Accessible and Assistive ICT

VERITAS

Virtual and Augmented **E**nvironments and **R**ealistic User **I**nteractions **T**o achieve Embedded **A**ccessibility Design**S**

Project Number: **247765**

User Model Interoperability Requirements

Deliverable No.		**D1.6.4**	
SubProject No.	1	**SubProject Title**	User Modelling
Workpackage No.	1.6	**Workpackage Title**	Integrated model and intelligent avatar
Activity No.	1.6.4	**Activity Title**	Interoperability requirements and interfaces between user models-CERTH/ITI
Authors		Konstantinos Moustakas, Nikolaos Kaklanis, Dimitrios Tzovaras (CERTH/ITI) , Matthias Peisner (FhG-IAO, MyUI), Michael Lawo (UniBremen-VICON), Yehya Mohamad (Fraunhofer FIT – VICON), Pradipta Biswas (CAM - GUIDE), Volker Hahn (FhG/IGD-GUIDE)	
Status		F (Final)	
Dissemination level		Pu (Public)	
File Name:		VERITAS_D1.6.4_final	
Project start date and duration		01 January 2010, 48 Months	

Table of Contents

List of Figures

List of Tables

Abbreviations list

Abbreviation	Definition
ICT	Information and Communications Technology
ETSI	European Telecommunications Standards Institute
HF	Human Factors
ISO	International Organization for Standardization
W3C	World Wide Web Consortium
OWL	Web Ontology Language
CC/PP	Composite Capability/Preference Profiles
RDF	Resource Description Framework
CSS	Cascading Style Sheets
HTTP	Hypertext Transfer Protocol
XML	Extensible Markup Language
BPMN	Business Process Modeling Notation
BPD	Business Process Diagram
UML	Unified Modeling Language
BPMI	Business Process Management Initiative
XPDL	XML Process Definition Language
WfMC	Workflow Management Coalition
ICF	International Classification of Functioning, Disability and
FMA	Foundational Model of Anatomy Ontology
RULA	Rapid Upper Limb Analysis
UIDL	User Interface Description Language
DISL	Dialog and Interface Specification Language
GIML	Generalized Interface Markup Language
GITK	Generalized Interface Toolkit
XSLT	Extensible Stylesheet Language Transformations
ISML	Interface Specification Meta-Language
RIML	Renderer-Independent Markup Language
SunML	Simple Unified Natural Markup Language
UsiXML	USer Interface eXtensible Markup Language
WSXL	Web Service eXperience Language
SOAP	Simple Object Access Protocol
WSDL	Web Service Definition Language
XICL	eXtensible user-Interface Markup Language
XIML	eXtensible Interface Markup Language
CTT	ConcurTaskTrees
GOMS	Goals, Operators, Methods, and Selection rules
GTA	Groupware Task Analysis
HTA	Hierarchical Task Analysis
TKS	Task Knowledge Structure
FKS	Fundamental Knowledge Structure
TOOD	Task Object-Oriented Description
TCS	Task Control Structure
SCXML	State Chart XML
YAWL	Yet Another Workflow Language
HCI	Human Computer Interaction

Abbreviation	Definition
MHP	Model Human Processor
KLM	Keystroke Level Model
SOAR	State Operator And Result
ACT-R	Adaptive Control of Thought- Rational
EPIC	Executive-Process/Interactive Control
CORE	Constraint-based Optimizing Reasoning Engine
UI	User Interface
SVG	Scalable Vector Graphics
XAML	Extensible Application Markup Language
CEN	European Committee for Standardization
CENELEC	European Committee for Electrotechnical Standardization
MBS	Multibody Systems

Executive summary

The main aim of this document is to provide an analysis of the existing standards, techniques and methodologies, in order to drive the work towards user modelling and standardisation of the ***"Virtual User Modelling and Simulation"*** cluster of EC co-funded projects VERITAS, GUIDE, VICON and MyUI.

