hulp. Vervolgens wil ik mijn begeleiders, Tim Clerckx en Geert Houben, bedanken. Zij hebben me goed op weg geholpen en waren altijd bereid om me met raad en daad uit de nood te helpen. Ten slotte een woord van dank voor de personen uit mijn naaste omgeving voor hun steun tijdens de moeilijke momenten. In het bijzonder mijn vriendin, Sarah, die naast een goede steun ook bereid was deze tekst na te lezen en te verbeteren.

Abstract

De groeiende wereld van “Ubiquitous Computing” brengt een aantal interactieproblemen met zich mee. De klassieke manier van interageren met een systeem door input van muis, toetsenbord, enz. is niet meer voldoende. Bijgevolg zijn nieuwe modellen voor interactie nodig. In deze thesis is het interactiemodel “Digital Aura”, dat een oplossing vormt voor de interactieproblemen, ontwikkeld. Door middel van een metaforische aura worden impliciete interacties tot stand gebracht.

Binnen dit interactiemodel kunnen een aantal onderdelen onderscheiden worden: gebruikersmodellen, geheimhouding en persoonlijke agenten. In deze thesis wordt voor elke onderdeel de studie en de ontwikkeling ervan besproken.

De implementaties van deze verschillende onderdelen vormen een concrete realisatie van het besproken interactiemodel. Ten slotte worden op basis van dit interactiemodel een tweetal toepassingen aangebracht, die de mogelijkheden van dit interactiemodel aantonen.

Inhoudsopgave

1	Inleiding	1
1.1	Ubiquitous Computing	1
1.2	Digital Aura	2
1.2.1	Inleiding	2
1.2.2	Toepassingen	3
1.2.3	Structuur	4
2	Gebruikersmodel	6
2.1	Inleiding	6
2.2	Gerelateerd Werk	7
2.3	Studie en vergelijking	8
2.3.1	Structuur	9
2.3.2	Inhoud	10
2.3.3	UM Toolkit	11
2.3.4	UserML en GUMO	13
2.4	Gebruikersprofiel	15
2.4.1	Inhoud	16
2.4.2	Structuur	17
2.5	Conclusie	17
3	Geheimhouding	19
3.1	Inleiding	19
3.2	Gerelateerd werk	20
3.3	Risicomodel	21
3.3.1	Analyse	22
3.3.2	Management	24
3.4	5 Valkuilen	25
3.5	Geheimhoudingsprofiel	27

3.5.1	UMToolkit en UserML	28
3.5.2	Profiel voor Digital Aura	28
3.6	Conclusie	30
4	Persoonlijke Agent	32
4.1	Inleiding	32
4.2	Gerelateerd Werk	33
4.3	Definitie	35
4.4	Topologie	37
4.5	Agent voor Digital Aura	38
4.5.1	Autonoom	39
4.5.2	Samenwerken	40
4.5.3	Leren	40
4.5.4	Andere eigenschappen	41
4.5.5	Collaboratieve agent	41
4.6	Conclusie	41
5	Implementatie	43
5.1	Inleiding	43
5.2	Communicatie	44
5.2.1	Bluetooth	46
5.2.2	Wireless LAN	48
5.3	Gebruikersmodel	49
5.3.1	Relatie gebruikersprofiel - geheimhoudingsprofiel	50
5.3.2	Gebruikersprofiel	51
5.3.3	Geheimhoudingsprofiel	53
5.3.4	Valkuilen	55
5.4	Geheimhouding	56
5.5	Persoonlijke agent	58
5.5.1	Eigenschappen	58
5.5.2	Taken	59
5.5.3	Uit-Agent	60
5.5.4	In-Agent	62
5.6	Toepassingen	63
5.6.1	Hulp nodig	63
5.6.2	Poster	64
5.7	Conclusie	65

6 Conclusie	67
6.1 Bevindingen	67
6.1.1 Gebruikersmodel	67
6.1.2 Geheimhouding	68
6.1.3 Persoonlijke agent	68
6.1.4 Implementatie	68
6.2 Toekomstig werk	68

Lijst van figuren

1.1	Metaforische aura's	3
2.1	Gebruikersprofiel van UMLToolkit	12
2.2	Gebruikersmodel in UserML	14
2.3	Gebruikersprofiel voor Digital Aura	16
3.1	Schematisch overzicht van risicomodel	22
3.2	Gebruikersmodel voor Digital Aura	30
4.1	Nwana's topologie van software agents	37
5.1	Schematische voorstelling van de werking van Digital Aura. . .	44
5.2	Berichten voor communicatie via Wireless LAN	49
5.3	XML fragment van een gebruikersprofiel	51
5.4	Screenshot van editor voor gebruikersprofiel	52
5.5	XML fragment van een geheimhoudingsprofiel	54
5.6	Screenshot van editor voor geheimhoudingsprofiel	55
5.7	Overzicht werking persoonlijke agent	60
5.8	Screenshot van melding voor gebruiker bij interessante data . .	63

Hoofdstuk 1

Inleiding

1.1 Ubiquitous Computing

In de laatste decennia heeft de technologische ontwikkeling een enorme vooruitgang gemaakt. Een belangrijke, zometert de belangrijkste rol in deze ontwikkeling stond in het teken van de computer. Naast een steeds toenemende rekenkracht van deze computers of andere gerelateerde systemen, werd ook hun grootte steeds kleiner. Deze vooruitgang zet zich nu nog voort en lijkt de komende jaren niet te zullen stilvallen. Nieuwe toepassingen en/of toepassingsgebieden zijn ontegensprekelijk verbonden met de ontwikkeling van nieuwe, kleinere en betere systemen.

Sinds enkele jaren is er een punt bereikt waarop computers klein en krachtig genoeg zijn om gebruikt te kunnen worden op andere manieren dan een standaard “desktop” of een “laptop”. Ze veroverden een plaats in tal van andere voorwerpen die dagelijks gebruikt worden zoals een koffiezetapparaat, auto, telefoon, enz. Zulke gebruiksvoorwerpen die een computer bevatten, worden ook wel “embedded systems” systemen genoemd. Een blik op de toekomst leert ons dat we uiteindelijk zullen leven in een wereld die omgeven is door computers, al dan niet zicht- of merkbaar.

De vele computers die ons omgeven, zijn onderling met elkaar verbonden, via een eventueel draadloos netwerk. Het doel is mensen in hun dagelijkse taken te ondersteunen. Een veelgebruikte term om dit soort van systemen

aan te geven is “Ubiquitous Computing”¹ en werd als eerste gegeven door Marc Weiser [Wei91, Wei96].

“Ubiquitous Computing is roughly the opposite of virtual reality. Where virtual reality puts people inside a computer-generated world, Ubiquitous Computing forces the computer to live out here in the world with people.” (Mark Weiser)

1.2 Digital Aura

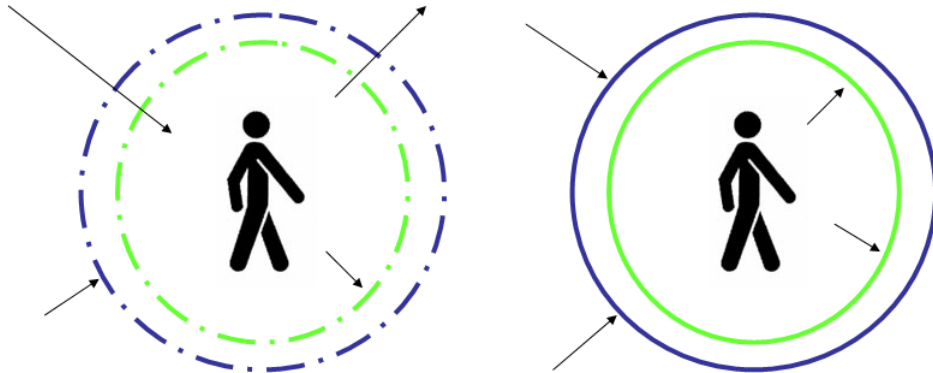
1.2.1 Inleiding

De intrede van “Ubiquitous Computing” en bijhorende integratie van computers in het dagelijkse leven zal zorgen voor een totaal andere visie met betrekking tot het gebruik van computers. De klassieke manier van interactie met een computer, namelijk het gebruik van een muis, toetsenbord, enz. voor input en scherm, printer, enz. voor output, volstaat niet meer. De geïntegreerde computers zijn niet meer zichtbaar of de gebruiksvoorwerpen zijn zo klein dat klassieke interactie niet meer kan plaatsvinden. Een andere reden is dat er bij de gebruikers nood is aan computers die hun werk doen zonder veel inbreng van de persoon zelf, bijvoorbeeld een auto waarvan de zetels zich automatisch in een juiste positie zetten afhankelijk van de bestuurder. Ten slotte moeten computers ook met elkaar kunnen interageren, zonder tussenkomst van mensen. Er is bijgevolg nood aan nieuwe interactiemodellen opdat “Ubiquitous Computing” goed en succesvol kan zijn.

Om voorgaande problemen tegemoet te komen werd het interactiemodel “Digital Aura” voorgesteld [FHM⁺04]. Dit model is gebaseerd op het “Focus and Nimbus” interactiemodel [Rod96], dat rekening houdt met de fysieke nabijheid, locatie, enz. van personen en/of computers. Het “Nimbus” concept beschrijft welke informatie men wilt laten zien binnen een bepaald gebied of aan welke personen. Het “Focus” concept daarentegen beschrijft in welke informatie een gebruiker of computer geïnteresseerd is. De doorsnede tussen beide concepten van verschillende gebruikers of computers geeft bijgevolg interactie tussen hen aan.

¹ander gebruikte termen voor Ubiquitous Computing zijn “pervasive computing”, “calm technologie” en “things that think”.

“Digital Aura” maakt gebruik van een metaforische aura ² in plaats van een “Focus” en een “Nimbus”. Wanneer zulke metaforische aura’s in de buurt van elkaar komen en interacteren ontstaat er een spontane interactie tussen beide bronnen van de aura’s. Op deze manier wordt een impliciete communicatie tussen personen en/of computers tot stand gebracht.



Figuur 1.1: Metaforische aura’s

In figuur 1.1 worden zulke metaforische aura’s getoond. Net als bij het “Focus and Nimbus” model bestaat een metaforische aura uit twee delen die aangeven in welke informatie men geïnteresseerd is en welke informatie men wilt dat uitgezonden wordt. In de linker aura wordt toegelaten dat informatie zowel ontvangen als uitgezonden wordt. Hierbij wordt echter niet alle informatie toegelaten bij het ontvangen of verzenden. In de rechteraura daarentegen wordt geen enkele informatie toegelaten en mag ook er ook geen verzonden worden. Indien twee metaforische aura’s elkaar zouden snijden, vindt tussen de twee bronnen van de aura impliciete interactie plaats.

1.2.2 Toepassingen

Het interactiemodel “Digital Aura” kan een basis vormen voor een aantal toepassingen binnen het domein van “Ubiquitous Computing”. Hierbij speelt voornamelijk de mogelijkheid tot impliciete interactie van het interactiemodel een grote rol. Personen en/of computers kunnen eender waar en wanneer spontaan met elkaar beginnen te communiceren.

²Een aura is een ovaal energieveld dat uitgezonden wordt door een levend wezen.

Een eerste voorbeeldtoepassing, die vermeld wordt in het werk van Ferscha e.a. [FHM⁺04] in verband met het interactiemodel “Digital Aura”, is een student die hulp nodig heeft bij een algoritme voor patroonherkenning. Deze student maakt dit duidelijk door een verzoek tot hulp aan te geven. Na het instellen dit verzoek gaat de student gewoon verder met haar andere bezigheden. Indien een leraar of assistent, die deze algoritmen beheerst, na verloop van tijd in de buurt van de student komt en dit verzoek spontaan opvangt, wordt hij/zij verwittigd van het feit de student hulp nodig heeft en kan hij/zij met die student contact zoeken en vervolgens uitleg verschaffen.

Een andere mogelijke toepassing, uit hetzelfde werk van Ferscha e.a., beschrijft het scenario waarbij een persoon voorbij een poster van een concert wandelt. Deze poster is voorzien van een kleine computer. Wanneer de persoon de poster passeert, ontstaat tussen de persoon en de poster een impliciete interactie waarbij informatie over het concert uitgezonden wordt. Indien de persoon interesse heeft voor zulke concerten en dit bekend gemaakt heeft binnen het interactiemodel, kan een herinnering aan dit concert toegevoegd worden in zijn/haar agenda. In dit scenario is dus sprake van interactie tussen een persoon en een computer.

1.2.3 Structuur

Het interactiemodel “Digital Aura” kan opgesplitst worden in een aantal verschillende onderdelen. Een eerste onderdeel is de communicatie ³, die ontstaat bij het intersecteren van de metaforische aura’s. Hierbij wordt informatie over de personen uitgewisseld. Deze informatie is een belangrijk aspect van dit interactiemodel. Er rijzen bijgevolg een aantal vragen over de informatie: welke informatie moet beschikbaar zijn? Hoe moet deze informatie geordend worden? ... Deze en tal van andere vragen hebben een grote invloed op de samenstelling van de informatie, en bijgevolg ook op de werking van het interactiemodel. Welke mogelijkheden er ook zijn, een gemeenschappelijke eigenschap van deze informatie is het beschrijven van karakteristieken van personen en/of computers. De informatie vormt een beschrijving voor een specifieke persoon (gebruikersmodel) of computer (sys-

³Vanaf nu wordt er enkel gesproken over communicatie tussen personen. Dit kan evenzeer vervangen worden door communicatie tussen computers onderling of tussen personen en computers.

teemmodel). Antwoorden op bovenstaande vragen en de ontwikkeling van een geschikt gebruikersmodel worden gegeven in hoofdstuk 2.

Wanneer spontane en impliciete interactie optreedt waarbij informatie over personen uitgewisseld wordt, stelt zich onmiddellijk de vraag naar geheimhouding. Een persoon wenst niet altijd alle gegevens van hem/haar bekend te maken of alle gegevens van een andere persoon te ontvangen. De mogelijkheid om informatie te weerhouden of te negeren moet dus in het interactiemodel voorzien worden. Hoofdstuk 3 beslaat geheimhouding in het algemeen, heeft betrekking tot geheimhouding in verband met het gebruikersmodel en past geheimhouding toe op het totale interactiemodel.

Op basis van de uitgewisselde gegevens van verschillende personen kunnen vergelijkingen gemaakt worden. Op basis van deze vergelijkingen kan verder interactie gebeuren op applicatieniveau. In hoofdstuk 4 worden persoonlijke agenten besproken die deze vergelijking voor hun rekening nemen.

Ten slotte wordt in hoofdstuk 5 een beschrijving gegeven van een werkend interactiemodel. Dit interactiemodel werd gebaseerd op de bevindingen uit deze thesis. Verder werden op basis van dit interactiemodel ook enkele toepassingen ontwikkeld, die ook beschreven staan in dit hoofdstuk.

Hoofdstuk 2

Gebruikersmodel

2.1 Inleiding

Naast de eisen voor meer rekenkracht, kleiner formaat en mooiere graphics van een systeem, is de laatste jaren de vraag ontstaan naar een systeem dat afgestemd is op de persoonlijke voorkeuren van een gebruiker. Om zulke systemen tot stand te brengen, moeten ze over gegevens beschikken van een gebruiker. Een oplossing voor het beschikbaar maken van die informatie werd gevonden in het gebruik van gebruikersmodellen. Deze modellen bevatten informatie over de karakteristieken en voorkeuren van de gebruiker. Op basis hiervan kan een systeem zich bijgevolg aanpassen aan de gebruiker, bijvoorbeeld door aanpassingen van getoonde informatie, applicaties, vormgeving, enz.

De personalisatie van informatie of applicaties heeft ook een grote invloed op de interactie tussen mens en machine. Informatie of een applicatie die afgestemd is op de noden van de gebruiker kan ervoor zorgen dat taken met behulp van de computer sneller en beter uitgevoerd worden. Daarnaast is het voor de gebruiker ook aangenamer om te werken met een systeem dat hem/haar aanvoelt in zijn/haar handelingen. Omwille van deze voordelen werd in tal van gebieden onderzoek verricht naar het modelleren van gebruikers met hun specifieke kenmerken en noden. Een bespreking van dit onderzoek kan men terugvinden in de volgende sectie.

Wanneer men spreekt over een gebruikersmodel ¹ is niet enkel de persoonlijke informatie van de gebruiker van belang, maar ook het beeld dat een gebruiker heeft van geheimhouding. Een gebruiker wenst niet ten alle tijden, alle of een gedeelte van zijn persoonlijke informatie vrij te geven. Bijgevolg kan een gebruikersmodel opgesplitst worden in twee afzonderlijke delen. Het eerste deel, zoals hierboven besproken, bevat persoonlijke informatie over de gebruiker en wordt het gebruikersprofiel genoemd. Het tweede deel van het gebruikersmodel, dat de geheimhouding van een bepaalde gebruiker beschrijft, noemen we het geheimhoudingsprofiel.

Op basis van voorafgaand onderzoek, zal in dit hoofdstuk getracht worden een geschikt gebruikersprofiel te ontwikkelen dat dienst kan doen binnen het interactiemodel “Digital Aura”. Om het gebruikersmodel te vervolledigen zal in hoofdstuk 3 het geheimhoudingsprofiel ontwikkeld worden, aangezien daar een basis gelegd wordt voor de ontwikkeling van goede geheimhouding.

2.2 Gerelateerd Werk

Een kort overzicht van de ontwikkelingen in dit domein gedurende de laatste decennia kan men terugvinden in [Kob01]. Hieronder worden een aantal toepassingen beschreven die gebruik maken van gebruikersmodellen, alsook enkele belangrijke en recente ontwikkelingen binnen het domein.

Een eerste toepassing van gebruikersprofielen is het bekende gebruik van assistenten in het Microsoft Office pakket [Hor93, HBH⁺98]. Deze assistenten ondersteunen de gebruiker in het gebruik van de applicatie, door specifieke informatie of acties te tonen. De getoonde informatie of acties worden bepaald door een aantal parameters. Deze parameters zijn onder andere observatie van de gebruiker, toestand van de applicatie, sequentie van acties doorheen de tijd, enz. Ook op het internet wordt gebruik gemaakt van gebruikersprofielen om aangepaste informatie te tonen op basis van noden of interesses van de gebruiker. De Amazone website [Ama] is een online shopping service met de mogelijkheid tot het creëren van een account. Zulke accounts bieden

¹In dit werk wordt de term “gebruikersmodel” beschouwd als de verzameling van persoonlijke data en data over de geheimhouding. In andere werken wordt soms enkel persoonlijke informatie bedoeld.

gebruikers gepersonaliseerde informatie aan over mogelijk interessante aankopen. Deze informatie is gebaseerd op de geschiedenis van aankopen of op het gedrag van gebruikers die dezelfde aankopen gedaan hebben.

Naast aangepaste informatie, kan ook de applicatie zelf aangepast worden voor een gebruiker. Door gebruik te maken van UIML [APB⁺99], kan de interface van een applicatie op maat gemaakt worden, dit met betrekking tot de kenmerken van een bepaald systeem zoals resolutie, processorcapaciteit. Dit kan ook gecombineerd worden met kenmerken of de context van een gebruiker. In [SM03] worden applicaties aangepast aan de context waarin de gebruiker zich bevindt, de noden van de gebruiker en de kenmerken van het systeem waarop de applicatie draait. Een voorbeeld van zulke situatie is een gebruiker die een e-mail ontvangt. Wanneer de gebruiker zich voor een desktop systeem bevindt kan deze mail gemakkelijk getoond worden. Maar wanneer de gebruiker dan een auto bestuurt, mag de mail niet getoond worden (kan eventueel voorgelezen worden). Eenzelfde aanpak vinden we terug in [CCPC04a], waarbij een “agent” hulp verleent aan de gebruiker en zich aanpast aan de situatie, zowel op inhoudelijk als presentatieniveau.