At the beginning of the document, a common VUMS Glossary is provided. Then, a detailed analysis of extisting standards related to user modelling is presented. For each standard an overview is initially given followed by a brief analysis of its relevance to user modelling. A comparative table illustrating the major advantages and weaknesses of the existing standards and methodologies in the context of a user modelling framework is presented.

An overview of the most popular User Interface Description Languages (UIDLs) as well as a comparative analysis between them is also presented, due to their great relevance to User Modelling. UIDLs are very relevant to User Modelling as they provide a formal way of describing various aspects of User Interfaces and useful means for adaptation of User Interfaces, according to user needs/preferences.

An overview of the most common task modelling techniques and a comparative analysis between them is provided, due to their great importance on User Modelling. Task Models describe how to perform activities to reach users' goals and can be represented at various abstraction levels. Task models offer the base for the development of new user modelling techniques as user models are often strongly-related with the tasks that the user is able to perform. Then, the most popular physical and cognitive modelling techniques are investigated.

The User Modelling methodology followed by each project (VERITAS, VICON, GUIDE, MyUI) of the VUMS cluster is finally presented in detail, taken into account the common VUMS glossary. The requirements of the VERITAS/VICON/GUIDE/MyUI User Modelling approach of the existing standards, methodologies and techniques is analysed. Moreover, the relevance of each project's approach with existing standards is examined while areas where each project could potentially contribute are identified.

The document concludes with the action plan of VUMS cluster towards standardization that will be followed in the upcoming period in coordination with 24751 "ISO IEC".

Introduction

User model in our case is an explicit representation of the properties of an individual user and can be used to reason about the needs, preferences or future behaviour of that user. Most computer systems that interact with humans contain some kind of implicit model of the user, but there are some difficulties when incorporating a user model into the design process of products and services.

There can be a wide variety of user model types, and models can be classified along the following four dimensions:

- What is modelled: Canonical user (all purpose systems -do your best to accommodate everyone) or Individual user (tailored to the single user)
- Source of modelling information: Model constructed explicitly by the user or Model abstracted by the system on the basis of the user's behaviour
- Time sensitivity of the model: Short-term (highly specific information) or Longer-term (more general information)
- Update methods: Static model or Dynamic model

The update methods often follow from the other three dimensions. In particular, individual user models, models abstracted on the basis of user behaviour and short-term models generally require dynamic update.

If the model contains very short-term information then it can become a task model, since it is relevant to the task at hand, and the individual user is not important. This is because the model will update immediately to reflect any task which a new user undertakes.

The most basic type of model is static and contains a canonical user. This type of model can be embedded within a system and almost does not need to be stored explicitly. In contrast, if the individual user is modeled, then dynamic update is required, and explicit methods are necessary to describe how the user model state affects the system performance.

The forms that a user model may take are as varied as the purposes for which user models are formed. User models may seek to describe:

- The physical processes (mechanics and control) as well as the cognitive processes that underlie the user's actions
- the differences between the user's skills and expert skills
- the user's behavioural patterns or preferences
- the user's characteristics

Another important dimension along which it is important to distinguish approaches is with respect to whether they model individual users or communities of users.

User Modelling process consists of many different aspects, including the interaction of the user with interfaces and devices, the analysis of user tasks and the analysis of user characteristics (sensory, physical and cognitive abilities, psychological and behavioural characteristics).

This document consists of:

- An overview of the most relevant to User Modelling standards and a comparative analysis between them.

- An overview and a comparative analysis of the most widely known User Interface Description languages.
- An overview and a comparative analysis of the most widely known Task Modelling techniques.
- An overview of the VERITAS/VICON/GUIDE/MyUI User Modelling Methodology
- VERITAS/VICON/GUIDE/MyUI requirements from the existing standards
- The potential use of the existing standards in VERITAS/VICON/GUIDE/MyUI
- The potential contribution of VERITAS/VICON/GUIDE/MyUI to the existing standards
- The action plan of VUMS cluster towards standardization

1 Common VUMS Glossary

User Model - An (abstract) user model is a set of user characteristics required to describe the user of a product. The characteristics are represented as variables. The user model is established by the declaration of these variables. It is formally described in a machine-readable and human-readable format. An instantiation of the user model is a user profile.

User Profile - The user profile refers to a machine readable instance of a user model representing a specific user.

Virtual user (VU) - The virtual user is a representation of a user based on a User Profile. It includes components, which are able to interact with other virtual entities e.g. virtual products or software applications. VU's intended for simulation purposes represent the human body as e.g. a kinematic system, a series of links connected by rotational degrees of freedom (DOF) that collectively represent musculoskeletal joints such as the wrist, elbow, vertebra, or shoulder. The basic skeleton of the model is described usually in terms of kinematics. In this sense, a human body is essentially a series of links connected by kinematic revolute joints. Each DOF corresponds to one kinematic revolute joint, and these revolute joints can be combined to model various musculoskeletal joints. A VU also includes low level control loops for muscles (basal ganglia-muscle loops) and higher levels of control in a hierarchical fashion: action (motion planning, cognition etc).

Environmental Model - An Environmental Model is a formal machine-readable set of characteristics used to describe the use environment. It includes all required contextual characteristics besides the user model, the interaction model, the device model, the product and related user tasks.

Device Model - A formal machine-readable representation of the features and capabilities of one or several physical components involved in user interaction. It is important to carefully discriminate between user and device model as they are two kinds of models. Too often they are conflated together, with device properties sprinkled into user profiles and vice versa. The device model expresses capabilities of the device. A given device can be used by many different users and a given user could use different devices and in different ways. By carefully separating the different functionalities of device modelling and user modelling in design scenarios it will be easier to enumerate the attributes for each model and from them develop the matching function and attributes of the adaptation process.

User Agent Capabilities Model – A formal machine-readable representation of the capabilities of the user agent related to user interaction.

Application Model – A formal machine-readable representation of the states, transitions and functions of the application.

User Agent - Any end user software (like browser, or other user interface component) that can

- retrieve and render application content
- invoke request to the User Agent Capabilities Model to modify the application content

Interaction Model - The interaction model is a machine readable representation of the interaction behaviour of an application. The interaction model is maintained UI-agnostic, which means it is independent of the concrete format of user interface output- and input data. Interaction model is often also referred to as abstract user interface model, like for example UIML, UI Socket, XForms, etc. It should be noted that the Interaction model can be used for adaptation of Human Machine Interfaces (HMI) and for simulating the use of an application /product with a virtual user.

Context Model - A machine-readable representation of information that can be used to characterize the situation of an entity. An entity is a person, a place, a device, or a product that is considered relevant to the interaction between a user and an application, including the user and applications themselves. (taken and adapted from Dey & Abowd "Towards a Better Understanding of Context and Context-Awareness")

Simulation - The process that enables the interaction of the virtual user with the application model within an artificial environment. The simulation can be real-time or off-line. Real-time simulation can be performed autonomously or manually, where the operator can interact with the environment from a 1st or 3rd person perspective. Accessibility assessment and evaluation can be performed automatically or subjectively by the operator.

Validation - A model can be defined as follows (ref: Oxford Dictionary)

- a simplified description, especially a mathematical one , of a system or process, to assist calculations and predictions
- a three-dimensional representation of a person or thing or of a proposed structure, typically on a smaller scale than the original.

If the model is mathematical then it needs a statistical validation process. If the model is non-mathematical then it should be validated through qualitative processes. We can standardize the type, process and metrics of validation as follows:

- Type : Qualitative vs Quantitative
- Process: Particular process used to validate (Example: Cross validation, Leave one out validation for quantitative, Grounded theory for qualitative evaluation and so on.)
- Metric: Quantitative vs Qualitative (Example: Task completion time, prediction error for quantitative, subjective preference, quality of life for qualitative metric)

Adaptive User Interfaces - User interfaces that adapt their appearance and/or interaction behaviour to an individual user according to a user profile. In contrast to adaptable user interfaces, which are modified by a deliberate and conscious choice of a user, adaptive user interfaces automatically initiate and perform changes according to an updated user profile. Changes in the user profile can be done manually by the user or can be inferred automatically by the system in a machine learning process during the interaction between the system and an individual user.