Een belangrijke ontwikkeling binnen het domein van user modeling is afkomstig van Judy Kay. In de door haar ontwikkelde UM toolkit [Kay95] worden gebruikers gemodelleerd door observaties van hun kennis, geloof, enz. met betrekking tot bepaalde programma's. Ieder informatiegegeven wordt voorzien van een lijst met bewijzen die de (on)waarheid ervan aantoont. Verder wordt er ook belang gehecht aan de toegankelijkheid van het gebruikersprofiel voor de gebruiker. Een andere belangrijke en recente ontwikkeling is gebaseerd op de nood aan een uniforme notatie voor de gebruikersprofielen, naarmate “Ubiquitous Computing” steeds toeneemt. Dominik Heckman e.a [Hec04, HSB⁺05, HK03] stelt zulke uniforme notatie voor aan de hand van de “User Modeling Markup Language” (UserML) en een bijhorende ontologie, namelijk “the General User Model Ontology” (GUMO).

2.3 Studie en vergelijking

Het werk van Dominik Heckmann e.a. en Judy Kay, beschreven in vorige sectie, biedt een goede uitgangspositie voor de ontwikkeling van een geschikt

gebruikersprofiel voor het interactiemodel “Digital Aura”. Het bestuderen en vergelijken van hun ontwikkelingen kunnen enkele fundamentele eigenschappen, waaraan een gebruikersprofiel moet voldoen, aan het licht brengen. Op basis van deze eigenschappen kan er verder worden gewerkt. Om een degelijke studie of vergelijking te maken is er nood aan achtergrondinformatie waaraan hun werk getoetst kan worden. Enkele belangrijke begrippen worden in de volgende secties kort beschreven. Verder wordt ook hun werk bestudeerd en vergeleken.

2.3.1 Structuur

Wanneer een hoeveelheid data gegeven is, is het belangrijk om deze data duidelijk te ordenen en te structureren. Een goede structuur bevordert de toegankelijkheid van de informatie. Niet alleen de gebruiker is hiermee gediend (bijvoorbeeld bij het aanpassen van de data of bij het extraheren van bepaalde stukken data), maar ook het systeem dat als doel heeft deze data te verwerken.

Voor het structureren van data zijn er verschillende mogelijkheden voorhanden. De meest gekende en in het oog springende is uiteraard de “Extensible Markup Language” (XML) [XML]. XML is een eenvoudige en structurende taal die opgebouwd wordt uit zelf gekozen elementen door middel van “tags”, die eventueel voorzien kunnen worden van attributen. De structuur die door het gebruik van XML ontstaat is hiërarchisch van aard. Hiernaast voorziet XML ook de mogelijkheid om een “Document Type Declarations” (DTD) te specificeren. Op deze manier kan een familie van verwante structuren opgebouwd worden die allemaal voldoen aan bepaalde vastgelegde eigenschappen. Ten slotte kan een XML bestand ook eenvoudig omgevormd worden naar een andere taal of structuur door middel van een “Extensible Stylesheet Language Transformations” (XSLT) [XSL]. Hierdoor is een grotere schaalbaarheid en flexibiliteit mogelijk.

Naast de reeds beschreven voordelen is XML ook een goed aanvaarde standaard binnen de internetgemeenschap. Vele applicaties ondersteunen XML omdat het verwerken van XML bestanden eenvoudig is. Alles in acht genomen kan XML dus beschouwd worden als een geschikte kandidaat om de informatie van personen of systemen te structureren.

XML is echter enkel een structurerende taal. De gebruiker moet zelf een notatie voorzien om de informatie mee voor te stellen. Om hiervoor een oplossing te bieden, werd het “Resource Description Framework” (RDF) [RDF] ontworpen. Deze standaard is ontworpen met als doel het representeren van metadata, waarbij XML als syntax gebruikt wordt. RDF is echter beperkt in interpreteerbaarheid door een systeem en in de mogelijkheid om eigenschappen en klassen te beschrijven. Een uitbreiding van RDF werd daarom voorzien in de “Web Ontology Language” (OWL) [OWL]. OWL is een taal voor het beschrijven van een ontologie. *Een ontologie geformuleerd over een domein is typisch een hiërarchische datastructuur die alle relevante entiteiten en hun onderlinge relaties en regels binnen dat domein bevat.* Een ontologie voor een gebruikersprofiel gebaseerd op OWL/RDF wordt beschreven in sectie 2.3.4.

2.3.2 Inhoud

De kern van een gebruikersprofiel bestaat uit de informatie. Deze informatie beschrijft bepaalde eigenschappen of mogelijkheden van een persoon of een systeem. Welke kenmerken en/of mogelijkheden beschreven moeten worden in zulk gebruikersprofiel is afhankelijk van de applicatie en/of de situatie. De applicatie bepaalt op basis van de informatie, die bevat ligt in het gebruikersprofiel, wat er moet gebeuren. Een voorbeeld hiervan kan men terugvinden in UIML (zie sectie 2.2). Een soort gebruikersprofiel wordt opgesteld waarin wordt beschreven welke de voorkeuren van de gebruiker zijn, zoals kleuren, figuren enz. Op basis van de inhoud van dit gebruikersprofiel wordt dan een aangepaste “Graphical User Interface” (GUI) gemaakt die aangepast is aan de wensen van de gebruiker.

De wisselwerking tussen inhoud en applicatie kan ook in omgekeerde richting werken. De inhoud van de informatie van een gebruikersprofiel is ook afhankelijk van hetgeen een applicatie moet realiseren. In het vorige scenario moet een gebruikersprofiel informatie bevatten over de eigenschappen van een bepaald apparaat. In het geval van Amazon (zie sectie 2.2), moet er informatie over de geschiedenis van aankopen van een gebruiker opgeslagen worden, opdat een lijst met mogelijk interessante aankopen getoond kan worden.

Verschillende situaties geven in sommige gevallen ook aanleiding tot een andere inhoud van gebruikersprofielen. Beschouw het scenario van een gebruiker die een programma volgt op televisie, plotseling uit de zetel opstaat om elders heen te gaan, en toch het programma wil blijven bekijken via zijn PDA. Wanneer de gebruiker zich in de zetel bevindt, zal zijn gebruikersprofiel informatie bevatten over de eigenschappen van de televisie. Wanneer hij echter rondloopt verandert deze informatie in informatie over de PDA. Op basis van deze informatie moet de beeldkwaliteit bijgevolg aangepast worden opdat de gebruiker het programma kan blijven volgen. Aanpassing van informatie met betrekking tot situatie hangt vaak ook samen met de geheimhouding van informatie zoals beschreven wordt in volgende sectie.

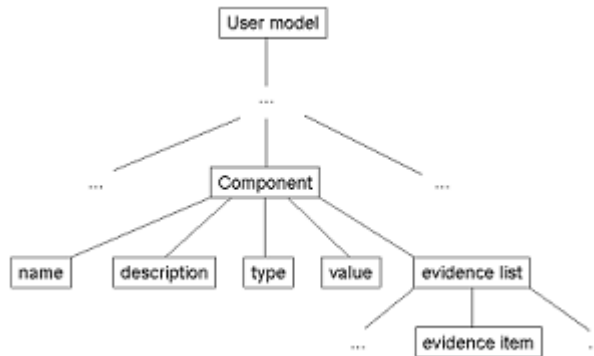
2.3.3 UM Toolkit

De UM toolkit is het werk van Judy Kay [Kay95] en ontstond uit de nood aan systemen die samenwerkten met de gebruikers in verschillende activiteiten. Door gebruik te maken van gebruikersprofielen², beschikt een systeem over meer informatie om zo de gebruiker beter van dienst te zijn. Twee concrete voorbeelden worden besproken: een coaching systeem om gebruikers meer te leren over een teksteditor en een adviseur voor een filmdatabank die de gebruiker helpt bij zijn filmkeuze. Hoewel er geen concrete specificatie of implementatie is voor de gebruikte gebruikersprofielen, worden toch enkele concepten en/of keuzes, die nuttig kunnen zijn, behandeld.

De kern van de gebruikersprofielen zijn de componenten. Deze stellen acties van de teksteditor of kenmerken van bepaalde concepten voor. Deze componenten worden voorzien van een bewijzenlijst. Op basis van deze bewijzen kunnen systemen die het gebruikersprofiel interpreteren, keuzes maken in hun handelingen ten opzichte van de gebruiker. Deze bewijzen geven een duidelijker beeld van de gebruiker en hebben bijgevolg een grote invloed op de werking van het systeem dat de gebruiker moet ondersteunen. Het concept zoals het wordt beschreven is applicatie-afhankelijk. Toch kan een veralgemening ervan zeker een meerwaarde bieden aan een gebruikersprofiel.

²In de UMToolkit wordt enkel gebruik gemaakt van persoonlijke informatie, hoewel ze er spreken over gebruikersmodellen.

Toegankelijkheid tot het gebruikersprofiel is een ander belangrijk concept binnen de UM toolkit. Indien het gebruikersprofiel gemakkelijk toegankelijk is, kan de gebruiker, indien er sprake is van conflicten met de verzamelde bewijzen, op eenvoudige wijze het gebruikersprofiel aanpassen. Een bijkomend voordeel hiervan is dat de gebruiker zich beter bewust is van de informatie die erin verwerkt is. De toegankelijkheid wordt vergroot door het gebruikersprofiel te bewaren in de bestandsruimte van de gebruiker zelf en door de informatie in een goede structuur te gieten. Er wordt geen melding gemaakt van het gebruik van een op XML gebaseerde taal. Toch kan er uit de gebruikte viewer tools een duidelijke boomstructuur afgeleid worden.



Figuur 2.1: Gebruikersprofiel van UMToolkit

In figuur 2.1 wordt een schematische voorstelling van het gebruikersprofiel uit de UM toolkit weergegeven. Tussen de root en de componenten is elke vertakkingsstructuur mogelijk. In het voorbeeld uit het werk van Judy Kay wordt gebruik gemaakt van een vertakkingsstructuur met vier niveaus. Componenten worden opgedeeld in vijf onderdelen: *name*, *description*, *type*, *value* en *evidence list*. Zoals de naam het reeds aangeeft, is deze laatste een lijst van verschillende *evidence items*. Meer details over de vertakkingsstructuur en onderdelen van de componenten, kunnen teruggevonden worden in het hierboven vermelde werk.

2.3.4 UserML en GUMO

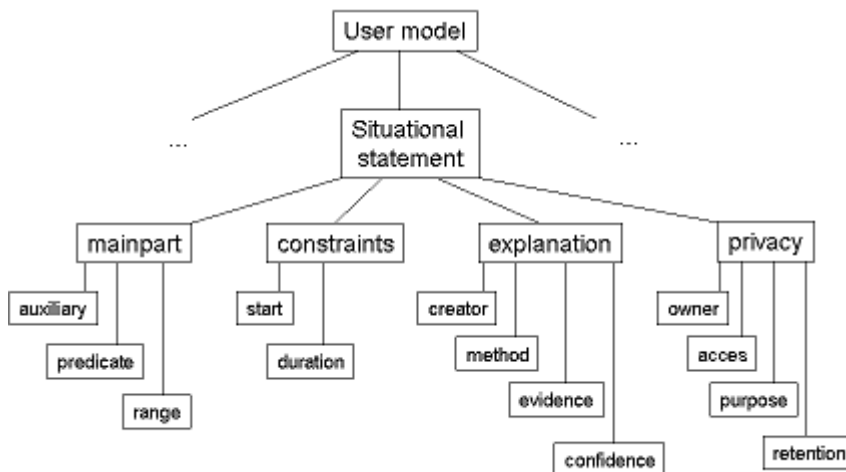
In de groeiende wereld van “Ubiquitous Computing” is er, zoals reeds werd aangehaald, nood aan een uniforme notatie. Dit werd in 1995 reeds beweerd door Jon Orwant in zijn Doppelgänger project [Orw95]: *We need a protocol for encoding information about users, any given user modeling system should be able to benefit from others and that user models should follow you around.* In het werk van Dominik Heckman e.a. [Hec04, HSB⁺05, HK03] wordt getracht een uniforme notatie te ontwikkelen. Deze notatie is gebaseerd op twee onderdelen: een “User Modeling Markup Language” (userML) en een “General User Model Ontology” (GUMO).

UserML is een taal voor het beschrijven van een gebruikersmodel ³, die gebaseerd is op XML. Het belangrijkste concept van userML zijn de *situational statements* die opgedeeld zijn in drie onderdelen: *auxiliary*, *predicate* en *range*. Auxiliaries zijn een soort werkwoord van de vorm *hasKnowledge*, *hasInterest*, enz. Het predicate is het onderwerp waarop de auxiliaries duiden, bijvoorbeeld *football*, *symphonies*, enz. Range ten slotte bepaalt het bereik waarin de waarde, geassocieerd met een bepaald auxiliary en predicate, moet liggen. Het hoofdonderdeel van een situational statement wordt verder uitgebreid met een *constraint*, een *explanation* en een *privacy* onderdeel. Constraints bevatten informatie over het tijdstip waarop de informatie werd vastgelegd en de periode waarin deze informatie als geldig beschouwd mag worden. Explanations geven bijkomende details over wie en op welke manier de informatie verkregen werd, alsook een graad van vertrouwen erin. Een constraint kan samen met een explanation beschouwd worden als een veralgemening van de bewijzenlijst bij de UM toolkit, zie vorige sectie. Het *privacy* onderdeel tenslotte geeft een aantal kenmerken van de informatie, namelijk aan wie de eigenaar is, welke toegangsrechten het heeft, wat het doel ervan is en wat weerhouden moet worden. Geheimhouding met betrekking tot de informatie wordt verder besproken in sectie 3.5.

Aanvankelijk werd alles beschreven in userML. Maar om de flexibiliteit en schaalbaarheid te vergroten werd gekozen voor een meer modulaire aanpak. In deze aanpak wordt gebruik gemaakt van een ontologie. Deze ontologie be-

³UserML bevat zowel persoonlijke informatie als geheimhoudingsinformatie, bijgevolg gebruiken we de term gebruikersmodel. (zie ook sectie 3)

schrijft de eigenschappen en kenmerken van de verschillende onderdelen die terug te vinden zijn in userML. Hierdoor kunnen verschillende gebruikersmodellen aangemaakt worden door een andere ontologie te hanteren binnen userML. De ontologie die ontwikkeld werd door Dominik Heckmann e.a. is GUMO. Deze ontologie, gebaseerd op OWL/RDF (zie sectie 2.3), bevat mogelijkheden voor het beschrijven van zowel personen als systemen. Informatie over een persoon kan gaan van basiskennmerken zoals naam en woonplaats tot uitgebreide beschrijvingen van de emotionele toestand of persoonlijkheid. Daarnaast is er ook de mogelijkheid voor het beschrijven van andere contextinformatie, zoals de huidige plaats en/of tijd waarin men zich bevindt, acties die men uitvoert (wandelen, geven), enz. Een meer uitgebreide beschrijving van GUMO kan men terugvinden op <http://www.gumo.org/>. Met behulp van GUMO kan zo goed als alles beschreven worden in een gebruikersmodel.



Figuur 2.2: Gebruikersmodel in UserML

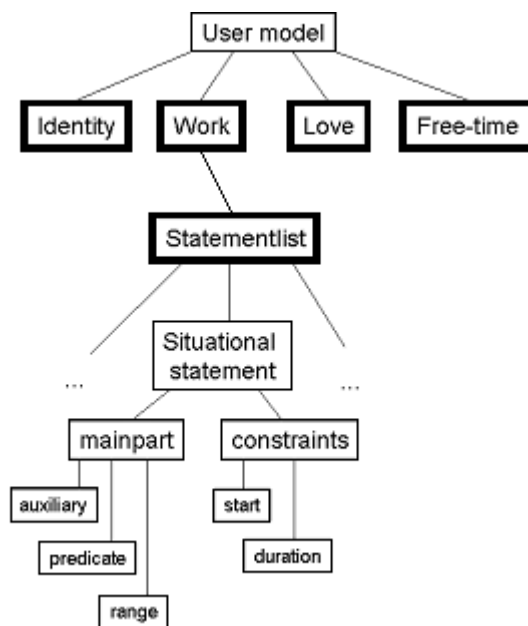
Figuur 2.2 toont een schematische voorstelling van een gebruikersmodel in UserML en GUMO. Het gebruikersmodel is hier een aaneenschakeling van verschillende *situational statements*. Gedetailleerde beschrijvingen van de verschillende onderdelen van een *situational statement* en hun waarde, kan men terugvinden in de reeds vermelde werken en de bijbehorende website.

2.4 Gebruikersprofiel

Het belangrijkste onderdeel van het interactiemodel “Digital Aura” is het gebruikersprofiel. Het gebruikersprofiel bepaalt voor het grootste deel de toepassingen en mogelijkheden die ondersteund worden door het interactiemodel. Het is dus noodzakelijk dat het gebruikersprofiel de nodige informatie bevat om een correcte werking te bekomen van de functionaliteiten die we wensen dat het interactiemodel biedt. Indien de gewenste functionaliteiten bepaald worden, beschikt men over een aantal restricties waaraan het gebruikersprofiel moet voldoen. Met andere woorden, er wordt een doel vastgelegd waar, bij de ontwikkeling van het gebruikersprofiel, naartoe gewerkt kan worden.

In figuur 2.3 wordt een schematische voorstelling van het beoogde gebruikersprofiel getoond. De dunne kaders duiden op concepten die in de reeds ontwikkelde gebruikersprofielen en gebruikersmodellen aan bod kwamen. De dikke kaders daarentegen stellen nieuwe concepten voor. In dit schema is enkel het “Work” concept uitgewerkt. De gelijkaardige concepten “Identity”, “Love” en “Free-time” beschikken over eenzelfde onderliggende structuur dan “Work”. De manier waarop dit gebruikersprofiel ontwikkeld werd en de keuzes die hierbij genomen werden worden beschreven in volgende secties.

De voorbeelden in het werk van Ferscha e.a. [FHM⁺04] in verband met het interactiemodel “Digital Aura”, leggen vooral de nadruk op sociale interactie. Dit type interactie kan plaats vinden tussen personen zelf of tussen een persoon en een systeem. Bij dit laatste is het systeem meer een gedigitaliseerde vorm van een mens dan een systeem dat personen ondersteunt in hun taken. Deze observatie legt een beperking op aan de hoeveelheid en de inhoud van de informatie die beschikbaar moet zijn in het gebruikersprofiel. Dit kan helpen bij het maken van bepaalde keuzes bij de ontwikkeling van een gebruikersprofiel. Verder wordt de ontwikkeling bepaald door een andere belangrijke peiler waarop een goed gebruikersprofiel steunt, namelijk de structuur, en de reeds besproken concepten en keuzes van andere ontwikkelde gebruikersprofielen en gebruikersmodellen.