User Interface Design Pattern - Approved user interface solution to a recurring design problem. User Interface Design has a formalized description. For the use in adaptive user interfaces, design patterns have a representation in form of reusable software components which can be put together to complete user interfaces during run-time.

2 Relevant Standards and Methodologies

This chapter analyzes Standards related to user modelling along with popular methodologies and languages for user interface description and task modelling. For each individual element an overview is initially given followed by a brief analysis of its relevance to user modelling. Sections 2.1, 2.2 and 2.3 conclude with comparative tables illustrating the major advantages and weaknesses of existing standards and methodologies in the context of a user modelling framework.

2.1 Standards Related to User Modeling

The existing standards related to User Modelling provide guidance to ICT and non-ICT product and service designers on issues and design practices related to Human Factors. They aim to help designers and developers to maximize the level of usability of products and services by providing a comprehensive set of Human Factors design guidelines and meta-models in machine-readable formats.

2.1.1 Standards Related to User Modelling - Overview

2.1.1.1 ETSI TS 102 747; Human Factors (HF); Personalization and User Profile Management; Architectural Framework

ETSI TS 102 747 defines an architectural framework supporting the personalization and user profile management concepts described in EG 202 325. ETSI TS 102 747 addresses issues related to network requirements, functions and procedures. It also covers User Profile security and privacy issues.

Capabilities provided by the architecture are:

- data editing (e.g. creation, templates, update);
- data storage;
- synchronization;
- backup;
- access control respecting user preferences and legal policies;

Profile solutions within the scope of the “ETSI TS 102 747” standard are:

- those provided for the primary benefit of the end-user;
- those which the end-user has rights to manage the profile contents;
- those where the end-user has the right to have a dialogue with the information owning stakeholder.

The key aim of the architecture is to allow many devices to share a single profile, either in full or in part (referred to as a profile component), and to allow some profile data items of the profile to be set depending on the context in which the device or service is operating.

Standard's potential relevance to User Modelling
ETSI TS 102 747 is very relevant to User Modelling as: • It specifies the main requirements of user profile management architecture and provides recommendations in order to fulfill these requirements. • VICON: this is less relevant to the developments in VICON

2.1.1.2 ETSI ES 202 746; Human Factors (HF); Personalization and User Profile Management; User Profile Preferences and Information

The "ETSI ES 202 746" standard specifies information and preferences, which are choices made by the user, that will result in driving the behaviour of the system, and builds on the user profile concept described in EG 202 325. The concept of a user profile usually refers to a set of preferences, information and rules that are used by a device or service to deliver a customized version of capabilities to the user. In practice, most devices and services contain profiles specific to that product and unrelated to any other. This requires that, on change of service or device, the user has to re-educate themselves in how to personalize their services or devices and re-enter their information and preferences. This will result in variable success rate and user satisfaction.

In general, a profile contains:

- *Information*: data about or related to the user (e.g. name, address).
- *Preferences*: choices made by the user about a given parameter that will define or modify the system behaviour. More complex preferences can be expressed in the form of rules (see below).

NOTE: When something is considered essential to the user, it would be more appropriate if a preference is instead called a "need" (e.g. a blind user sets the modality to "sound"). However, for simplification, in the present document the word "preference" is used.

- *Rules*: statements that can be automatically interpreted in order to define or modify the system behaviour.

More specifically, the profile is organized into several blocks. The major organisational units of the profile are:

- *Personal information*: data about or related to the user (e.g. name, address, location).

- *Human centred preferences*: These are the overall preferences that might apply across the user's usage of a wide variety of different devices and services.

  As these preferences are not mapped precisely to specific features of services and devices, they may be presented in ways that must be interpreted before they can be used as the definition for a precise setting for a service or device feature.

- *Service/device category related information and preferences*: The information and preferences in this clause are related to service categories (e.g. Communications services), further sub-categories of the service category (e.g. Real-time communication), and specific services/devices.