Figuur 2.3: Gebruikersprofiel voor Digital Aura

2.4.1 Inhoud

Bij het modelleren van personen, is uiteraard persoonlijke informatie van belang. Elementen van deze informatie zijn naam, adres, leeftijd, e-mailadres, enz. Indien sociale interactie beschouwd wordt als hoofdtoepassing, kunnen een aantal categorieën opgesteld worden. In de huidige maatschappij beperkt het sociale contact tussen personen zich dikwijls tot een aantal onderwerpen zoals werk, liefde/relaties, vrije tijd. Op basis van deze onderwerpen kunnen analoge categorieën opgesteld worden binnen het gebruikersprofiel. Elke categorie heeft zijn eigen informatiegegevens. Zo kan de categorie over werk informatie bevatten over welke job een persoon uitvoert, in welk bedrijf hij werkt, enz. Liefde kan informatie bevatten over de huidige status (getrouwd, vrijgezel), de naam van de partner en/of kinderen, welke type persoon hij/zij aantrekkelijk vindt. De categorie vrije tijd ten slotte kan beschrijven welke hobby's een persoon heeft, in welke vereniging hij/zij actief, enz.

Een belangrijke aanvulling bij voorgaande inhoud, is informatie over de betrouwbaarheid van de gegevens. De betrouwbaarheid van bepaalde stukken informatie is van belang bij het verwerken en het maken van conclusies

met betrekking tot de ontvangen informatie. Bepaalde delen van de informatie zijn betrouwbaarder dan andere, en kunnen gedurende een langere periode geldig zijn. Bijvoorbeeld de persoonlijke informatie, zoals naam en geboortjaar is langer geldig dan informatie over een relatie of werk.

De concepten in userML en GUMO kunnen aangewend worden om een notatie te voorzien om de hierboven besproken informatie op te slaan. Weliswaar is er enkel een beperkt deel van de uitgebreide GUMO nodig, en moet deze verder uitgebreid worden om bijvoorbeeld de informatie in de categorie liefde voor te stellen.

2.4.2 Structuur

Door gebruik te maken van userML, wordt de informatie reeds gestructureerd doordat userML een op XML gebaseerde taal is. In userML worden echter alle situational statements achter elkaar geplaatst in het gebruikersmodel. Dit zorgt bijgevolg slechts voor een beperkte structuur. Indien we een goede toegankelijkheid tot het gebruikersprofiel willen, zoals aangegeven in de UM toolkit, is er nood aan een betere structuur. Een goede toegankelijkheid tot het gebruikersprofiel betekent naast een overzichtelijke weergave ook dat er gemakkelijk aanpassingen kunnen gemaakt worden en een nieuw model opgebouwd kan worden.

De informatie kan beter gestructureerd worden door gebruik te maken van een diepere vertakkingsstructuur. Deze vertakkingsstructuur kan bekomen worden door de inhoud in acht te nemen. De inhoud van een gebruikersprofiel voor het interactiemodel kan opgedeeld worden in een aantal categorieën zoals vermeld in de vorige sectie. Op basis van deze indeling kan een overeenkomende structuur binnen het gebruikersprofiel opgebouwd worden, waardoor een diepere vertakkingsstructuur mogelijk wordt. Om dit te realiseren moet het ontwerp en de structuur van userML aangepast worden.

2.5 Conclusie

In dit hoofdstuk is getracht een concept te ontwikkelen voor een gebruikersprofiel dat dienst kan doen binnen het interactiemodel “Digital Aura”. Het concept van dit gebruikersprofiel werd gebouwd op basis van twee belangrijke

peilers, namelijk inhoud en structuur. De inhoud van het gebruikersprofiel is afhankelijk van hetgeen men wenst dat het interactiemodel kan verwezenlijken. De structuur van het gebruikersprofiel speelt een belangrijke rol in de toegankelijkheid van het gebruikersprofiel ten opzichte van de gebruiker.

Naast deze twee peilers, werden reeds ontwikkelde gebruikersmodellen bestudeerd. Enkele belangrijke concepten en keuzes kwamen hierbij aan het licht, en werden toegepast of niet goed genoeg bevonden. UserML en bijhorende GUMO bieden veel potentieel in de zoektocht naar een uniforme notatie voor gebruikersprofielen door het gebruik van situational statements en hun onderverdeling. Ondanks de applicatie-afhankelijke gebruikersprofielen, brengt de UM toolkit enkele zinvolle concepten naar voren, zoals betrouwbaarheidsbewijzen voor de informatie (ook terug te vinden in userML en GUMO) en het belang van toegankelijkheid.

Het resulterende concept is een gebruikersprofiel dat gebaseerd is op de sociale interactie tussen personen. Deze interactie bepaalt de inhoud van het gebruikersmodel (informatiecategorieën voor identiteit, werk, liefde en vrije tijd). Deze inhoud wordt verder uitgebreid met informatie over situatie/context en geïnteresseerde personen omwille van de privacy. Ten slotte vertoont het gebruikersprofiel een diepere vertakkingsstructuur dan die van userML.

Hoofdstuk 3

Geheimhouding

3.1 Inleiding

Geheimhouding of privacy is een belangrijk goed in onze maatschappij. In tal van domeinen wordt aandacht besteed aan het voorzien en bewaren van geheimhouding. De bankwereld, bedrijven en internet zijn goede voorbeelden van zulke domeinen waar de nood aan geheimhouding groot is. Nu “Ubiquitous Computing” zijn intrede doet en informatie over personen of systemen bijna onmerkbaar uitgewisseld worden, begint ook in dit domein de nood aan geheimhouding zich op te dringen.

De verschillende sensoren en “embedded systems”, horende bij “Ubiquitous Computing”, zullen de activiteiten van personen en/of andere systemen waarnemen. Hierbij wordt ook bijkomende informatie vergaard die al dan niet bewaard zal blijven. Niet iedereen of niet elk systeem wenst alle informatie beschikbaar te maken voor andere personen of systemen. Ze moeten zelf kunnen beslissen welke informatie ze wensen af te staan. Welke informatie ze wensen af te staan, of anders gezegd, wat geheimhouding precies inhoudt, is afhankelijk van persoon tot persoon of van systeem tot systeem. Zoals Westin reeds aanhaalde, beleeft iedereen geheimhouding op een andere manier:

No definition of privacy ... is possible, because those issues are fundamentally matters of values, interests and power. (Westin 1967 quoted in [HNLL04])

Ook al is er geen éénduidige definitie voor geheimhouding voor handen, kan er toch beroep gedaan worden op onderzoek dat zich reeds toespitste op geheimhouding in het algemeen of specifiek gericht op “Ubiquitous Computing”. In de volgende sectie zal dit werk worden besproken, waarop in de volgende secties aan de hand van dit werk specifiek gezocht wordt naar geheimhoudingstechnieken voor het interactiemodel “Digital Aura”.

Geheimhouding moet tenslotte de nodige aandacht krijgen bij de ontwikkeling van een applicatie omdat in vele gevallen deze staat of valt met zijn mogelijkheden tot geheimhouding. Geheimhouding mag anderzijds ook niet te complex worden. Een ingewikkeld systeem voor geheimhouding kan zelfs leiden tot een negatief effect, waarbij geheimhouding verwaarloosd wordt door de gebruiker. Er moet dus een geschikt compromis gevonden worden om tot een efficiënte en bruikbare geheimhouding te komen.

3.2 Gerelateerd werk

Een goede basis voor geheimhouding in dit interactiemodel, kan men terugvinden in de richtlijnen voor “fair information practice” [LLCL98]. Een commissie werd in het leven geroepen om een oplossing te zoeken voor geheimhouding in de “on-line” wereld. Het resultaat bestaat uit een vijftal principes of richtlijnen:

- “Notice/Awareness”
Een consument moet geïnformeerd worden over welke informatie opgevraagd wordt, door wie en waarvoor deze informatie moet dienen.
- “Choice/Consent”
De consument moet de keuze hebben over of zijn toestemming geven voor het gebruik van de informatie.
- “Access/Participation”
De consument heeft het recht om de informatie te zien en eventueel te veranderen.
- “Integrity/Security”
De informatie moet correct zijn en moet veilig beheerd worden.

- “Enforcement/Redress”

Er moet een mechanisme zijn dat de voorgaande richtlijnen toepast.

Op basis van de richtlijnen van “fair information practice”, analyse van systemen met betrekking tot geheimhouding en ervaring met het ontwikkelen van eigen geheimhoudingsapplicaties, hebben Lederer e.a. een leidraad ontwikkeld voor ontwikkelaars van applicaties die geheimhouding vereisen [LHDL04]. Deze leidraad bevat vijf valkuilen, die best vermeden worden om een degelijke geheimhouding te voorzien binnen een applicatie. De vijf valkuilen worden meer in detail besproken in sectie 3.4.

Een ander belangrijk aspect is dat de gebruiker een duidelijk en concreet beeld heeft van de mogelijke risico’s, hij/zij deze risico’s begrijpt en hij/zij deze risico’s kan plaatsen met betrekking tot de applicatie waarvoor men geheimhouding wenst. Een methode om zulk beeld te schetsen, kan men terugvinden in een risicomodel dat opgesteld werd door Lederer e.a. [HNLL04]. Dit risicomodel wordt gehanteerd en besproken in sectie 3.3 met betrekking tot “Digital Aura”.

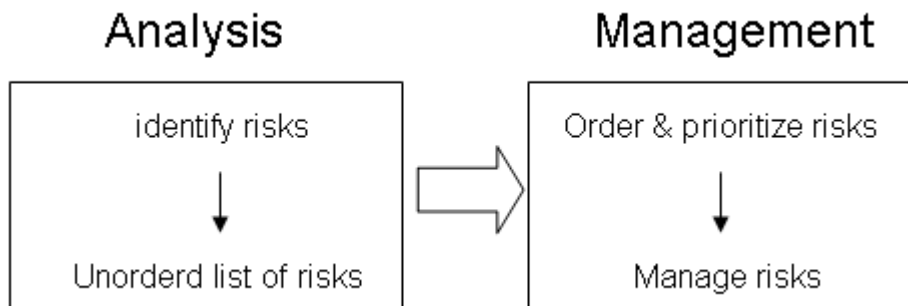
Ten slotte wordt geheimhouding in het algemeen grotendeels bepaald door 2 aspecten: wie wenst de informatie te verkrijgen en wat is de situatie tijdens het opvragen van de informatie? Een voorbeeldscenario dat beide aspecten bevat, is een werknemer die enkel tijdens de werkuren zijn informatie (plaats, bezigheid) deelt met zijn werkgever, maar gedurende de hele dag zijn informatie deelt met zijn echtgenote/vriendin. Deze twee aspecten bepalen de nauwkeurigheid van de informatie die vrijgegeven wordt [LDM02]. De relatieve belangrijkheid tussen beide aspecten, werd eveneens bestudeerd door Lederer [LMD03]. Het toont aan dat personen meer belang hechten aan wie de informatie opvraagt, dan aan de situatie waarin de informatie opgevraagd wordt.

3.3 Risicomodel

Om een applicatie van goede geheimhouding te voorzien, is het belangrijk dat de ontwikkelaars van de applicatie over een goed en duidelijk inzicht beschikken met betrekking tot de geheimhoudingsrisico’s. Aan de hand van

dit inzicht kunnen ze de risico's, voor de specifieke applicatie die ze aan het ontwikkelen zijn, identificeren, begrijpen en er prioriteiten aan toekennen. Op deze manier zijn de ontwikkelaars misschien niet in staat om perfecte geheimhouding te ontwikkelen, maar kunnen ze toch een aanvaardbare geheimhouding voorzien voor de eindgebruikers.

Het verwerven van zulk inzicht in de geheimhoudingsrisico's, kan bekomen worden met een risicomodel. Een geschikt risicomodel dat ontwikkeld werd en gebruikt kan worden in het specifieke domein van "Ubiquitous Computing", is dat van Lederer e.a. [HNLL04], zoals reeds aangehaald in de vorige sectie. Dit risicomodel bestaat uit twee onderdelen: *privacy risk analysis* en *privacy risk management*. In figuur 3.1 wordt een schematisch overzicht getoond van de werking van het risicomodel.



Figuur 3.1: Schematisch overzicht van risicomodel

3.3.1 Analyse

Het eerste deel van dit risicomodel bestaat uit een reeks vragen. Deze vragen hebben als doel de ontwikkelaar te doen nadenken over de intentie en technologische basis van de applicatie, en de bijhorende geheimhouding. De vragen zijn hierbij opgesplitst in twee groepen: *social and organizational context* en *technology*. De eerste groep vragen behandelt zaken zoals wie de gebruikers zijn, in welke omstandigheden de applicatie gebruikt wordt, de relaties tussen datahouders en dataobserveerders, enz. De tweede groep vragen richt zich specifiek op technologische aspecten zoals hoe de informatie

vergaard wordt, of informatie discreet of continu uitgewisseld wordt, of informatie opgevraagd (“pull”) of zelf uitgezonden (“push”) wordt, enz. Het uiteindelijke resultaat van deze vragenlijst bestaat uit een aantal mogelijke ongeordende risico’s.

Deze analyse kan toegepast worden op het interactiemodel “Digital Aura”. Door een aantal vragen te beantwoorden verkrijgt men een aantal risico’s die gepaard gaan met dit interactiemodel. Hierbij zullen de vragen niet expliciet vermeld worden, maar zullen ze wel duidelijk afgeleid kunnen worden uit de antwoorden.

De gebruikers van een applicatie, op basis van het voorgestelde interactiemodel, zijn typische gebruikers of systemen die stukken persoonlijke informatie willen delen met anderen, of op zoek zijn naar specifieke eigenschappen van andere personen of systemen. In het algemeen is er geen sprake van een specifieke relatie tussen de datahouder en de dataobserveerder, waardoor de relatie eerder bestempeld wordt met een laag vertrouwensgehalte. Tussen de dataobserveerders kunnen bijgevolg ook “slechte” personen of systemen zitten die de informatie willen misbruiken. De informatie zelf kan verschillende domeinen beslaan, zoals bijvoorbeeld locatie, hobby’s, gezichtskenmerken, enz en wordt op discrete tijdstippen uitgewisseld. Het uitwisselen van de informatie gebeurt via een draadloos communicatiemedium, waarbij de datahouder meestal niet op de hoogte gebracht wordt wanneer een uitwisseling van data plaatsvindt. Het gebruik van de applicatie ten slotte, vindt plaats in normale dagelijkse omstandigheden, al dan niet in openbare plaatsen.

Aan de hand van deze analyse kunnen een tweetal risico’s waargenomen worden. Een eerste risico is de mogelijkheid dat informatie wordt opgevangen door systemen of personen met slechte bedoelingen. Deze personen of systemen kunnen misbruik maken van de informatie, bijvoorbeeld door deze te verkopen aan reclamebedrijven of door de gebruiker in gevaar te brengen bijvoorbeeld door stalking. Een tweede risico bestaat uit de kans dat er meer informatie wordt uitgewisseld dan gewenst is. Dit risico wordt eveneens sterk bepaald door de ontvanger van deze informatie.

3.3.2 Management

Het doel van het tweede deel van het risicomodel is het resultaat afkomstig van de *analyse*, de lijst met mogelijke risico's, te ordenen en er prioriteiten aan toe te kennen. Dit kan de ontwikkelaars helpen oplossingen te vinden, naar de eindgebruiker toe, om zo de risico's aan te pakken. Prioriteiten worden bepaald aan de hand van drie factoren die overgenomen werden van de kosten-baten analyse van Hand, die werd overgenomen door Lederer:

- de kans L dat ongewenste informatie vrijgegeven wordt
- de schade D bij ongewenst vrijgeven
- de kost C om voor geheimhouding te zorgen

Voor elk risico wordt aan deze factoren een waarde toegekend waarna men deze factoren met elkaar kan vergelijken. In het algemeen wordt de factor C vergeleken met het product van de factoren L en D . Indien bijvoorbeeld de kost om geheimhouding te voorzien kleiner is dan de kost van de schade bij het ongewenste vrijgeven van informatie, met andere woorden $C < LD$, geeft dit aanleiding om geheimhouding te implementeren. De factoren dienen niet exact toegekend te worden maar vereisen eerder dat de ontwikkelaars nadenken over de risico's en mogelijke bijhorende gevaren bij ongewenst vrijgeven van de informatie.

Na het rangschikken van de risico's en het toekennen van de prioriteiten, wordt in dit deel ook getracht deze risico's te beheren. Net als in de *analyse* gebeurt dit aan de hand van een reeks vragen. Deze vragen kunnen handelen over hoe ongewenste informatie wordt vrijgegeven, of er controle- of feedbackmechanismen voorzien zijn, welke de standaardinstellingen zijn, enz. Op basis van deze vragen kan de ontwikkelaar meer inzicht vergaren in de tekortkomingen van de applicatie of de structuur.

Een eerste stap in dit tweede deel van het risicomodel, is het toekennen van waarden aan de factoren bij de verschillende risico's uit het *analyse* gedeelte. Uit vorige sectie blijken deze risico's het ontvangen van informatie door personen of systemen met slechte bedoelingen of het ongewenst vrijgeven van te veel informatie. Bij het eerste risico is de kans, L , dat het gebeurt hoog aangezien iedereen informatie kan ontvangen. De schade achteraf kan

variëren tussen schaamte tot het in gevaar zijn van de gebruiker. Bijgevolg kennen we aan de schadefactor, D , een hoge waarde toe. Het werk dat vereist is om geheimhouding te voorzien, C , is niet gering, maar het vereist ook geen maandenlange inspanningen in het implementatieproces. De factor C kan dus een gemiddelde waarde toegekend krijgen. Bij het vergelijken van deze factoren blijkt dat kans dat het gebeurt en de schade die erdoor opgelopen wordt, sterk de kost van het voorzien van geheimhouding overtreft. Dit leert ons dat functionaliteiten voor geheimhouding, en dus het beperken of zelfs wegwerken van dit risico sterk aanbevolen is. Aangezien het tweede risico sterk gerelateerd is aan het zojuist besproken risico, kunnen overeenkomstige conclusies getrokken worden voor dit tweede risico.

De tweede stap van het management van risico bestaat, zoals reeds aangehaald, uit een aantal vragen. Net als in de vorige sectie zullen enkel de antwoorden geformuleerd worden waaruit de vragen opgemaakt kunnen worden. De antwoorden zijn gebaseerd op de implementatie van dit interactiemodel, zie hoofdstuk 5.

De gebruikers van de applicatie hebben de mogelijkheid om het vrijgeven of ontvangen van informatie in of uit te schakelen. Wanneer het vrijgeven van informatie is ingeschakeld, is de keuze van de gebruiker uitgebreid met de mogelijkheid om te specificeren welke details van de informatie vrijgegeven mogen worden. Hierbij zijn standaardinstellingen voorzien die ervoor zorgen dat er geen informatie wordt vrijgegeven. Indien er toch informatie wordt vrijgegeven, heeft de gebruiker de keuze om geïnformeerd te worden bij elke vrijgave. Ook wordt een geschiedenis bijgehouden van elke uitwisseling waardoor de gebruiker kan leren uit mogelijk voorgaande problemen.

3.4 5 Valkuilen

Hoewel, zoals reeds aangehaald, geheimhouding belangrijk is in deze maatschappij, kan men helaas nog steeds vaststellen dat hieraan niet altijd voldoende aandacht wordt besteed in de wereld van de informatica. Bijgevolg voldoen een aantal applicaties niet aan de vereisten voor geheimhouding. Daarom werden een vijftal valkuilen gegeven door Lederer e.a. [LHDL04] (ter ondersteuning van de ontwikkelaar), die zeker vermeden moeten worden

bij de ontwikkeling van een applicatie. Hierdoor kunnen de moeilijkheden in dit domein overkomen worden. Deze valkuilen zijn *obscuring potential information flow*, *obscuring actual information flow*, *emphasizing configuration over action*, *lacking coarse-grained control* en *inhibiting established practice*. Deze valkuilen werden onderverdeeld in twee groepen. De eerste groep bestaat uit de eerste twee valkuilen, en richt zich op het begrip dat de gebruiker heeft over gevolgen van het systeem met betrekking tot geheimhouding. De overige valkuilen, die de tweede groep vormen, bespreken de acties van de gebruiker. In praktisch opzicht lenen sommige valkuilen zich beter of uitsluitend tot het gebruikersmodel, terwijl de anderen beter of enkel geschikt zijn voor de applicatie in het algemeen. De concrete realisatie van mogelijkheden voor het vermijden van de eerste groep valkuilen met betrekking tot het gebruikersmodel worden besproken in de volgende secties. Voor het vermijden van de valkuilen die invloed hebben op de applicatie, wordt verwezen naar sectie 5.4.

Een eerste valkuil, *obscuring potential information flow*, wijst op het feit dat het bereik of de gevolgen van de verschillende geheimhoudingsmogelijkheden duidelijk moeten zijn voor de gebruiker. Deze valkuil leent zich beter voor het gebruikersmodel. Concreet betekent dit voor het gebruikersmodel dat bij een bepaalde geheimhouding, het duidelijk moet zijn welke stukken informatie vrijgegeven of welke achtergehouden moeten worden. Een mogelijke aanpak hiervoor is ieder stuk informatie te voorzien van een bepaald attribuut dat de precieze geheimhouding hiervoor aangeeft.

Obscuring actual information flow is een valkuil die beter geschikt is bij de ontwikkeling van de applicatie. Deze valkuil zegt dat de applicatie duidelijk moet maken aan de gebruiker welke geheimhouding er voor het moment geldig is. Dit wil zeggen dat wanneer informatie uitgewisseld wordt, de gebruiker weet welke informatie aan wie wordt vrijgegeven. De richtlijn heeft het bijkomende voordeel dat gebruikers hun mogelijke fouten weer ongedaan kunnen maken.

De derde valkuil die vooral van toepassing is op een gebruikersmodel, is *emphasizing configuration over action*. Bij het aanmaken of onderhouden van een gebruikersmodel, mag geen excessieve configuratie van de gebruiker

verwacht worden om tot geheimhouding te komen. Geheimhouding moet eenvoudig en natuurlijk aangebracht kunnen worden.

Een applicatie dat een goede geheimhouding wil ondersteunen, moet beschikken over de mogelijkheid om via een top-level mechanisme deze geheimhouding op eenvoudige wijze in of uit te schakelen. Deze vereiste vormt de derde valkuil *lacking coarse-grained control*. Naast het in- of uitschakelen behoren ook meerdere niveaus van geheimhouding tot de mogelijkheden. Met een top-level mechanisme wordt bedoeld dat de gebruiker via een simpel commando, bijvoorbeeld op een knop of toets drukken, het niveau van geheimhouding kan bepalen. Het duidt dus niet zozeer op welke niveau's mogelijk moeten zijn. Uiteraard is zulk top-level mechanisme enkel van toepassing binnen de applicatie en niet binnen het gebruikersmodel.

Ten slotte beschrijft de valkuil *inhibiting established practice* dat designers op de hoogte moeten zijn van sociale praktijken in verband met geheimhouding. Mensen gebruiken soms privacy door middel van een aantal genuanceerde praktijken. Een voorbeeld hiervan is een gsm die uitgeschakeld wordt wanneer een persoon niet wenst gestoord te worden. De persoon die hem dan oproept krijgt een antwoordapparaat of dergelijke te horen, maar kan wel weten wat deze persoon aan het doen is. Ook deze richtlijn leent zich beter tot de applicatie zelf.

3.5 Geheimhoudingsprofiel

In sectie 2 is het gebruikersmodel besproken, waarbij de aandacht vooral gericht was op het gebruikersprofiel. Om het gebruikersmodel te vervolledigen, moet ook een geheimhoudingsprofiel er onderdeel van uitmaken. Een geheimhoudingsprofiel beschrijft wat de gebruiker onder geheimhouding verstaat, met andere woorden welke informatie een gebruiker aan wie wil vrijgeven en onder welke omstandigheden. Hieronder zal getracht worden om een geheimhoudingsprofiel dat gebruikt kan worden in het interactiemodel van "Digital aura" te ontwikkelen. Net als bij de ontwikkeling van het gebruikersprofiel, zullen we eerst een blik werpen de wijze waarop zulke geheimhouding beschreven werd in de UMToolkit en UserML.

3.5.1 UMToolkit en UserML

In de UMToolkit van Judy Kay is geen enkele aanwezigheid van geheimhouding merkbaar. In de voorgestelde en/of gebruikte scenario's is privacy geen echte prioriteit, en werd het daarom waarschijnlijk niet behandeld. Een eventueel andere reden is dat het voorgestelde werk niet recent is: in die periode was "Ubiquitous Computing" nog niet voldoende ontwikkeld en bijgevolg was de noodzaak van privacy hierbij niet of onvoldoende duidelijk.

In UserML en Gumo is er wel sprake van geheimhouding, hoewel dit slechts in beperkte mate aanwezig is. In UserML bevatten "situational statements" een "privacy"-onderdeel. Dit onderdeel geeft aan wie de eigenaar van het stuk informatie en voorziet voor de gebruiker een mogelijkheid om te bepalen welke toegangsrechten dit stuk informatie moet hebben en wat het doel van deze informatie is. De geheimhouding hier is dus redelijk beperkt: de toegang bijvoorbeeld van de informatie is voor iedereen hetzelfde en er is geen mogelijkheid tot het specificeren van de situatie. De geheimhouding in UserML en GUMO is hierdoor niet geschikt om gebruikt te worden binnen dit interactiemodel.

3.5.2 Profiel voor Digital Aura

De belangrijke parameters in geheimhouding zijn, zoals reeds aangehaald, de personen die de informatie opvragen en de situatie waarin men zich bevindt. Dit laatste kan gemakkelijk voorzien worden in het geheimhoudingsprofiel met behulp van userML en GUMO. Deze bieden namelijk de mogelijkheid tot het specificeren van spatiale en temporele data door gebruik van de zogenaamde "Predicates". Met behulp van deze spatiale en temporele data kan de gebruiker een situatie definiëren binnen het geheimhoudingsprofiel. Wanneer informatie wordt vrijgegeven bij de interactie tussen personen kan zo een situatie bepalen welke informatie daadwerkelijk vrijgegeven mag worden en welke niet.

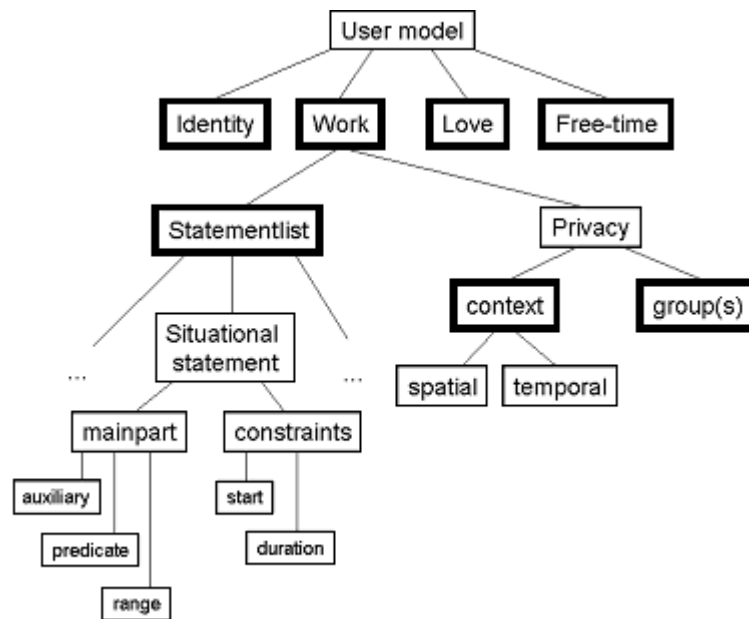
Mogelijkheden om bepaalde delen van het geheimhoudingsprofiel te linken aan de personen die informatie kunnen opvragen zijn er, zoals vermeld in vorige sectie, niet bij userML en GUMO. Bijgevolg moet dit voorzien worden. Een manier om dit probleem op te lossen, bestaat erin personen te groeperen en stukken informatie te linken aan een bepaalde groep of zelfs meer-

dere groepen. Deze methode wordt ook aangewend binnen het framework “The Precision Dial”, dat ontwikkeld werd door Scott Lederer in [Led03]. De doelstelling van deze methode is dat de gebruiker verschillende groepen aanmaakt en in elke groep een aantal personen plaatst. Wanneer een ondervrager tijdens de interactie informatie opvraagt van een andere persoon zal de ondervrager enkel informatie ontvangen die overeenkomt met de groep waarin deze ondervrager werd geplaatst door de andere persoon.

Goede geheimhouding met betrekking tot het gebruikersmodel kan bekomen worden door rekening te houden met de twee valkuilen, die besproken werden in sectie 3.4. Een goede stap om de eerste valkuil, *obscuring potential information flow*, die betrekking heeft tot het gebruikersmodel, te vermijden, volgt uit de keuze van een goede structuur. Een goede structuur biedt een goede toegankelijkheid. Hierdoor krijgt de gebruiker een beter inzicht in het gebruikersmodel, en bijgevolg ook in de geheimhouding die erin verwerkt zit.

Daarnaast biedt een goede structuur ook de mogelijkheid om geheimhouding op een algemene wijze te specificeren, in plaats van per informatiegegeven aan te geven wat de geheimhouding moet zijn. Ook door het gebruik van groepen, zoals hierboven aangegeven, kunnen bepaalde delen van de informatie op eenvoudige wijze gekoppeld worden aan bepaalde personen. Aan de tweede richtlijn voor het gebruikersmodel, *emphasizing configuration over action*, kan op deze manier ook voldaan worden opdat het configureren van de geheimhouding sterk beperkt wordt.

In figuur 3.2 werd het reeds ontwikkeld gebruikersprofiel, beschreven in het vorige hoofdstuk, verder uitgebreid met het geheimhoudingsprofiel. Beide delen vormen bijgevolg samen het gebruikersmodel. In het geheimhoudingsprofiel wordt dezelfde structuur behouden als in het gebruikersprofiel. Per component (Identity, Work, Love, FreeTime) kunnen een aantal geheimhoudingsonderdelen toegevoegd worden. Deze geheimhoudingsonderdelen bevatten spatiale en temporele onderdelen voor het specificeren van de situatie, en een aantal groepen die op hun beurt een aantal personen bevatten voor het specificeren van wie de informatie mag ontvangen. Deze structuur laat voldoende vrijheid toe voor het specificeren van een bepaalde geheimhouding per component, alsook het eenvoudig herbruiken van groepen en situaties waardoor de configuratie niet te ingewikkeld wordt.



Figuur 3.2: Gebruikersmodel voor Digital Aura

3.6 Conclusie

Geheimhouding is een belangrijk aspect binnen het interactiemodel “Digital aura” en dient voldoende aandacht te krijgen. Om deze reden werd in dit hoofdstuk onderzoek verricht naar manieren om een goede geheimhouding te verkrijgen. Aan de hand van een risicomodel werden de belangrijkste risico’s in dit interactiemodel aangehaald. Deze risico’s werden vervolgens bestudeerd en gewogen. Uit het risicomodel kan vastgesteld worden dat een implementatie voor het vermijden van de risico’s minder kostelijk is dan de schade die zulke risico’s met zich mee kunnen brengen.

Naast de identificatie van de risico’s werden ook een vijftal valkuilen geïdentificeerd die vermeden moeten worden bij de ontwikkeling van een applicatie. Het vermijden van deze valkuilen biedt geen garantie op een sluitende geheimhouding, maar biedt wel een uitstekende basis om tot goede geheimhouding te komen. Deze valkuilen hebben zowel betrekking op het geheimhoudingsprofiel als op de applicatie zelf.

Ten slotte werd in dit hoofdstuk een geheimhoudingsprofiel ontwikkeld om het gebruikersmodel te vervolledigen. Het doel van dit geheimhoudingsprofiel is de gebruiker een mogelijkheid te bieden om het gedeelte van de informatie te specificeren dat vrijgegeven mag worden in welke situaties en aan wie. De situatie bestaat uit temporele en spatiale data, terwijl de personen aan wie informatie vrijgegeven mag worden aangeduid kunnen worden door het gebruik van groepen.

Hoofdstuk 4

Persoonlijke Agent

4.1 Inleiding

In een “actieve” omgeving, zoals bij “Ubiquitous Computing” waarbij computers ons omgeven, wordt informatie uitgewisseld. Indien hierbij interactie-modellen zoals “Digital Aura” gehanteerd worden, is het niet altijd duidelijk voor de gebruiker wanneer uitwisselingen van informatie plaatsvinden, en is het nog minder duidelijk waaruit deze informatie bestaat. Bij uitwisseling van informatie die mogelijk interessant is voor de gebruiker, moet de gebruiker hiervan op de hoogte gebracht worden zodat deze de nodige stappen kan ondernemen.

Een mogelijke methode om de gebruiker op de hoogte te houden is het melden van uitwisselingen door middel van meldingen binnen de applicatie, bijvoorbeeld door een tekstberichtje of een geluid. Deze eenvoudige wijze kan uitstekend dienst doen wanneer uitwisselingen zelden of weinig voorkomen. De gebruiker wordt hierbij steeds gewaarschuwd bij een uitwisseling en kan zelf bepalen of hij/zij de informatie kan gebruiken of niet.

Wanneer echter uitwisselingen vaak of constant gebeuren, schiet bovenstaande methode echter tekort. De gebruiker wordt dan namelijk overspoeld door meldingen. De gebruiker moet bijgevolg veel tijd besteden aan het afhandelen van deze meldingen, waardoor hij/zij zich niet meer kan bezig houden met zijn/haar werk. Een ander gevolg kan zijn dat de gebruiker na verloop van tijd de meldingen gaat negeren en hierdoor misschien interessante en/of belangrijke informatie aan zijn zicht ontsnapt. Een andere methode

om de gebruiker op de hoogte te houden van uitwisseling van interessante informatie dringt zich op.

Een oplossing tot bovenstaand probleem kan gevonden worden in het gebruik van “persoonlijke agenten”. Deze “persoonlijke agenten” zijn eigenlijk stukjes software, waardoor ze ook bekend zijn als “software agents”. Het doel van deze agenten is de gebruiker te ontlasten van een aantal taken. Opdat de agenten deze taken naar behoren kunnen uitvoeren moeten ze in staat zijn beslissingen te nemen. De achterliggende reden voor het nemen van een beslissing, in naam van de gebruiker, is hierbij gebaseerd op de kennis die de agent heeft van de gebruiker, zoals bijvoorbeeld voorkeuren of karakteristieken. Een meer gedetailleerde beschrijving van een agent kan men terugvinden in sectie 4.3.

Men zou deze agenten bijgevolg kunnen gebruiken om in naam van de gebruiker de uitwisseling van informatie bij “Ubiquitous Computing” in de gaten te houden. Zonder tussenkomst van de gebruiker kan deze agent de informatie die gepaard gaat met de uitwisseling controleren. Doordat de agent op de hoogte is van de interesses van de gebruiker, kan deze gepast reageren, bijvoorbeeld door zelf een actie te ondernemen of door de gebruiker een melding te laten zien wanneer er interessante informatie uitgewisseld werd. Hierdoor kan de gebruiker ongestoord zijn werk verder zetten en toch steeds op de hoogte blijven bij speciale gebeurtenissen.

In dit hoofdstuk wordt van start gegaan met een overzicht van reeds bestaande studies en werk met betrekking tot agenten. Vervolgens wordt een duidelijke omschrijving gegeven van wat agenten zijn en wat ze inhouden. De definitie van een agent kan verder gebruikt worden om een topologie van de agenten op te stellen. Hiermee is het mogelijk om de agenten in te delen in verschillende klassen. Ten slotte wordt op basis van de definitie, een agent gedefinieerd die dienst kan doen binnen het interactiemodel “Digital Aura”.

4.2 Gerelateerd Werk

Zoals reeds aangehaald, bestaat de taak van een agent eruit de gebruiker van bepaalde taken te ontlasten. Agenten beschikken hierdoor over een groot

potentieel. Het is daarom niet verwonderlijk dat het gebruik van deze agenten reeds uitvoerig bestudeerd en toegepast is. In deze sectie zal daarom een kort overzicht gegeven worden over reeds ontwikkelde agenten en hun toepassingen binnen een aantal domeinen.

Een eerste en tevens belangrijk domein waarin agenten bestudeerd en ontwikkeld werden, is het domein van de artificiële intelligentie (AI). Een toepassing van agenten binnen dit domein kan men terugvinden in [SL01]. In dit werk van Shearin en Lieberman wordt een agent gebruikt om de voorkeuren van de gebruiker te leren bij het zoeken naar woningen of appartementen. Op basis van eerdere zoektochten met bepaalde criteria, leert deze agent naar wat de gebruiker op zoek is. Het profiel dat de agent voor de gebruiker opmaakt, kan deze agent in volgende zoektochten gebruiken zodat de gebruiker niet steeds opnieuw alle criteria handmatig moet ingeven. Een andere toepassing in het domein van AI, is het gebruik van agenten om het gedrag van mensen, robots, enz. na te bootsen. Zulke agenten werden gebruikt in de “Lord of the rings” films [MNLP]. Hierbij werden agenten ingeschakeld om grote legers tot stand te brengen die autonoom reageren. Een ander voorbeeld van zulke agenten kan men terugvinden in [JT98]. In dit werk worden agenten gebruikt om het dierlijk gedrag te simuleren.

Op andere gebieden kan het gebruik van agenten eveneens nuttig zijn. Zo worden agenten gebruikt bij datamining, bijvoorbeeld in [GKS03] om patronen of trends op te sporen in grote hoeveelheden informatie. Verder kunnen ook agenten aangetroffen worden in het domein van het internet. In dit domein worden agenten in verschillende takken gebruikt, gaande van spamfilters [PO05], voor het verwijderen van ongewenste e-mails, tot zoekrobots [LSMG⁺00], voor het vinden van de gewenste informatie.

In het domein van “Ubiquitous Computing” ten slotte, is het gebruik van agenten ook reeds bestudeerd en heeft men een aantal toepassingen ontwikkeld. De “Digital Me” (D-ME) agent uit het werk van De Carolis e.a. [CPP03, CCP04a], is een agent die de gebruiker representeert in een actieve omgeving. De D-Me agent interageert met andere agenten in de omgeving, en bepaalt welke taken een gebruiker kan uitvoeren in deze omgeving aan de hand van de context en de noden van de gebruiker. Een gebruiker kan bijvoorbeeld op voorhand een aantal taken ingeven, waaraan de agent

hem/haar herinnert indien deze taken uitgevoerd kunnen worden. Eenmaal de taken ingegeven zijn, hoeft de gebruiker zichzelf er niet steeds meer aan te herinneren dat hij/zij nog bepaalde dingen te doen heeft. Een ander gebruik van agenten in “Ubiquitous Computing”, eveneens van De Carolis e.a. [CCPC04b, CCP04b], is een toepassing waarbij gebruikers, voorzien van een gebruikersprofiel, interageren met een digitale reisgids. De digitale reisgids bevat een agent die, op basis van de omgeving en de informatie uit het gebruikersprofiel, de communicatie door middel van natuurlijke taal, verzorgt met de gebruiker. Hierbij beslist de agent welke informatie aan de gebruiker getoond wordt en op welke manier dit gebeurt. Naargelang het antwoord van de gebruiker, past de agent zijn “doelen” aan zodat er steeds gepersonaliseerde informatie ter beschikking is.