Standard's potential relevance to User Modelling
ETSI ES 202 746 is very relevant to User Modelling as: • It specifies user's preferences including needs of people with disabilities. • It specifies device related preferences (including assistive devices, such as Braille, etc.). • It provides UML class diagrams describing the structure of the user profile. • It provides a set of categorized preferences related to disabilities, concerning visual, motor and hearing impairments as well as cognitive and learning difficulties. • MyUI: Many of the profile items listed in the Annex "Preferences related to disabilities" might be included in the MyUI's generic design patterns. However, the preference-centred approach of this standard cannot be incorporated easily as of the dynamic detection of functional constraints in the user interaction.

2.1.1.3 ISO/IEC 24751-1:2008 (Information technology -- Individualized adaptability and accessibility in e-learning, education and training -- Part 1: Framework and reference model)

ISO/IEC 24751 is intended to facilitate the matching of individual user needs and preferences with educational digital resources that meet those needs and preferences. It is intended to address mismatches between personal needs and preferences caused by any number of circumstances, including requirements related to client devices, environments, language proficiency or abilities. The terms and definitions within ISO/IEC 24751 are not judgmental but functional; the purpose is not to point out flaws in educational digital resources with respect to accessibility and adaptability, but to facilitate the discovery and use of the most appropriate content components for each user.

In ISO/IEC 24751, it is recognized that learners experience a disability when there is a mismatch between the learner's needs (or preferences) and the education or learning experience delivered. Disability is therefore not viewed as a personal trait but as a consequence of the relationship between a learner and a learning environment or resource delivery system. An individual who is blind is not disabled when the lesson is delivered in audio, but an individual who does not have the necessary background knowledge to understand the lesson, or an individual who is listening to the lesson in a noisy environment, is disabled. Given this reframing, a learning environment is accessible when learner needs can be addressed or matched (through adaptation, re-aggregation or substitution of digital learning resources). Accessibility is determined by the flexibility of the learning environment (with respect to presentation, control methods, structure, access mode, and learner supports, for example) and the availability of adequate alternative-but-equivalent content and activities. The needs and preferences of a user may arise from the user's context or environment, the technical requirements of the user's device, the tools available (e.g. assistive technologies such as Braille devices, voice recognition systems, alternative keyboards, etc.), the user's background, or a disability in the traditional sense. Accessible systems adjust the user interface or configuration of the learning environment, locate needed resources and adjust the resources to match the characteristics of the resources to the needs and preferences of a user.

This part of ISO/IEC 24751 provides a common framework for additional parts. These additional parts provide two complementary sets of information:

- the description of a learner's accessibility needs and preferences, including
    1) how digital resources are to be displayed and structured,
    2) how digital resources are to be controlled and operated, and
    3) what supplementary or alternative digital resources are to be supplied;
- the description of the characteristics of the resource that affect how it can be perceived, understood or interacted with by a user, including
    1) what sensory modalities are used in the resource,
    2) the ways in which the resource is adaptable (i.e. whether text can be transformed automatically),
    3) the methods of input the resource accepts, and
    4) the available alternatives.

Standard’s potential relevance to User Modelling
ISO/IEC 24751-1:2008 is very relevant to User Modelling as: • It provides a common framework to describe and specify learner needs and

preferences on the one hand and the corresponding description of the digital learning resources on the other hand so that individual learner preferences and needs can be matched with the appropriate user interface tools and digital learning resources. • It provides a UML diagram representing the abstract model of the user needs and preferences as well as the digital resources and resource delivery systems. • It provides a UML process diagram illustrating a possible process for matching a digital resource to user's needs and preferences.

2.1.1.4 ISO/IEC 24751-2:2008 (Information technology -- Individualized adaptability and accessibility in e-learning, education and training -- Part 2: "Access for all" personal needs and preferences for digital delivery)

This part of ISO/IEC 24751 provides a common information model for describing the learner or user needs and preferences when accessing digitally delivered resources or services. This description is one side of a pair of descriptions used in matching user needs and preferences with digital delivery (as described in ISO/IEC 24751-1). This model divides the personal needs and preferences of the learner or user into three categories:

a. Display: how resources are to be presented and structured;

b. Control: how resources are to be controlled and operated; and,

c. Content: what supplementary or alternative resources are to be supplied.