4.3 Definitie

Zoals reeds vermeld, is een agent een stukje software dat bepaalde taken op zich neemt. Deze taken kunnen taken zijn die in naam van de gebruiker uitgevoerd worden, of taken binnen een applicatie zelf. Welke taken precies door de agent worden uitgevoerd, is afhankelijk van het doel van de applicatie of hetgeen de gebruiker verlangt. Doordat het gebruik van de agenten of de taken die ze moeten uitvoeren zeer uiteenlopend kunnen zijn, is er geen eenduidige definitie van een agent beschikbaar. Wel kunnen agenten beschreven worden aan de hand van een aantal kenmerken of eigenschappen. Enkele belangrijke eigenschappen en kenmerken, die veelvuldig voorkomen in andere werken [Wik, Fon93, Bel97], zijn:

- *Persistentie*:
De software die de agent voorstelt, wordt voortdurend uitgevoerd. Tijdens deze uitvoering wordt in de code beslist of er verdere activiteit wordt ondernomen of dat er taken uitgevoerd moeten worden.
- *Autonomie*:
De autonomie van een agent betekent dat een agent op zichzelf kan opereren zonder tussenkomst van de gebruiker. Dit autonoom opereren bestaat erin doelen op te stellen, zelf beslissingen te nemen met betrekking tot de uitvoering van taken, prioriteiten op te stellen voor de verschillende taken, enz. Opdat een agent dit kan verwezenlijken

moet er sprake zijn van “proactiveness”, dit wil zeggen dat een agent initiatieven neemt in plaats van te reageren op gebeurtenissen uit de omgeving.

- *Leren/Intelligentie:*

Om goede beslissingen te kunnen nemen ten opzichte van de gebruiker, moet een agent over enige intelligentie beschikken. Intelligentie bestaat hier voornamelijk uit kennis van de voorkeuren/karakteristieken van de gebruiker/omgeving, en het goed hanteren van deze kennis voor het nemen van de beslissingen. Het is echter niet altijd mogelijk om de voorkeuren/karakteristieken van een gebruiker/omgeving op voorhand in te schatten. Daarom is het soms vereist dat een agent deze voorkeuren/karakteristieken kan leren tijdens de uitvoering. Het leren heeft het bijkomende voordeel dat, indien de voorkeuren/karakteristieken plots wijzigen, de agent zijn gedrag kan aanpassen.

- *Samenwerking:*

Een belangrijke eigenschap waarover een agent moet beschikken is een mogelijkheid tot samenwerking. Deze samenwerking kan zowel plaats vinden tussen agenten onderling, als tussen een agent en de gebruiker. Samenwerking tussen agenten kan leiden tot een groter geheel, waarbij elke agent een specifieke taak vervult in dit geheel. Samenwerking kan ook simpelweg uitwisseling van informatie beslaan. Samenwerking tussen een agent en een gebruiker daarentegen, is tweeledig. Ten eerste is het belangrijk voor de gebruiker dat hij/zij zijn voorkeuren/karakteristieken bekend kan maken aan de agent, zodat deze de juiste stappen onderneemt tijdens zijn uitvoering. Anderzijds moet een gebruiker op de hoogte gebracht worden van de uitgevoerde taken, en eventueel bijhorende resultaten van de agent. In principe bestaat de samenwerking tussen gebruiker en agent uit het opvragen en voorzien van de nodige informatie tussen beide.

- *Mobiliteit:*

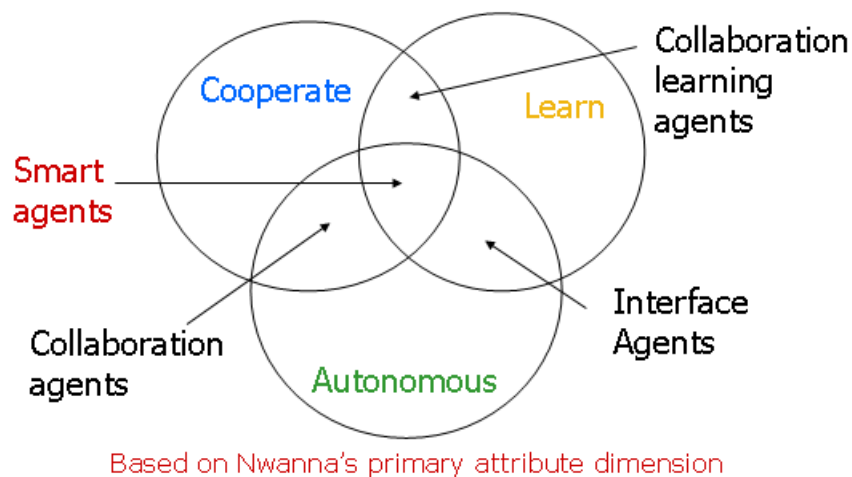
Indien een agent over deze eigenschap beschikt, kan deze agent zich verplaatsen over het netwerk. Deze eigenschap kan zeer nuttig zijn in de mobiele, dagelijkse wereld waarbij de gebruiker bijna overal aanspraak kan maken op het gebruik van een agent. Indien een agent zich niet kan verplaatsen over het netwerk, wordt deze een “statische” agent

genoemd.

Een agent voldoet niet altijd aan alle hierboven vermelde eigenschappen. Veel kenmerken of eigenschappen waarover een agent beschikt, zijn afhankelijk van het doel waarvoor, of het domein waarin de agent gebruikt wordt. Aan de hand van de domeinafhankelijkheid kunnen ook nog tal van andere eigenschappen of kenmerken toegeschreven worden aan een agent. Voorbeelden van deze kenmerken zijn personaliseerbaarheid, redenerend of stimulus/responsie, veiligheid, enz. Bovenstaande eigenschappen beschrijven bijgevolg niet exact wat een agent is, maar geven toch een goed beeld over hetgeen men van een agent kan verwachten.

4.4 Topologie

Op basis van de eigenschappen die beschreven werden in de vorige sectie, werd een topologie¹ opgesteld door Nwana [Nwa96]. De ontwikkelde topologie werd gebaseerd op de drie eigenschappen: *Autonomie*, *Leren* en *Samenwerking*. Een schematisch overzicht hiervan is te zien in Figuur 4.1.



Figuur 4.1: Nwana's topologie van software agents

¹een verzameling van eigenschappen op basis waarvan een indeling gemaakt kan worden.

Deze topologie is niet bedoeld exact te zijn, maar moet een grove classificatie van de agenten mogelijk maken. De gekozen eigenschappen waarop deze topologie gesteund is, zijn de elementaire eigenschappen van een agent. Deze drie eigenschappen geven aanleiding tot een classificatie van vier mogelijke agenten, zoals ook te zien is in de figuur:

- *collaboratieve agenten*
- *collaboratieve, lerende agenten*
- *interface agenten*
- *slimme agenten*

Het onderscheid dat tussen de verschillende soorten agenten gemaakt wordt in deze topologie, moet niet letterlijk genomen worden. Het is niet zo dat bijvoorbeeld *collaboratieve, lerende agenten* geen autonoom gedrag vertonen of dat *interface agenten* nooit de eigenschap van *Samenwerking* bezitten. Dit onderscheid moet eerder op die manier gezien worden dat de eigenschappen binnen de intersectie het belangrijkste zijn.

Verder maakt deze topologie, zoals reeds vermeld, slechts een grove classificatie mogelijk. Van de andere eigenschappen en kenmerken uit de vorige sectie kan ook gebruik gemaakt worden om tot een meer specifieke classificatie te komen. Niet alle agenten kunnen echter altijd onder zulke classificatie ondergebracht worden of sommige agenten zijn te verdelen over meerdere klassen. Zulke agenten, die over meerdere klassen verdeeld kunnen worden, worden dan ook wel “Hybride” agenten genoemd.

Ten slotte kan men aan de hand van deze topologie ook bepaald worden of een stuk software voldoet aan de eisen van een agent. Simpelweg, als het stuk software niet binnen één van de intersecties valt, is dit geen agent.

4.5 Agent voor Digital Aura

In het interactiemodel “Digital Aura” is, zoals vermeld in sectie 4.1, nood aan een agent. De agent die dienst doet binnen het interactiemodel zou de gebruiker moeten ontlasten van de taak met betrekking tot het opvolgen van

de uitwisseling van informatie. Indien dit werk kan overgelaten worden aan de agent, kan de gebruiker zich concentreren op andere dingen, zoals zijn werk, of moet hij de applicatie die het interactiemodel toepast niet voortdurend in de gaten te houden. Een agent die het werk degelijk uitvoert kan bijgevolg zorgen voor een betere bruikbaarheid van het interactiemodel. Met andere woorden, het succes van het interactiemodel of de applicatie voor dit interactiemodel, hangt samen met een goed werkende agent.

De ontwikkeling van een goede agent bestaat erin te bepalen wat de precieze taken van deze agent zijn. Deze taken bepalen vervolgens over welke eigenschappen de agent moet beschikken. In de context van dit interactiemodel kunnen we de taken van de agent specificeren als:

- Informatie ontvangen/versturen van/naar andere agenten
- De ontvangen informatie bestuderen en bepalen of data interessant is voor de gebruiker op basis van zijn/haar voorkeuren en karakteristieken
- De gebruiker op de hoogte brengen indien er in de informatie interessante data aanwezig is

Aan de hand van deze taken kunnen de eigenschappen van de agent bestudeerd worden. Om te beginnen worden de drie elementaire eigenschappen van de topologie, uit de vorige sectie, beschouwd. De concrete invulling van deze eigenschappen en hun onderling belang geven een classificatie van de agent volgens de topologie. Vervolgens worden ook nog enkele andere mogelijke eigenschappen van de agent besproken.

4.5.1 Autonoom

De agent in dit interactiemodel moet zeker over de *autonoom* - eigenschap beschikken. Deze eigenschap neemt zelfs een belangrijke positie in. Het doel van de agent is namelijk de taak in naam van de gebruiker uit te voeren zonder dat de gebruiker hierin tussen komt. Hierbij heeft de agent de verantwoordelijkheid over het nemen van bepaalde beslissingen na het bestuderen van de ontvangen informatie. Deze beslissingen bestaan meestal uit het al dan niet op de hoogte brengen van de gebruiker indien interessante data ontvangen werd. Ook zal de agent prioriteiten moeten kunnen inschatten met

betrekking tot meerdere interessante data. Kortom, de agent is steeds waakzaam over de ontvangen informatie en zal een initiatief ondernemen indien data interessant is voor de gebruiker. Verder moet de agent, indien andere agenten dit verwachten, bepaalde informatie vrijgeven zonder dat de gebruiker hierin tussenkomt. Hij moet zelfstandig kunnen bepalen welk deel van deze informatie mag en welk deel niet mag uitgewisseld worden.

4.5.2 Samenwerken

In “Ubiquitous Computing” is er sprake van een actieve omgeving. In deze omgeving wordt informatie tussen verschillende computers uitgewisseld. De meeste computers in deze omgeving, waarin gebruik gemaakt wordt van dit interactiemodel, hebben een agent in uitvoering. Dit betekent dat informatie tussen de verschillende agenten uitgewisseld zal worden. De agenten moeten dus met elkaar samenwerken om te kunnen interageren met elkaar. In tegenstelling tot gedistribueerde systemen, werken deze agenten op zichzelf en vormen ze geen groter geheel.

De samenwerking tussen de agent en de gebruiker speelt hier ook een belangrijke rol. Enerzijds moet de gebruiker zijn interesses voor bepaalde data aan de agent kunnen duidelijk maken. Een agent kan slechts behoorlijk werken indien deze op de hoogte is van deze interesses. Anderzijds moet een goede agent de gebruiker degelijk op de hoogte brengen indien interessante informatie ter beschikking is gekomen door andere agenten in de omgeving.

4.5.3 Leren

Zoals vermeld in de vorige secties, moet een agent de interesses en voorkeuren van de gebruiker kennen opdat de agent goede beslissingen zou kunnen nemen. Doch is het “echt” leren van deze interesses in dit interactiemodel en bijhorend gebruikersmodel geen prioriteit. De agent kan deze interesses en voorkeuren voor geheimhouding opmaken uit het gebruikersmodel dat de gebruiker kan meegeven aan de agent. Men kan hier bijgevolg niet echt spreken over een agent die tijdens het verloop van uitvoering de interesses leert.

4.5.4 Andere eigenschappen

De agent voor dit interactiemodel beschikt verder nog over een aantal andere eigenschappen. Een eerste eigenschap, *veiligheid*, moet er voor zorgen dat de gebruiker vertrouwen kan hebben in de agent. Dit vertrouwen mag niet beschadigd worden door het vrijgeven van ongewenste informatie. Verder is dit soort van agenten niet *mobiel*. Agenten kunnen zich namelijk niet verder verspreiden in de actieve omgeving. Ze zijn verbonden aan het apparaat van de gebruiker. Ook *personaliseerbaarheid* kan een belangrijke eigenschap zijn voor een goede bruikbaarheid van de agent, aangezien de agent in rechtstreeks contact staat met de gebruiker.

4.5.5 Collaboratieve agent

Uit de studie en bespreking van de verschillende eigenschappen voor dit soort agenten kan de agent in de topologie van Nwana geplaatst worden in de intersectie tussen *Samenwerking* en *Autonoom*, met andere woorden het is een *Collaboratieve agent*. Deze twee eigenschappen van een *Collaboratieve agent* zijn duidelijk het belangrijkste indien men over een goede agent wilt beschikken voor dit interactiemodel. Maar zoals gezegd in sectie 4.4, is de topologie niet sluitend en moeten bijgevolg de andere eigenschappen van hierboven niet vergeten of verwaarloosd worden.

4.6 Conclusie

Agenten zijn nuttige hulpmiddelen voor een gebruiker om deze te ontlasten van bepaalde taken. Dit is ook het geval in het interactiemodel “Digital Aura”. In dit interactiemodel kunnen veel informatie-uitwisselingen plaatsvinden. Door de verwerking van deze informatie over te laten aan een agent, kan de gebruiker zich blijven concentreren op dagelijkse taken. Naast het overnemen van de taken van de gebruiker, speelt ook de samenwerking van deze agenten in het interactiemodel een belangrijke rol.

Beide kenmerken, *Samenwerking* en *Autonoom*, vormen de grootste onderdelen van agenten binnen dit interactiemodel. Hierdoor kunnen we dit soort van agenten classificeren als *Collaboratieve agenten*, volgens de topologie die ontwikkeld werd voor “persoonlijke agenten”. Deze twee kenmerken

zijn niet de enige waarover deze agent beschikt. De kenmerken *veiligheid* en *personaliseerbaarheid* behoren ook tot het karakter van de agent.

Een goede agent, ten slotte, is een agent die de kennis over de gebruiker goed hanteert in het maken van beslissingen in naam van de gebruiker. Bijgevolg is er nood aan een goede relatie tussen de gebruiker en zijn/haar agent. Deze relatie moet het mogelijk maken voor de gebruiker om zijn/haar kennis door te geven. De omgekeerde richting van deze relatie, namelijk van agent naar gebruiker, laat de agent toe om meldingen te maken van gebeurtenissen of beslissingen die genomen werden.

Hoofdstuk 5

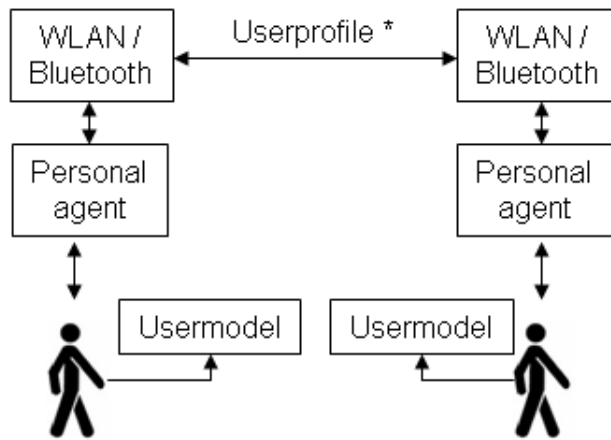
Implementatie

5.1 Inleiding

De theoretische achtergrond uit de vorige hoofdstukken zal in dit hoofdstuk gebruikt worden om tot een concrete implementatie te komen voor het interactiemodel “Digital Aura”. “Ubiquitous Computing”, waar dit interactiemodel in toegepast kan worden, vereist een omgeving waarin een aantal computers bevat liggen, en dus een “actieve” omgeving vormen. Om zulke “actieve” omgeving tot stand te brengen zal een omgeving gecreëerd worden door middel van mensen, die uitgerust worden met een “Personal Digital Assistant” (PDA).

De PDA’s van de mensen beschikken over een applicatie, die de implementatie van het interactiemodel bevat. Door gebruik te maken van deze applicatie kunnen de mensen impliciet met elkaar interageren, zoals de doelstelling van het interactiemodel aangeeft. De applicatie voor de PDA is geschreven in C-Sharp (C#) en vereist dat de PDA beschikt over Bluetooth en een Wireless LAN adapter (zie volgende sectie).

In figuur 5.1 wordt de werking van het interactiemodel “Digital Aura” schematisch weergegeven, waarbij de verschillende onderdelen duidelijk zichtbaar zijn. Een gebruiker maakt een gebruikersmodel aan, dat een gebruikersprofiel en een geheimhoudingsprofiel bevat. Wanneer een andere gebruiker zich in de buurt bevindt, ontstaat een spontane interactie tussen beiden. Tijdens deze interactie wordt informatie uitgewisseld tussen beide personen. Deze informatie bestaat uit een aangepast gebruikersprofiel waarbij mogelijk informatie



Figuur 5.1: Schematische voorstelling van de werking van Digital Aura.

werd weggelaten. Welke informatie hierbij weggelaten wordt, is afhankelijk van het geheimhoudingsprofiel van de gebruiker, waarin situaties en groepen van personen gedefinieerd zijn. Het aangepaste gebruikersprofiel komt vervolgens aan bij de agent, die de inhoud ervan controleert. Wanneer de agent vindt dat er interessante data aanwezig is, zal deze de gebruiker hiervan op de hoogte brengen door middel van een melding. De gebruiker kan ten slotte bepalen wat hij met deze informatie doet.

De implementatie van de verschillende onderdelen van het interactiemodel, zoals die in het theoretische gedeelte werden beschreven, worden in de komende secties besproken. Hierbij wordt duidelijk de link gelegd tussen de praktijk en de theorie. Het geheel van de verschillende onderdelen vormt uiteindelijk een volledige en bruikbare implementatie voor het interactiemodel. Tot slot worden nog enkele toepassingen op basis van dit interactiemodel besproken.

5.2 Communicatie

Opdat de verschillende applicaties met elkaar kunnen interageren moet er een communicatiemedium beschikbaar zijn. In dit interactiemodel kunnen we best gebruik maken van een draadloos medium zodat gebruikers vrij kunnen rondbewegen. Verder is dit interactiemodel gebaseerd op een metaforische

Medium	datasnelheid	Bereik	# connecties
infrarood	115.2 kbps tot 16 Mbps	2 m	1 - 2
Bluetooth	1 Mbps	10 m - 100 m	7
WLAN	11 Mbps of 54 Mbps	40 m - 100m	veel

Tabel 5.1: Overzicht draadloze communicatiemedia

aura. Zulke aura kan ontwikkeld worden door middel van communicatiezenders die een kort bereik hebben.

Op technologisch vlak bestaan een aantal mogelijkheden met betrekking tot draadloze media met een kort bereik, namelijk infrarood, Bluetooth en Wireless LAN. In tabel 5.1 worden een aantal kenmerken van de verschillende mogelijkheden vergeleken. Op het eerste zicht komen alle drie mogelijkheden in aanmerking voor het uitwisselen van informatie. Toch zijn ze niet allemaal even geschikt.

Om voor impliciete interactie te zorgen binnen het interactiemodel, is het nodig dat computers ontdekt kunnen worden binnen de “actieve” omgeving. Het impliciete aspect gaat immers verloren wanneer een gebruiker zelf een verbinding tot stand moet brengen met een andere computer. In principe kunnen alle mogelijkheden hiervoor in aanmerking komen. In het bijzonder beschikken infrarood en Bluetooth in hun software en/of hardware expliciet over de mogelijkheid om gelijkaardige apparaten te ontdekken. De andere eigenschap van het interactiemodel, namelijk de metaforische aura, legt beperkingen op aan het zendbereik van de zenders voor de media. Ook aan deze eigenschap voldoen infrarood en Bluetooth met hun bereik van respectievelijk 2 en 10 meter. De keuze van het medium is uiteindelijk gevallen op Bluetooth. De reden hiervoor is van praktische aard. Om met infrarood een verbinding tot stand te brengen, moeten zender en ontvanger in één lijn liggen en zichtbaar zijn voor elkaar. Ook is een bereik van twee meter redelijk kort, waardoor veel mogelijke interacties verloren zouden gaan. Bluetooth echter, legt geen restricties op aan de zichtbaarheid van apparaten. Mensen hoeven bijgevolg hun apparaat niet steeds zichtbaar te houden, en kunnen vrij bewegen binnen deze straal van tien meter. De straal van 10 meter is ten slotte een goed compromis omdat mensen niet dicht bij elkaar moeten zitten en de actieve omgeving geen al te grote ruimte beslaat.

Naast het ontdekken van apparaten, is het noodzakelijk dat informatie uitgewisseld kan worden tussen de verschillende apparaten opdat we interactie zouden kunnen ondersteunen. Een mogelijkheid om het uitwisselen mogelijk te maken, is ook hiervoor gebruik te maken van Bluetooth. Een belangrijke reden echter, om niet verder te gaan met Bluetooth, ligt in de tekortkomingen van de beschikbare software ter ondersteuning van Bluetooth (zie sectie 5.2.1). Infrarood kan evenwel niet gebruikt worden om uitwisseling van informatie te ondersteunen omwille van het feit dat, zoals reeds vermeld, infrarood slechts een beperkt bereik heeft en dat de apparaten voor elkaar zichtbaar moeten zijn. Wireless LAN ten slotte, biedt wel alle mogelijkheden om meerdere apparaten met elkaar te laten communiceren. Een Wireless LAN heeft ook een bijkomend praktisch voordeel in de situatie waarbij een apparaat slechts de rand van het bereik van Bluetooth passeert. Eenmaal de detectie van het andere apparaat gebeurd is, kunnen de apparaten over langere tijd met elkaar blijven communiceren. De duur van detectie is redelijk kort, maar de uitwisseling van informatie kan een lange periode in beslag nemen. Hierdoor ontstaat een grotere kans voor interacties, die tevens langer kunnen duren.

Het interactiemodel zal dus uiteindelijk gebruik maken van Bluetooth en Wireless LAN om de detectie en communicatie mogelijk te maken. In de volgende secties zullen de moeilijkheden en de details in verband met de implementatie van de communicatie op basis van beide media besproken worden.

5.2.1 Bluetooth

Een communicatieframework met behulp van Bluetooth maakt gebruik van een Bluetooth stack. Voor deze stack zijn er twee mogelijkheden, namelijk de stack van Windows of de Widcomm Bluetooth Stack. De pda's die ter beschikking waren, ondersteunen enkel de Bluetooth op basis van de Widcomm Bluetooth Stack [Bro]. De officiële Software Development Kit (SDK) voor deze stack is echter niet gratis beschikbaar. Daarom werd gewerkt met de alternatieve SDK van High Point [HPS]. Deze SDK voorziet alle benodigheden voor de instellingen van de Bluetooth zender alsook de mogelijkheden om andere apparaten te detecteren.

Na de detectie van andere apparaten moet met deze andere apparaten informatie uitgewisseld worden. Deze informatie is noodzakelijk om later een verbinding tussen de verschillende apparaten te kunnen maken via een Wireless LAN. De hoeveelheid informatie hierbij is beperkt waardoor kortstondige verbindingen via Bluetooth volstaan. De beschikbare SDK van High Point voorziet de mogelijkheid om zulke verbindingen tot stand te brengen via een seriële poort. De SDK voorziet echter alleen de initialisatie van deze verbindingen. De daadwerkelijke implementatie van seriële poorten en de hierbij horende mogelijkheid om data te verzenden, kan met behulp van deze SDK niet. Om dit probleem op te lossen kan gebruik gemaakt worden van een implementatie voor seriële poorten voor Pocket PC van OpenNETCF [Ope].

In een eerste poging om de detectie van apparaten, en het uitwisselen van de benodigde informatie voor communicatie via een Wireless LAN te verwezenlijken, bleek de methode om informatie uit te wisselen via een seriële verbinding niet naar behoren te functioneren. In het eenvoudige geval waarbij een apparaat een ander apparaat ontdekt en hiermee een verbinding tot stand brengt werkt deze methode goed. Een probleem ontstaat wanneer meerdere apparaten ontdekt worden. Het is in deze situatie niet mogelijk om meerdere verbindingen via seriële poorten tot stand brengen. Dit zou eventueel wel opgelost kunnen worden door opeenvolgende verbindingen naar de betreffende apparaten tot stand te brengen en vervolgens terug af te breken. Maar elk apparaat kan verder ook nog eens alle andere apparaten ontdekken, waarmee ook verbindingen opgezet moeten worden. In deze situatie moeten zowel ingaande als uitgaande verbindingen tot stand gebracht en beheerd worden. De SDK van High Point laat het hierbij afweten. Tenslotte laat de SDK van High Point ook niet toe dat er verder gedetecteerd wordt wanneer een verbinding tot stand gebracht werd, waardoor mogelijke interacties verloren kunnen gaan.

Om bovenstaand probleem, met name de communicatie tussen de verschillende apparaten met behulp van Bluetooth, op te lossen kan gebruik gemaakt worden van een minder conventionele methode. Door de mogelijkheden, die de SDK van High Point biedt, kan de naam van het apparaat veranderd worden. Deze naam wordt gebruikt bij de detectie van de apparaten om aan te geven welk apparaat ontdekt werd. Het detecterende apparaat kan deze naam makkelijk opvragen. Indien de naam van het apparaat wordt uitge-

breid met de informatie, die nodig is voor het tot stand brengen van een verbinding via Wireless LAN, beschikt dus het detecterende apparaat over deze informatie. Deze informatie voor een verbinding via een Wireless LAN omvat een IP-adres en een poortnummer. In deze implementatie heeft de aangepaste apparaatnaam volgende opmaak:

apparaatnaam + “|” + ip-adres + “|” + poortnummer

Een mogelijk voorbeeld volgens deze opmaak is “PocketPC|192.168.0.112|4321“. Met behulp van zulke apparaatnamen beschikken apparaten onmiddellijk bij de detectie over de nodige informatie. Dit levert een snelheidswinst op omdat er geen verbindingen meer tot stand gebracht moeten worden. Een bijkomend voordeel van deze methode is ook dat de detectie van andere apparaten niet onderbroken moet worden, zoals het geval was bij het opzetten van verbindingen. Hierdoor gaan, zoals reeds vermeld, geen mogelijke interacties verloren.

5.2.2 Wireless LAN

Een implementatie van een communicatieframework voor een Wireless LAN is op eenvoudige wijze realiseerbaar met behulp van sockets. Sockets laten verschillende typen van verbindingen toe, zoals TCP of UDP. Aangezien er over een draadloos netwerk informatie verzonden wordt, is het aangewezen gebruik te maken van een TCP verbinding. De kans op verloren pakketten is namelijk redelijk groot bij draadloze netwerken. Een TCP-verbinding voorziet een aantal mogelijkheden om zulk verlies van pakketten tegen te gaan, mits wat extra vertraging tijdens het verzenden. Het volledig aankomen van data bij de zender is in dit interactiemodel belangrijker dan tijdswinst.

In het communicatieframework wordt per PDA een kleine server voorzien. Op deze manier kunnen verschillende apparaten een verbinding tot stand brengen met één enkel apparaat. Deze verbindingen kunnen in stand gehouden worden totdat een apparaat het bereik van het Wireless LAN verlaat, of totdat er geen verder uitwisseling van informatie meer nodig is. Verder kan een PDA ook meerdere verbindingen maken met andere apparaten en deze onderhouden zolang dit mogelijk en nodig is.

In het interactiemodel “Digital Aura” bestaat de communicatie zelf, uit het uitwisselen van gebruikersprofielen, zoals weergegeven is in Figuur 5.1. Voor deze uitwisseling wordt gebruik gemaakt van “push”-systeem. In zulk systeem moet een gebruiker of computer, die informatie wenst, de informatie vragen aan de andere gebruiker of computer. Voor de realisatie van dit “push”-systeem wordt gebruik gemaakt van een tweetal berichten: het *request*- en het *reply*-bericht. Beide berichten bevatten een berichttype dat aangeeft welk soort bericht het is. Het *request*-bericht wordt aangevuld met de voor- en achternaam van de ondervrager. Deze data is nodig voor het bepalen van het deel van het gebruikersprofiel dat uitgezonden mag worden, zie Sectie 5.5. Het *request*-bericht tenslotte wordt aangevuld met het aangepaste gebruikersprofiel. Een schematisch overzicht van beide berichten is te zien in Figuur 5.2.

Berichttype	Informatie	
REQUEST	Voornaam	Achternaam
REPLY	Gebruikersprofiel*	

Figuur 5.2: Berichten voor communicatie via Wireless LAN

De verschillende onderdelen van beide berichten worden eenvoudig gecodeerd als een string waarbij elk onderdeel wordt gescheiden door middel van het “|”-teken. Na de detectie van een ander apparaat wordt een verbinding via een Wireless LAN opgezet. De applicatie die de verbinding opzet, stuurt vervolgens een *request*-bericht met daarin de voor- en achternaam van de gebruiker. Bij ontvangst van zo’n *request*-bericht moet de ontvanger een *reply*-bericht terugsturen. Indien geen geschikt gebruikersprofiel bestaat dat vrijgegeven mag worden, is de string van het gebruikersprofiel een lege string.

5.3 Gebruikersmodel

Het belangrijkste onderdeel van het interactiemodel is het gebruikersmodel, zoals aangegeven in sectie 2. Dit gebruikermodel bestaat uit twee delen, namelijk het gebruikersprofiel en het geheimhoudingsprofiel. Het gebruikersprofiel, dat de karakteristieken en kenmerken van de gebruiker bevat, is de

informatie die in het interactiemodel door de verschillende apparaten wordt uitgewisseld. Het geheimhoudingsprofiel, dat de voorkeuren van de gebruiker betreffende de geheimhouding bevat, bepaalt welke data uit het gebruikersprofiel uitgewisseld mag worden en welk niet uitgewisseld mag worden met bepaalde apparaten.

Het beoogde concept van een gebruikersmodel, zoals getoond wordt in figuur 3.2, toont een gebruikersmodel aan, waarbij het gebruikersprofiel en het geheimhoudingsprofiel sterk gekoppeld zijn. In de praktijk is het echter beter om beide profielen van elkaar los te koppelen. Een gebruiker kan op deze manier een gebruikersprofiel koppelen aan een ander geheimhoudingsprofiel, of omgekeerd. Deze manier van werken laat een hogere schaleerbaarheid toe.

Tussen beide delen van het gebruikermodel moet bijgevolg wel een soort relatie zijn. Zulke relatie moet duidelijk maken welke informatie in het gebruikersprofiel hoort bij een bepaalde voorkeur in het geheimhoudingsmodel. Hoe deze relatie uitgewerkt is en welke specifieke kenmerken beide delen bevatten wordt besproken in de volgende secties.

5.3.1 Relatie gebruikersprofiel - geheimhoudingsprofiel

De relatie tussen het gebruikersprofiel en het geheimhoudingsprofiel werd voorzien door middel van een attribuut, dat “Detail” genoemd werd in deze implementatie. Dit attribuut kan één van de waarden aannemen uit de verzameling {“High”, “Medium”, “Low”, “None”}. De benamingen van het attribuut en van de waarden, staan voor de mate van detail dat een stuk informatie bevat over de gebruiker. Dit “Detail”-attribuut kan in het gebruikersprofiel gekoppeld worden aan elk stukje informatie, zoals beschreven in de volgende sectie. In het geheimhoudingsprofiel wordt een gelijkaardige methode toegepast.

Een kort voorbeeld zal deze relatie duidelijk maken. Beschouw een stuk informatie in het gebruikersprofiel, bijvoorbeeld het adres van een gebruiker. Het “Detail”-attribuut heeft de waarde “High” gekregen omdat de gebruiker vindt dat deze informatie veel details over hem geeft. In het geheimhoudingsprofiel heeft de gebruiker een aantal groepen, zie sectie 5.3.3, opgesteld die

informatie mogen ontvangen van hem of haar. Een eerste groep, “Collega’s”, krijgt als waarde voor het “Detail”-attribuut “Medium”, terwijl de tweede groep, “Vrienden”, als waarde “High” krijgt. Indien nu een persoon uit de groep “Collega’s” informatie opvraagt van de gebruiker, krijgt deze persoon geen informatie over het adres van de gebruiker omdat de persoon enkel informatie mag ontvangen die uit ten hoogste middelmatige details mag bestaan. Een persoon uit de “Vrienden”-groep krijgt het adres van de gebruiker wel tot zijn beschikking, evenals alle andere informatie waarvan het detail lager of gelijk aan “High” is.

5.3.2 Gebruikersprofiel

```

<?xml version="1.0" encoding="utf-16" ?>
- <userprofile>
- <identity>
- <situationalstatement>
  <auxiliary>hasProperty.600100</auxiliary>
  <predicate>givenname.800410</predicate>
  <range>Text.640002</range>
  <value>jan</value>
  <start>4/05/2006 19:03:49</start>
  <duration>Inherent.520002</duration>
  <detail>medium</detail>
</situationalstatement>
</identity>
- <work>
- <situationalstatement>
  <auxiliary>hasKnowledge.600120</auxiliary>
  <predicate>UbiquitousComputing.120586</predicate>
  <range>TerribleExcellent.640230</range>
  <value>Excellent</value>
  <start>4/05/2006 19:04:29</start>
  <duration>Year.520020</duration>
  <detail>low</detail>
</situationalstatement>
</work>
<love />
<freetime />
</userprofile>

```

Figuur 5.3: XML fragment van een gebruikersprofiel

De implementatie van het gebruikersprofiel werd gedaan met behulp van XML, gelijkaardig aan UserML. In figuur 5.3, dat een XML fragment uit een gebruikersprofiel toont, blijkt dat de theoretische structuur voor het gebruikersprofiel werd behouden, waardoor het gebruikersprofiel over een goede toegankelijkheid beschikt. Deze structuur bevat vier onderdelen, namelijk *Identity*, *Work*, *Love* en *Free-time*. Elke groep beschikt over zijn eigen lijst van *Situational Statements*. Om de toegankelijkheid van het gebruikersprofiel verder te verhogen werd een editor voorzien waarin de gebruiker makkelijk nieuwe *Situational Statements* aan een onderdeel in het gebruikersmodel kan toevoegen, of bestaande *Situational Statements* kan veranderen (zie Figuur 5.4).



Figuur 5.4: Screenshot van editor voor gebruikersprofiel

De *Situational Statements* bestaan op hun beurt uit een aantal datavelden. De mogelijke waarden die toegekend kunnen worden aan de datavelden *Auxiliary*, *Predicate*, *Range* en *Duration*, zijn afkomstig van GUMO. Het overige dataveld *Start* bevat een eenvoudige tijdstempel die aanduidt wanneer het *Situational Statement* voor het laatst werd aangepast.

Voor het bekomen van de waarden van GUMO, werd gebruik gemaakt van een RDF document dat de ontologie beschrijft. Een probleem echter bij het gebruik van C# voor Pocket PC is het feit dat de ondersteunende XML-parser geen elementen zoals *DOCTYPE* ondersteunt, welke wel gebruikt worden binnen RDF. Om deze tekortkoming op te lossen werd gebruik gemaakt van een eigen conversieprogramma. Dit conversieprogramma zet een RDF document om in een zelf gespecificeerd XML formaat, dat ook ondersteund wordt door het programma in C#. De graafstructuur van het RDF document wordt hierbij omgezet in een aangepaste boomstructuur.

Ten slotte werd een *Situational Statement* uitgebreid met het attribuut "Detail". Dit attribuut moet zorgen voor de relatie tussen het gebruikersprofiel en het geheimhoudingsprofiel. Deze relatie werd reeds besproken in de vorige sectie.

5.3.3 Geheimhoudingsprofiel

Net als voor het gebruikersprofiel, wordt voor het geheimhoudingsprofiel gebruik gemaakt van XML, zie figuur 5.5. Hierbij wordt dezelfde structuur aangehouden als de vier onderdelen. Maar in plaats van *Situational Statements*, bevatten de groepen in het geheimhoudingsprofiel enerzijds een lijst van groepen, en anderzijds een lijst van *Context statements*. Ook voor dit profiel werd een editor voorzien voor het aanpassen van de informatie in het geheimhoudingsprofiel, waardoor de toegankelijkheid tot het geheimhoudingsprofiel vergroot (zie Figuur 5.6).

De groepen zijn, zoals vermeld in hoofdstuk 3, een mogelijkheid voor de gebruiker om te bepalen wie welke informatie mag ontvangen. In het geheimhoudingsprofiel worden enkel de namen van de groepen gebruikt. Hierbij worden de namen van de groepen gekoppeld aan het "Detail"-attribuut. Door deze scheiding kan een gebruiker eenmaal hij/zij de groepen heeft aangemaakt, deze herbruiken in andere geheimhoudingsprofielen. De beschrijving van de groepen, met name de lijst van personen die tot een bepaalde groep behoren, worden opgeslagen in een ander bestand. In het begin van het geheimhoudingsprofiel wordt naar dit bestand van de groepen verwezen.

```

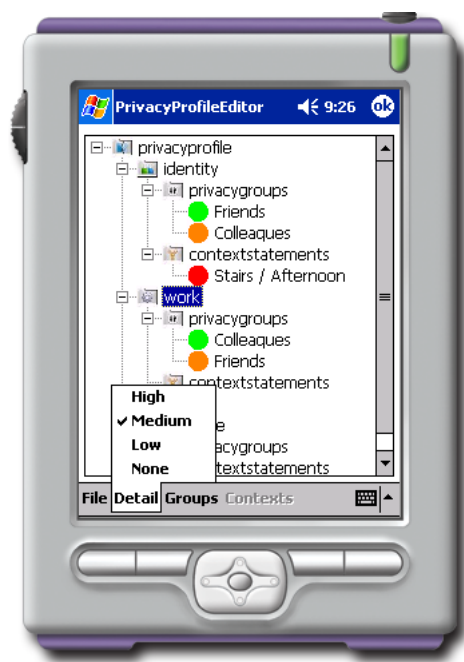
<?xml version="1.0" encoding="utf-16" ?>
- <privacyprofile>
  <groupsresource>\My Documents\PROFILES\Groups.xml</groupsresource>
- <identity defaultdetail="high">
  - <privacygroups>
    <group detail="high">friends</group>
    <group detail="medium">colleagues</group>
  </privacygroups>
  - <contextstatements>
    - <contextstatement>
      <temporal>always.510024</temporal>
      <spatial>Paris.400024</spatial>
      <detail>low</detail>
    </contextstatement>
  </contextstatements>
</identity>
- <work defaultdetail="high">
  <privacygroups />
  <contextstatements />
</work>
- <love defaultdetail="high">
  <privacygroups />
  <contextstatements />
</love>
- <freetime defaultdetail="high">
  <privacygroups />
  <contextstatements />
</freetime>
</privacyprofile>

```

Figuur 5.5: XML fragment van een geheimhoudingsprofiel

De *Context statements* bestaan elk uit een *Temporal*- en een *spatial*-element. Voor de waarden, die aan deze twee elementen kunnen worden toegekend, wordt zoals bij de *Situational Statements* gebruik gemaakt van GUMO. Deze mogelijke waarden worden samen met die voor de *Situational Statements* naar een specifiek XML formaat geconverteerd. Het “Detail”-attribuut maakt deze *Context statements* compleet.