This part of ISO/IEC 24751 is intended to meet the needs of learners with disabilities (as defined in ISO/IEC 24751-1) and of anyone in a disabling context.

The purpose of this part of ISO/IEC 24751 is to provide a machine-readable method of stating user needs and preferences with respect to digitally based education or learning. This part of ISO/IEC 24751 can be used independently, for example to deliver the required or desired user interface to the learner/user, or in combination with ISO/IEC 24751-3 to deliver digital resources that meet a user's needs and preferences.

Standard's potential relevance to User Modelling
ISO/IEC 24751-2:2008 is very relevant to User Modelling as: • It focuses on meeting the needs of learners with disabilities in a disabling context. • It provides a machine-readable method of stating user needs and preferences

with respect to digitally based education or learning. • It provides a detailed model which includes a large set of attributes concerning the human-computer interaction in general as well as in the context of assistive technologies usage.

2.1.1.5 MARIA XML

One important evolution in software applications is the spread of service-oriented architectures in ubiquitous environments. Such environments are characterized by a wide set of interactive devices, with interactive applications that exploit a number of functionalities developed beforehand and encapsulated in Web services.

MARIA XML [32] is a novel model-based UIDL, which can provide useful support both at design and runtime for these types of applications. MariaXML is the successor of TeresaXML in order to support dynamic behaviors, events, rich internet applications, multi-target user interfaces, in particular those based on web services. In this way, it is possible to have a UI specified in MariaXML attached to a web service. At runtime the language is exploited to support dynamic generation of user interfaces adapted to the different devices at hand during the user interface migration process, which is particularly important in ubiquitous environments.

MariaXML is also compatible with the Cameleon Reference Framework [22].

Standard's potential relevance to User Modelling
MARIA XML is quite relevant to User Modelling as: • It supports dynamic generation of user interfaces adapted to different devices. • It is strongly-related to ConcurTaskTrees (CTT), which is a very popular approach for task models development. If a web service is considered as a system task (a task whose performance is entirely allocated to the application), a corresponding task model (as a workflow describing the functionality of the web service) expressed in CTT can be developed. Once the task model has been obtained, it is possible to generate the various UI descriptions in a top-down manner, and then refine them up to the implementation, by using the MARIA tool. • It provides meta-models that describe in detail the design of the user interfaces.

2.1.1.6 W3C Delivery Context Ontology

The Delivery Context Ontology provides a formal model of the characteristics of the environment in which devices interact with the Web or other services. The Delivery

Context includes the characteristics of the Device, the software used to access the service and the Network providing the connection among others.

The Delivery Context is an important source of information that can be exploited to create context-aware applications, thus providing a compelling user experience.

The ontology is formally specified in the Web Ontology Language (OWL). This document describes the ontology and gives details of each term that it contains.

The normative definition of the ontology terms is generated automatically from the OWL file.

Standard's potential relevance to User Modelling
W3C Delivery Context Ontology is very relevant to User Modelling as: • It can be used to adapt web content & applications to make them useable on a wide range of different devices (including assistive devices) with different capabilities. • The Ontology represents a normative, common understanding about the Delivery Context. As such it can be used as a normative reference to create specific vocabularies, while at the same time enabling the interoperability between them. • The Delivery Context Ontology itself constitutes a vocabulary of terms and can be used in conjunction with generic APIs for retrieving Context Properties, such as DCCI[1]. • VICON: the ontologies developed within this standard are relevant to the goals of VICON as they can be partially used to create a vocabulary for the VICON user model • MyUI: the vocabulary of terms included in this standard might be used to extend the existing MyUI model with regard to the incorporation of additional input devices.

2.1.1.7 W3C Composite Capability/Preference Profiles (CC/PP)

A CC/PP profile is a description of device capabilities and user preferences that can be used to guide the adaptation of content presented to that device. "profile" does not refer to a subset of a particular specification, for example the CSS Mobile profile, but refers to the document(s) exchanged between devices that describe the capabilities of a device.

As the number and variety of devices connected to the Internet grow, there is a corresponding increase in the need to deliver content that is tailored to the capabilities of

[1] http://www.w3.org/TR/2007/CR-DPF-20071221/

different devices. Some limited techniques, such as HTTP 'accept' headers and HTML 'alt=' attributes, already exist. As part of a framework for content adaptation and contextualization, a general purpose profile format is required that can describe the capabilities of a user agent and preferences of its user. CC/PP is designed to be such a format.

CC/PP is based on RDF, the Resource Description Framework, which was designed by the W3C as a general purpose metadata description language. RDF provides the framework with the basic tools for both vocabulary extensibility, via XML namespaces, and interoperability. RDF is a standard format for data interchange on the Web. There is a specification that describes the complete semantics of RDF, as well as of the RDF Schema description language. RDF can be serialized into XML using the RDF/XML format as defined in RDFXML. This serialization is used in this document, although other serializations of RDF could be permissible (like Turtle).

A CC/PP profile contains a number of CC/PP attribute names and associated values that are used by a server to determine the most appropriate form of a resource to deliver to a client. It is structured to allow a client to describe its capabilities by reference to a standard profile, accessible to an origin server or other sender of resource data, and a smaller set of features that are in addition to or different than the standard profile. A set of CC/PP attribute names, permissible values and associated meanings constitute a CC/PP vocabulary.

Some information contained in a profile may be sensitive, and adequate trust and security mechanisms must be deployed to protect users' privacy. As a part of a wider application, CC/PP cannot fully cover such issues, but is intended to be used in conjunction with appropriate mechanisms.

It is anticipated that different applications will use different vocabularies; indeed this is needed if application-specific properties are to be represented within the CC/PP framework. But for different applications to work together, some common vocabulary, or a method to convert between different vocabularies, is needed. (XML namespaces can ensure that different applications' names do not clash, but does not provide a common basis for exchanging information between different applications.) Any vocabulary that relates to the structure of a CC/PP 2.0 profile must follow this specification.

CC/PP 2.0 is designed to be broadly compatible with the UAProf 2 specification from the OMA (formerly known as WAP Forum), in the same way CC/PP 1.0 used to accommodate UAProf 1.0 profiles.

CC/PP is compatible with IETF media feature sets (CONNEG) [87] in the sense that all media feature tags and values can be expressed in CC/PP. However, not all CC/PP profiles can be expressed as media feature tags and values, and CC/PP does not attempt to express relationships between attributes.

Standard’s potential relevance to User Modelling
W3C CC/PP is very relevant to User Modelling as:

- It provides a description of device capabilities and user preferences based on RDF that can be used to guide the adaptation of content presented to the specific device.
- VICON: could utilise this standard for describing the device capabilities, which will be simulated in the target VICON platform.

2.1.1.8 The Universal Remote Console - URC Standard (ISO/IEC 24752)

[102] The goal of URC technology is to allow any device or service to be accessed and manipulated by any controller. Users can then select a user interface that fits their needs and preferences, using input and output modalities, and interaction mechanisms that they are familiar with and work well with them. In the following, we refer to the devices and services that are to be controlled as targets, and to the controller devices and their user interfaces as URCs.

To enable URCs to control a target without any prior knowledge of each other some "common understandings" need to be in place.

The first part of ISO/IEC 24752, Part 1: Framework, defines the components of the URC framework and specifies the "common understandings" between them as conformance requirements, stated in terms of high-level interaction. A key part of this interaction is the sharing of control and access information through XML documents.

ISO/IEC 24752 does not determine a specific networking protocol between a URC and a target. It only defines requirements for such a networking platform. The idea is that the URC related interaction could be implemented on top of existing networking platforms that support device discovery, control and eventing — such as UPnP (universal plug and play), Web services, HomePlug, etc.

You can run a URC environment at home and use pluggable user interfaces and similar resources in a constrained environment such as a local network. However, the real power of the URC framework unfolds if applied to a global ecosystem.

In such a scenario, different parties contribute the different parts necessary to build flexible user interfaces: providers of devices and services, providers of network services, providers of controllers, providers of pluggable user interfaces, and the users.

A key component of the URC ecosystem is the resource server, which acts as a marketplace for sharing various resources enabling personalized and pluggable user interfaces. Currently, a pilot resource server is being operated by dot UI (http://dotui.com).

Standard's potential relevance to User Modelling
URC Standard (ISO/IEC 24752) is very relevant to User Modelling as: • It provides a description of personalized access to different devices per re-

mote control standard • VICON: In VICON we aim at the simulation of the access of several equipment e.g. washing machines and mobile devices, here we could consider using this standard for user access per remote control to these devices.

2.1.1.9 IMS Access For All Personal Needs and Preferences Description for Digital Delivery Information Model

This part of the Access For All Specification provides a common information model for describing the learner or user needs and preferences when accessing digitally delivered resources or services. This description is one side of a pair of descriptions used in matching user needs and preferences with digital delivery. This model divides the personal needs and preferences of the learner or user into three categories:

a) Display: how resources are to be presented and structured;

b) Control: how resources are to be controlled and operated; and,

c) Content: what supplementary or alternative resources are to be supplied.

This part of the Access For All Specification is intended to meet the needs of learners with disabilities and of anyone in a disabling context.

The purpose of this part of Access For All Specification is to provide a machine-readable method of stating user needs and preferences with respect to digitally based education or learning. This part of Access For All Specification can be used independently, for example to deliver the required or desired user interface to the learner/user, or in combination with Access For All Specification Digital Resource Description to deliver digital resources that meet a user's needs and preferences.

This document is based upon the original ISO/IEC 24751-1:2008 *Information technology — Individualized adaptability and accessibility in e-learning, education and training — Part 2: "Access For All Personal Needs and Preferences for Digital Delivery"*. The ISO/IEC 24751-1:2008 document was a further development of the original IMS GLC Access For All Learner Information Package Specification, July 2003.

Standard's potential relevance to User Modelling
IMS Access For All Personal Needs and Preferences Description for Digital Delivery Information Model is very relevant to User Modelling as: • It provides a machine-readable method of stating user needs and preferences with respect to digitally based education or learning.

2.1.1.10 ETSI EG 202 116; Human Factors (HF); Guidelines for ICT products and services; "Design for All"

ETSI EG 202 116 provides guidance to Information and Communication Technology (ICT) product and service designers on Human Factors issues, good Human Factors design practice, and relevant international and national standards. In particular, it aims to help designers to maximize the level of usability of products and services by providing a comprehensive set of Human Factors design guidelines.

The guidelines are intended to encourage a "Design for All" approach so as to make products and services accessible to as many people as possible, including elderly people and persons with disabilities, without the need for adaptation or specialized design.

“ETSI EG 202 116” is applicable to ICT products with a user interface that are connectable to all kinds of fixed and mobile telecommunications networks. This includes products such as telephones, Multimedia terminals, Personal digital Assistants (PDAs) and services such as e-mail, Short Message Services (SMS) and voice messaging. It is applicable to public and private access devices and services.

Standard’s potential relevance to User Modelling
ETSI EG 202 116 is very relevant to User Modelling as: • Provides guidelines to make products and services accessible to as many people as possible, including elderly people and persons with disabilities, without the need for adaptation or specialized design. • Describes user characteristics, including sensory, physical and cognitive abilities. • Describes how user abilities are changing over years. • Provides recommendations concerning assistive technologies. • Provides recommendations concerning UI design.

2.1.1.11 ETSI TR 102 068; Human Factors (HF); Requirements for assistive technology devices in ICT

“ETSI TR 102 068” provides guidance on the needs of older and disabled people for assistive technology devices and the requirements for the interconnection of such devices to Information and Communications Technologies (ICT) systems. The report considers devices for user interface input (e.g. keyboards) and output (e.g. display content) as well as speech and video transmission. It reviews available transmission technologies (e.g. Bluetooth and DECT) and requirements for transmission protocols.

It