Ten slotte wordt voor elk onderdeel van het geheimhoudingsprofiel een standaard “Detail”-attribuut voorzien. Dit extra attribuut moet het geheimhoudingsprofiel compleet maken. Indien voor een bepaald onderdeel niet voor iedere persoon of situatie geheimhouding gespecificeerd wordt, wordt de waar-



Figuur 5.6: Screenshot van editor voor geheimhoudingsprofiel

de van dit extra attribuut gebruikt bij het bepalen van de delen informatie die mogen uitgezonden worden. Het is bijna onmogelijk om elke persoon of situatie te voorzien, waardoor dit attribuut belangrijk is om voor een goede geheimhouding te zorgen.

5.3.4 Valkuilen

In sectie 3.4 werden twee valkuilen besproken die invloed hebben op het gebruikersmodel. Een eerste valkuil, *obscuring potential information flow*, werd vermeden door een goede structuur te gebruiken. Een mogelijke, bijkomende eigenschap van het gebruikersmodel om deze eigenschap te voorkomen, ligt in het gebruik van het “Detail”-attribuut. Gebruikers kunnen zelf bepalen welke informatie zij belangrijk vinden. Deze gedachten over de belangrijkheid van informatie zal een gebruiker redelijk lang bijblijven.

De tweede valkuil met betrekking tot het gebruikersmodel, *emphasizing configuration over action*, wordt om te beginnen gedeeltelijk vermeden door de goede structuur en het gebruik van groepen. Naast de goede structuur,

kan een gebruiker ook gebruik maken van een editor, die het aanpassen van het geheimhoudingsprofiel vereenvoudigt en bijgevolg ook minder configuratietijd vereist. Ten slotte kan, door aanwezigheid van het standaard “Detail”-attribuut, de gebruiker zelf bepalen hoe ver hij/zij wil gaan in het specificeren van de geheimhouding. Een gebruiker kan simpelweg gebruik maken van de standaard “Detail”-attributen waarbij weinig configuratie vereist wordt. Anderzijds kan hij/zij ook elke situatie en elke persoon definiëren waarbij veel configuratie vereist is.

5.4 Geheimhouding

In de vorige sectie werd reeds de implementatie van geheimhouding met betrekking tot het gebruikersmodel besproken. Naast de geheimhouding in het gebruikersmodel, zoals vermeld in hoofdstuk 3, is ook nood aan geheimhouding binnen de applicatie zelf. Na het uitvoeren van het risicomodel, blijken de kosten van de implementatie van deze geheimhouding niet op te wegen tegen de schade die de gebruiker kan oplopen wanneer ongewenste informatie wordt vrijgegeven. In deze sectie wordt besproken hoe deze geheimhouding tot stand gebracht werd in de implementatie aan de hand van de drie valkuilen *obscuring actual information flow*, *lacking coarse-grained control* en *inhibiting established practice*, zoals aangegeven in sectie 3.4.

Voor het vermijden van de eerste valkuil, *obscuring actual information flow*, moet de gebruiker via de applicatie in staat zijn om te weten te komen welke geheimhouding er geldt tijdens de informatie-uitwisseling. Om deze mogelijkheid voor de gebruiker te voorzien werd de applicatie uitgerust met een notificatiesysteem dat de gebruiker op de hoogte brengt wanneer er informatie wordt uitgewisseld en welke geheimhouding er op het moment van de uitwisseling van kracht was. Deze manier van werken, waarbij de gebruiker steeds op de hoogte gebracht wordt, kan leiden tot overdadige berichten indien er veel uitwisselingen plaatsvinden. Daarom kan de gebruiker kiezen om niet meer op de hoogte gebracht te worden. Om ervoor te zorgen dat de valkuil toch vermeden wordt in deze situatie, kan de gebruiker nog steeds een geschiedenis raadplegen van alle gebeurde uitwisselingen. Op deze manier kan de gebruiker eventuele fouten later herstellen.

Met behulp van het concept van de metaforische aura, kunnen we een top-level mechanisme voorzien binnen de applicatie voor het specificeren van een bepaalde geheimhouding. De metaforische aura bestaat uit een onderdeel voor de uitgaande en de binnenkomende informatie. Voor de uitgaande informatie heeft de gebruiker de mogelijkheid om deze in of uit te schakelen of om zijn geheimhoudingsprofiel te overschrijven, door in alle gevallen een bepaalde waarde van het “Detail”-attribuut te kiezen waaraan alle uitgaande informatie moet voldoen. Dit in of uitschakelen, of het kiezen van een bepaalde waarde, kan eenvoudig gebeuren met een “combobox”. Voor de binnenkomende informatie heeft de gebruiker enkel de mogelijkheid om deze in of uit te schakelen via een druk op een knop. Een bepaalde “Detail” waarde kiezen heeft in dit geval weinig zin, omdat een gebruiker liefst alle mogelijke informatie wenst te ontvangen. Door het gebruik van beide methoden wordt de tweede valkuil *lacking coarse-grained control* bijgevolg vermeden.

De laatste valkuil betreffende de applicatie, *inhibiting established practice*, wordt vermeden door het dagelijkse, sociale gebruik van geheimhouding in de dagelijkse samenleving. Net als bij gsm’s wordt in deze applicatie voor het interactiemodel “Digital Aura” geen reden gegeven indien een gebruiker geen informatie wenst uit te wisselen. Degene die wel informatie wenst uit te wisselen wordt hierbij in het ongewisse gelaten wanneer er geen interactie met andere personen plaatsvindt. Mogelijke redenen voor het niet interageren kunnen zijn dat de andere gebruiker niet over deze applicatie beschikt, dat hij/zij geen informatie uitzendt, ...

Samen met de andere valkuilen van het gebruikersprofiel, vermijdt deze implementatie voor het interactiemodel “Digital Aura” bijgevolg alle vijf valkuilen. Dit betekent echter niet dat de geheimhouding sluitend is; er kunnen namelijk altijd menselijke fouten optreden. Maar door het vermijden van de vijf valkuilen worden een groot aantal risico’s met betrekking tot geheimhouding uitgesloten. Indien niet voldaan werd aan de vijf valkuilen, zou het voor de gebruiker moeilijk of zo niet onmogelijk zijn om aan een goede geheimhouding te doen.

5.5 Persoonlijke agent

Een persoonlijke agent voor het interactiemodel Digital Aura moet, zoals vermeld in Hoofdstuk 4, een collaboratieve agent zijn. De belangrijkste eigenschappen van dit soort agenten zijn *samenwerken* en *autonoom*. Gezien de belangrijkheid van beide eigenschappen, is het noodzakelijk dat bij een persoonlijke agent voor het interactiemodel Digital Aura deze eigenschappen goed geïmplementeerd zijn. Voorts moet een agent voor dit interactiemodel een aantal taken kunnen uitvoeren: het uitwisselen van gebruikersprofielen met de andere agenten, het bestuderen van de ontvangen data en ten slotte het op de hoogte brengen van de gebruiker indien er interessante data aanwezig is in het gebruikersprofiel. Hoe deze eigenschappen verwezenlijkt worden in de implementatie, en hoe de taken precies uitgevoerd worden, staat beschreven in de volgende secties.

5.5.1 Eigenschappen

Autonoom

De taak betreffende het uitwisselen van gebruikersprofielen, wordt door de agent zelf uitgevoerd. Wanneer een ander apparaat gedetecteerd wordt, wordt de agent hiervan op de hoogte gebracht. Samen met deze melding, wordt ook de informatie horende bij de detectie overgemaakt aan de agent. De agent beschikt hierdoor over alle nodige informatie om een verbinding tot stand te kunnen brengen met de andere agent via een Wireless LAN. Hierbij hoeft de gebruiker dus niet tussen te komen.

Voor het bestuderen van het ontvangen gebruikersprofiel en het op de hoogte brengen van de gebruiker indien er interessante data aanwezig is, wordt de agent voorzien van het gebruikersprofiel van de gebruiker. Uit dit gebruikersprofiel kan de agent alle nodige data halen die nodig is om een degelijk onderzoek uit te voeren. Wanneer dit gebruikersprofiel aangepast wordt door de gebruiker worden deze veranderingen ook kenbaar gemaakt aan de agent.

Samenwerken

De samenwerking tussen de agenten gebeurt op basis van een “pull”-systeem met uitwisseling van berichten, zoals vermeld in Sectie 5.2.2. Anderzijds betekent deze eigenschap ook dat er samenwerking tussen de gebruiker en de agent nodig is.

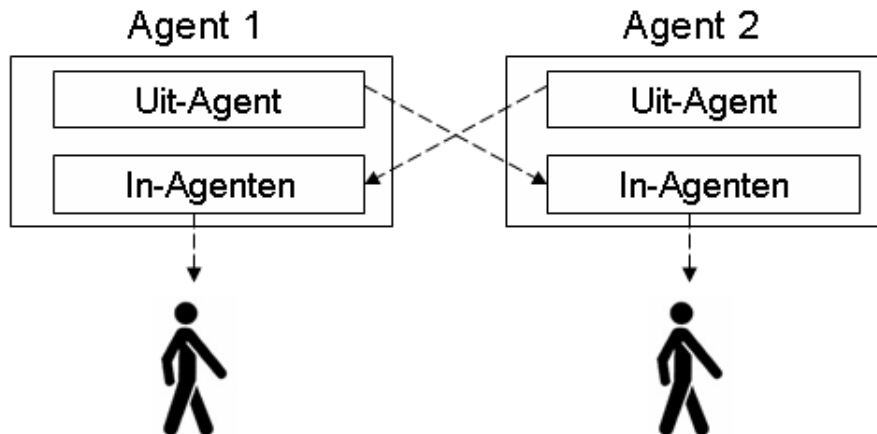
Opdat een agent in naam van de gebruiker goede beslissingen zou kunnen maken bij het bestuderen van de ontvangen gebruikersprofielen, moet de agent beschikken over de interesses en karakteristieken van de gebruiker. Deze informatie over de gebruiker wordt door middel van het gebruikersprofiel van de gebruiker bekend gemaakt aan de agent. Ook moet de agent over bepaalde informatie beschikken om taken uit te voeren, zoals het bepalen van welke informatie vrijgegeven mag worden en welke ontvangen mag worden. Deze informatie wordt door de gebruiker bekend gemaakt via het hoofdscherm van de applicatie.

Anderzijds moet de agent de gebruiker op de hoogte brengen indien interessante informatie ontdekt werd in het ontvangen gebruikersprofiel. Dit kan de agent doen door het afspelen van een kort geluidje en het tonen van een melding. In deze melding wordt de interessante data, die ontdekt werd, getoond.

5.5.2 Taken

De twee belangrijkste taken die een agent moet kunnen uitvoeren is het verzenden van aangepaste gebruikersprofielen en het bestuderen van ontvangen gebruikersprofielen. Voor het uitvoeren van deze taken maakt de agent gebruik van twee kleinere agenten, namelijk een *Uit-Agent* en een *In-Agent*. De *Uit-Agent* moet bepalen welke informatie van het gebruikersprofiel vrijgegeven mag worden op basis van het geheimhoudingsprofiel. De *In-Agent* heeft als taak het binnenkomende gebruikersprofiel te bestuderen en mogelijk de interessante data eruit te halen. Een agent moet over één *Uit-Agent* en over minstens één *In-Agent* beschikken om te kunnen functioneren. De reden hiervoor is dat er slechts één uitgaand gebruikersprofiel mag zijn, maar meerdere agenten kunnen een ontvangen gebruikersprofiel onderzoeken waarbij elke subagent een specifiek onderzoek uitvoert. Voor beide agenten

werd ten slotte een interface voorzien, waardoor gemakkelijk nieuwe subagenten geschreven en/of toegevoegd kunnen worden aan de agent. De werking van de agenten en de subagenten wordt ook weergegeven in Figuur 5.7. De geïmplementeerde agenten worden beschreven in de volgende secties.



Figuur 5.7: Overzicht werking persoonlijke agent

Een bijkomende taak van de agent is het bijhouden van een geschiedenis van de taken die de agent uitgevoerd heeft en de bijhorende resultaten. Deze geschiedenis maakt deel uit van een goede geheimhouding, zie Hoofdstuk 3. De subagenten die deel uitmaken van de agent moeten bijgevolg bij elke actie die ze ondernemen een melding hiervan maken bij de agent. Verder moet de agent, indien de gebruiker dit wenst, geen vrijgave van gebruikersprofielen naleven. Dit kan de agent doen door zich via Bluetooth niet detecteerbaar te maken.

5.5.3 Uit-Agent

Zoals reeds vermeld, staat deze subagent in voor het bepalen van de informatie van het gebruikersprofiel die al dan niet vrijgegeven mag worden op basis van het geheimhoudingsprofiel. De implementatie van deze subagent is afgeleid van de interface voor een *Uit-Agent*. Hierbij moeten twee welbepaalde methoden voorzien worden, die door de agent aangeroepen worden. Het resultaat van één van deze methoden moet het aangepaste gebruikers-

profiel zijn, dat de informatie bevat die vrijgegeven mag worden. De andere methode moet een log aanmaken voor elke vrijgave.

Voor de implementatie van het interactiemodel van “Digital Aura” is een volledig werkende *Uit-Agent* voorzien. Tijdens het berekenen van het aangepaste gebruikersprofiel maakt deze subagent gebruik van de voor- en achternaam van de ondervrager, of van de huidige context van de gebruiker. De voor- en achternaam zijn informatiegegevens die meegeleverd worden tijdens de uitwisseling van informatie tussen de agenten. De informatie over de context anderzijds, wordt door de gebruiker zelf gespecificeerd. De gebruiker bepaalt uiteindelijk zelf aan welke restrictie de voorkeur wordt gegeven, namelijk een aangepast gebruikersprofiel op basis van de naam van de ondervrager of op basis van de context van de gebruiker.

Het bepalen van de informatie die uitgezonden mag worden in het gebruikersprofiel, gebeurt aan de hand van het geheimhoudingsprofiel van de gebruiker. Dit geheimhoudingsprofiel bevat, zoals vermeld in sectie 5.3, een lijst van groepen met personen en een lijst van contexten. Aan elke groep en aan elke context werd met behulp van het “Detail”-attribuut een restrictie toegekend. Deze bepaalt welke informatie mag vrijgegeven worden aan die groep van personen of in de betreffende context.

Wanneer een gebruikersprofiel moet berekend worden op basis van de naam van de ondervrager, zal deze persoon gezocht worden in het geheimhoudingsprofiel. De waarde voor het “Detail”-attribuut van de groep waar deze persoon aan toe behoort, zal dan gebruikt worden bij de berekening. Indien geen enkele persoon met de bepaalde naam gevonden wordt, zal de waarde van standaard “Detail”-attribuut gebruikt worden. Dit standaard “Detail”-attribuut is voorzien voor elk onderdeel van het gebruikersmodel (“Identity”, “Work”, enz.).

Indien de voorkeur gegeven wordt aan de context van de gebruiker, zal gezocht worden naar een voorgedefinieerde context in het geheimhoudingsprofiel. Deze context is afhankelijk van de plaats waar de gebruiker zich bevindt tijdens de ondervraging en het tijdstip hiervan. Als er geen voorgedefinieerde context bestaat waaraan de context tijdens de ondervraging

voldoet, zal ook hier gebruik gemaakt worden van het standaard “Detail”-attribuut voor elke onderdeel van het gebruikersmodel.

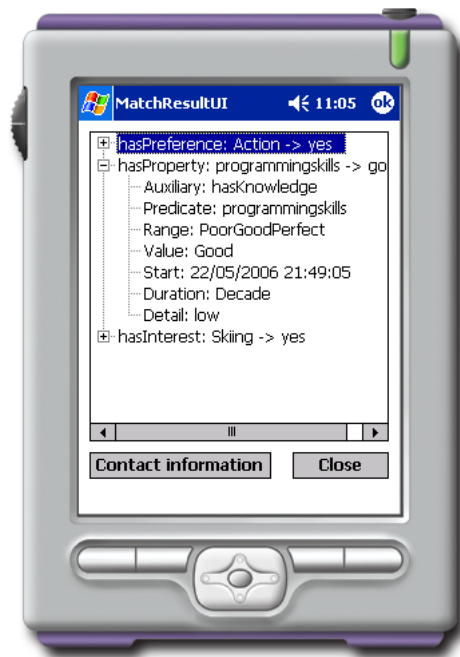
5.5.4 In-Agent

De tweede subagent, *In-Agent*, wordt door de agent uitgevoerd indien een gebruikersprofiel ontvangen wordt. Het doel van deze subagent is het onderzoeken van, voor de gebruiker, interessante informatie in het ontvangen gebruikersprofiel. Net als bij de *Uit-Agent* moet een implementatie voor deze agent afgeleid worden van de overeenkomende interface. De twee verplichte methoden voor deze subagent moeten, enerzijds vergelijkingen uitvoeren van het eigen en het ontvangen gebruikersprofiel, en anderzijds een log voorzien voor elke vergelijking die plaats vindt.

In een eerste fase werd een basis *In-Agent* ontwikkeld, die voorzien werd van een functionaliteit om mogelijke overeenkomsten te bepalen. Het berekenen van deze mogelijke overeenkomsten gebeurt aan de hand van de *predicates* die aanwezig zijn in beide gebruikersprofielen. Door het gebruik van de uniforme notatie, met behulp van GUMO, kunnen overeenkomsten gemakkelijk gevonden worden door een simpele vergelijking van de labels of id's horende bij deze *predicates*, zie sectie 2.4. Deze lijst van mogelijke overeenkomsten tussen beide gebruikersprofielen biedt een goede uitgangspositie voor het bepalen van specifieke overeenkomsten, waarbij rekening gehouden kan worden met de *auxiliaries* in het , de geldigheid van het *situational statement*, enz. Naar welke specifieke overeenkomsten gezocht moet worden is afhankelijk van de toepassing die men wenst. Een tweetal geïmplementeerde toepassingen worden beschreven in 5.6, waarbij ook de details van het zoeken naar de specifieke overeenkomsten vermeld worden.

Indien de vergelijking van beide gebruikersprofielen interessante data oplevert, moet de methode voor de vergelijking, deze interessante data bekend maken aan de gebruiker. Het formaat waarin de resultaten beschreven worden, mag zelf gekozen worden. Het op de hoogte brengen van een gebruiker kan door een melding te genereren, bijvoorbeeld door een dialoogvenster of een meer uitgebreid scherm. Een voorbeeld van zulke meldingen wordt weergegeven in Figuur 5.8. In dit voorbeeld wordt de interessante informatie uit het ontvangen gebruikersprofiel getoond. In deze implementatie kan ook

informatie over de identiteit van de persoon waarvan het gebruikersprofiel ontvangen werd, opgevraagd worden om deze persoon eventueel te kunnen contacteren.



Figuur 5.8: Screenshot van melding voor gebruiker bij interessante data

5.6 Toepassingen

Aan de hand van het ontwikkelde interactiemodel werden een tweetal concrete toepassingen gerealiseerd, gebaseerd op de toepassingen die beschreven werden in het werk van Ferscha [FHM⁺04] en in sectie 1.2.2. In de volgende secties worden de scenario's van beide toepassingen besproken, alsook de details betreffende de implementatie. Hoewel deze toepassingen eenvoudig zijn, geven ze toch een goed beeld van de mogelijkheden van het ontwikkelde interactiemodel.

5.6.1 Hulp nodig

Indien een persoon hulp nodig heeft met een bepaald onderwerp, zoals bijvoorbeeld algoritmen voor patroonherkenning, kan deze persoon dit bekend

maken aan de omgeving met behulp van het interactiemodel en kan hij/zij verder gaan met andere taken. Wanneer, na verloop van tijd, een andere persoon, bijvoorbeeld een docent in de informatica, deze vraag tot hulp ontvangt kan hij/zij de hulpzoekende persoon contacteren en helpen.

Een eerste stap in deze toepassing is het bekend maken van de nood aan hulp aan de omgeving. Om dit te realiseren kan gebruik gemaakt worden van het gebruikersprofiel. De hulpzoekende persoon kan hiervoor tijdelijk zijn/haar gebruikersprofiel aanpassen door deze vraag naar hulp toe te voegen. De hulpzoekende persoon stelt de mate van geheimhouding voor dit stuk informatie best niet hoog in, opdat alle mogelijke hulpverlenende personen deze informatie kunnen ontvangen. Eenmaal deze aanpassing gemaakt werd, wordt het nieuwe gebruikersprofiel vrijgegeven wanneer interactie tot stand gebracht wordt door middel van het interactiemodel.

Bij ontvangst van een gebruikersprofiel, waarin een verzoek tot hulp beschreven staat, wordt het ontvangen gebruikersprofiel vergeleken met het gebruikersprofiel van de ontvanger. In deze toepassing moet de *In-Agent* een specifieke overeenkomst zoeken tussen het domein waarvoor hulp gezocht werd en de domeinen waarover de ontvanger kennis heeft. Een *In-Agent* voor deze toepassing is gebaseerd op de basis *In-Agent*, zie sectie 5.5.4. Uit de beschikbare lijst van mogelijke overeenkomsten kan met behulp van de *auxiliaries* in het gebruikersprofiel bepaald worden of er kennis is van het domein of als er hulp gevraagd wordt over het domein. Ook de geldigheid (*start* en *Duration*) van de vraag naar hulp kan hierbij in rekening gebracht worden om verlopen vragen naar hulp te negeren. Indien de ontvanger hulp kan verlenen, wordt de vraag naar hulp bekend gemaakt en kan hij/zij de nodige stappen ondernemen om hulp te bieden.

5.6.2 Poster

Een poster van een muziekconcert of een tentoonstelling wordt niet altijd opgemerkt door voorbijgangers. Door gebruik te maken van een kleine computer en dit interactiemodel kan dit probleem met posters opgelost worden. Voorbijgangers worden, zonder dat ze op de posters moeten letten, verwittigd wanneer het concert of de tentoonstelling voor hun interessant is. Posters hoeven hierbij ook geen onnodige ruimte meer te beslaan.

Indien we de poster beschouwen als een persoon die de informatie bekend maakt, kan een gebruikersprofiel opgemaakt worden voor hetgeen waar de poster reclame voor maakt. In dit gebruikersmodel kunnen de datum, plaats, enz. beschreven worden. Door het interactiemodel wordt deze informatie vrijgegeven aan personen die passeren. Net als bij de vraag naar hulp in de vorige toepassing, wordt ook hier best de mate van geheimhouding laag gespecificeerd zodat alle mensen deze informatie kunnen ontvangen.

Een *In-Agent* voor deze toepassing moet overeenkomsten zoeken in de voorkeuren van personen betreffende muziek, film, museum, enz. en de karakteristieken van het concert of de tentoonstelling. Deze *In-Agent* is ook hier gebaseerd op de basis *In-Agent* uit sectie 5.5.4. Specifieke overeenkomsten kunnen bijgevolg gevonden worden door de lijst van mogelijke overeenkomsten te onderzoeken en rekening te houden met de *auxiliaries* en de geldigheid (*start* en *Duration*) van de *situational statements* in het gebruikersprofiel. Na een succesvolle vergelijking wordt de nodige informatie over het concert of de tentoonstelling bekend gemaakt aan de voorbijganger.

5.7 Conclusie

Dit hoofdstuk bespreekt een implementatie van de verschillende onderdelen van het interactiemodel “Digital Aura”. De theoretische bevindingen van de verschillende onderdelen, die beschreven zijn in de vorige hoofdstukken, werden hierbij zorgvuldig in acht genomen. Deze verschillende onderdelen werden bijeengebracht tot een werkend interactiemodel. Aan de hand van dit interactiemodel werden vervolgens een tweetal toepassingen ontwikkeld.

Voor het communicatieonderdeel van het interactiemodel werd beroep gedaan op de mogelijkheden van Bluetooth en Wireless LAN. Met behulp van Bluetooth kan de metaforische aura, die vereist is voor dit interactiemodel gemakkelijk ontwikkeld worden. Communicatie door middel van een Wireless LAN werd aangewend om de tekortkomingen van Bluetooth op te lossen, namelijk het correct uitwisselen van berichten tussen meerdere apparaten en het korte bereik.

De implementatie van het gebruikersmodel is een logisch vervolg op de theoretische bevindingen. Om de relatie tussen het gebruikersprofiel en het

geheimhoudingsprofiel te voorzien, werd een attribuut, “Detail” genaamd, gebruikt dat in beide profielen ondergebracht werd.

Geheimhouding in het gebruikersprofiel en de applicatie bestaat voornamelijk uit het vermijden van de vijf valkuilen. Deze valkuilen werden verder vermeden in het gebruikersprofiel door middel van editors en de structuur van het gebruikersmodel. De valkuilen betreffende applicatie werden vermeden door meldingen, logs, knoppen, enz.

De persoonlijke agent, die taken uitvoert in naam van de gebruiker, werd opgebouwd vanuit subagenten. Hierbij neemt de subagent *Uit-Agent* de berekening van informatie die vrijgegeven mag worden voor zijn rekening. De andere subagent, *In-Agent*, zoekt interessante data in het ontvangen gebruikersprofiel en meldt dit aan de gebruiker.

Hoofdstuk 6

Conclusie

“Ubiquitous Computing” is een domein dat steeds meer aan belang toeneemt, gezien de evolutie van computers en hun integratie binnen de dagelijkse samenleving. Binnen dit domein is de klassieke manier van interactie niet meer voldoende en is er nood aan nieuwe technieken. Een voorstel voor een nieuwe interactietechniek is het interactiemodel van “Digital Aura”. Dit interactiemodel maakt gebruik van metaforische aura’s om impliciete interactie tussen personen en/of computers tot stand te brengen.

In deze thesis is het interactiemodel “Digital Aura” ontwikkeld. Dit interactiemodel kan opgedeeld worden in verschillende onderdelen: een gebruikersmodel, geheimhouding en een persoonlijke agent. Elk onderdeel heeft een eigen welbepaalde functie in het interactiemodel. De bevindingen van elk onderdeel, die tijdens deze thesis werden ontdekt, worden hieronder vermeld.

6.1 Bevindingen

6.1.1 Gebruikersmodel

Een goed gebruikersmodel beschrijft niet enkel de karakteristieken van een gebruiker, maar ook zijn/haar voorkeuren met betrekking tot geheimhouding. Beide aspecten kunnen teruggevonden worden in het gebruikersprofiel en het geheimhoudingsprofiel. Het gebruik van GUMO en UserML hierbij, biedt een antwoord op de nood aan een uniforme notatie.

6.1.2 Geheimhouding

Geheimhouding is een belangrijk goed en is noodzakelijk in dit interactiemodel volgens een risicomodel. Een goede geheimhouding kan tot stand komen door het vermijden van een vijftal valkuilen met betrekking tot het gebruikersmodel en de applicatie. Het vermijden van deze valkuilen biedt echter geen garantie op een sluitende geheimhouding.

6.1.3 Persoonlijke agent

Persoonlijke agenten voor dit interactiemodel vallen onder de categorie “Collaboratieve agenten” volgens de topologie op basis van de eigenschappen van persoonlijke agenten. De voornaamste kenmerken van zulke agenten zijn *samenwerken* en *autonoom*. Deze eigenschappen beschrijven het zelfstandig handelen van de agent en de goede samenwerking tussen de agent en de gebruiker.

6.1.4 Implementatie

Het samenvoegen van de implementatie van de verschillende onderdelen leidt tot een werkend interactiemodel, op basis waarvan een aantal toepassingen gerealiseerd kunnen worden. In deze implementatie wordt gecommuniceerd via Bluetooth en een Wireless LAN. De subagenten, horende bij de persoonlijke agent, bepalen welke informatie uit het gebruikersprofiel al dan niet mag vrijgegeven worden en melden interessante informatie aan de gebruiker.

6.2 Toekomstig werk

Aan het interactiemodel, dat ontwikkeld werd in deze thesis, kunnen in de toekomst nog enkele aspecten verbeterd of toegevoegd worden. Een tweetal belangrijke uitbreidingen worden hieronder beschreven. Een eerste uitdaging ligt bevat in de geheimhouding. Geheimhouding is moeilijk te realiseren en vergt een grondige kennis van het toepassingsdomein. Op basis van een studie of observatie van het gebruik van dit interactiemodel in het dagelijkse leven, wordt meer kennis vergaard en kunnen tekortkomingen van de huidige geheimhouding aan bod komen. Aan de hand van deze nieuwe kennis kan

de geheimhouding herzien worden en mogelijke verbeteringen aangebracht worden.

De ontwikkelde agenten voor dit interactiemodel zijn eenvoudig en hebben enkel het doel de mogelijkheden van dit interactiemodel aan te tonen. Meer geavanceerde agenten kunnen de gebruiker beter ondersteunen in zijn/haar taken, waardoor ook meer en betere toepassingen mogelijk zijn. Een mogelijke uitbreiding zou kunnen zijn waarbij de agent de huidige context leert in te schatten of zelf veronderstellingen leert maken betreffende de voorkeuren van de gebruiker. Op deze manier kan de inbreng van de gebruiker tijdens de uitvoering van deze implementatie sterk verlaagd worden.

Bibliografie

- [Ama] Amazon. <http://www.amazon.co.uk/>.
- [APB⁺99] Marc Abrams, Constantinos Phanouriou, Alan L. Batongbacal, Stephen M. Williams, and Jonathan E. Shuster. UIML: an appliance-independent XML user interface language. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1695–1708, May 1999.
- [BCdR03] Peter Brusilovsky, Albert T. Corbett, and Fiorella de Rosis, editors. *User Modeling 2003, 9th International Conference, UM 2003, Johnstown, PA, USA, June 22-26, 2003, Proceedings*, volume 2702 of *Lecture Notes in Computer Science*. Springer, 2003.
- [Bel97] Werner Van Belle. Design and implementation of an open mobile multi-agent system, April 1997.
- [Bro] Broadcom. Widcomm bluetooth stack. <http://www.widcomm.com/>.
- [CCP04a] Giovanni Cozzolongo, Berardina De Carolis, and Sebastiano Pizutilo. A personal agent supporting ubiquitous interaction. In Matteo Baldoni, Flavio De Paoli, Alberto Martelli, and Andrea Omicini, editors, *WOA*, pages 55–61. Pitagora Editrice Bologna, 2004.
- [CCP04b] Giovanni Cozzolongo, Berardina De Carolis, and Sebastiano Pizutilo. Supporting personalized interaction in public spaces. In *AIMS '04: Proceedings of the working conference on Artificial Intelligence in Mobile Systems*, September 2004.

- [CCPC04a] Addolorata Cavalluzzi, Berardina De Carolis, Sebastiano Pizzutilo, and Giovanni Cozzolongo. Interacting with embodied agents in public environments. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 240–243, New York, NY, USA, May 2004. ACM Press.
- [CCPC04b] Addolorata Cavalluzzi, Berardina De Carolis, Sebastiano Pizzutilo, and Giovanni Cozzolongo. Interacting with embodied agents in public environments. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 240–243, New York, NY, USA, 2004. ACM Press.
- [CPP03] Berardina De Carolis, Sebastiano Pizzutilo, and Ignazio Palmisano. D-me: Personal interaction in smart environments. In Brusilovsky et al. [BCdR03], pages 388–392.
- [FHM⁺04] A. Ferscha, M. Hechinger, R. Mayrhofer, M. dos Santos Rocha, M. Franz, and R. Oberhauser. Digital Aura. In A. Ferscha, H. Hörtner, and G. Kotsis, editors, *Advances in Pervasive Computing*, volume 176, pages 405–410. Austrian Computer Society (OCG), April 2004. part of the Second International Conference on Pervasive Computing (Pervasive 2004).
- [Fon93] Leonard N. Foner. What’s an agent anyway? - a sociological case study. FTP Report - MIT Media Lab, May 1993.
- [GKS03] Vladimir Gorodetsky, Oleg Karsaev, and Vladimir Samoilov. Multi-agent technology for distributed data mining and classification. *iat*, 00:438, 2003.
- [HBH⁺98] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, WI, July 1998.
- [Hec04] Dominik Heckmann. A specialised representation for ubiquitous computing. *Workshop on User Modelling for Ubiquitous Computing*, 2004.

- [HK03] Dominik Heckmann and Antonio Krüger. A user modeling markup language (userml) for ubiquitous computing. In Brusilovsky et al. [BCdR03], pages 393–397.
- [HNLL04] Jason I. Hong, Jennifer D. Ng, Scott Lederer, and James A. Landay. Privacy risk models for designing privacy-sensitive ubiquitous computing systems. In *DIS '04: Proceedings of the 2004 conference on Designing interactive systems*, pages 91–100, New York, NY, USA, 2004. ACM Press.
- [Hor93] Eric Horvitz. Lumiere project: Bayesian reasoning for automated assistance. <http://research.microsoft.com/~horvitz/lum.htm>, 1993. Microsoft Research.
- [HPS] Inc High Point Software. High point software. <http://www.high-point.com/>.
- [HSB⁺05] Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta von Wilamowitz-Moellendorff. Gu-mo - the general user model ontology. In Liliana Ardissono, Paul Brna, and Antonija Mitrovic, editors, *User Modeling*, volume 3538 of *Lecture Notes in Computer Science*, pages 428–432. Springer, 2005.
- [JT98] Catholijn M. Jonker and Jan Treur. Agent-based simulation of reactive, pro-active and social animal behaviour. In *IEA/AIE '98: Proceedings of the 11th international conference on Industrial and engineering applications of artificial intelligence and expert systems*, pages 584–595, London, UK, 1998. Springer-Verlag.
- [Kay95] Judy Kay. The um toolkit for cooperative user modeling. *User Model. User-Adapt. Interact.*, 4(3):149–196, 1995.
- [Kob01] Alfred Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(1-2):49–63, 2001.
- [LDM02] Scott Lederer, Anind K. Dey, and Jennifer Mankoff. A conceptual model and a metaphor of everyday privacy in ubiquitous. Technical report, University of California at Berkeley, Berkeley, CA, USA, 2002.

- [Led03] Scott Lederer. Designing disclosure: Interactive personal privacy at the dawn of ubiquitous computing. Technical report, University of California at Berkeley, Berkeley, CA, USA, 2003.
- [LHDL04] Scott Lederer, I. Hong, K. Dey, and A. Landay. Personal privacy through understanding and action: five pitfalls for designers. *Personal Ubiquitous Comput.*, 8(6):440–454, 2004.
- [LLCL98] Martha K. Landesberg, Toby Milgrom Levin, Caroline G. Curtin, and Ori Lev. Privacy online: a report to congress. Technical report, Federal Trade Commision, June 1998.
- [LMD03] Scott Lederer, Jennifer Mankoff, and Anind K. Dey. Who wants to know what when? privacy preference determinants in ubiquitous computing. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 724–725, New York, NY, USA, 2003. ACM Press.
- [LSMG⁺00] M. López-Sánchez, F. Martín, J. García, X. Canals, X. Drudis, N. Ruiz, and A. Reyes. Agent communication within a search engine architecture. In *Third Catalan Conference on Artificial Intelligence CCIA*, pages 39–45, 2000.
- [MNLP] Inc. MMV New Line Productions. Lord of the rings. <http://www.lordoftherings.net/effects/>.
- [Nwa96] Hyacinth S. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11(3):1–40, September 1996.
- [Ope] Opennetcf.
- [Orw95] Jon Orwant. Heterogenous learning in the doppelganger user modeling system. *User Model. User-Adapt. Interact.*, 4(2):107–130, 1995.
- [OWL] Web ontology language (owl). <http://www.w3.org/2004/OWL/>.
- [PO05] Vipul Ved Prakash and Adam J. ODonnell. A reputation based approach for efficient filtration of spam. Technical report, Cloudmark Inc., September 2005.

- [RDF] Resource description framework (rdf). <http://www.w3.org/RDF/>.
- [Rod96] Tom Rodden. Populating the application: a model of awareness for cooperative applications. In *CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 87–96, New York, NY, USA, 1996. ACM Press.
- [SL01] Sybil Shearin and Henry Lieberman. Intelligent profiling by example. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 145–151, New York, NY, USA, 2001. ACM Press.
- [SM03] Robbie Schaefer and Wolfgang Müller. Adaptive profiles for multi-modal interaction in intelligent environments. In *Joint Workshop IJCAI Workshop on Artificial Intelligence, Information Access, and Mobile Computing*, Acapulco, Mexico, August 2003.
- [Wei91] Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.
- [Wei96] Mark Weiser. Ubiquitous computing. <http://www.ubiq.com/hypertext/weiser/UbiHome.html>, March 1996.
- [Wik] Wikipedia. <http://wikipedia.org/>.
- [XML] Extensible markup language (xml). <http://www.w3.org/XML/>.
- [XSL] Xsl transformations (xslt). <http://www.w3.org/TR/xslt>.

Auteursrechterlijke overeenkomst

Opdat de Universiteit Hasselt uw eindverhandeling wereldwijd kan reproduceren, vertalen en distribueren is uw akkoord voor deze overeenkomst noodzakelijk. Gelieve de tijd te nemen om deze overeenkomst door te nemen en uw akkoord te verlenen.

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

Digital Aura: Pervasive Human-Human Interaction

Richting: **Master in de informatica**

Jaar: **2006**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Deze toekenning van het auteursrecht aan de Universiteit Hasselt houdt in dat ik/wij als auteur de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij kan reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

U bevestigt dat de eindverhandeling uw origineel werk is, en dat u het recht heeft om de rechten te verlenen die in deze overeenkomst worden beschreven. U verklaart tevens dat de eindverhandeling, naar uw weten, het auteursrecht van anderen niet overtreedt.

U verklaart tevens dat u voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen hebt verkregen zodat u deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal u als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze licentie

Ik ga akkoord,

Carl BRUNINX

Datum: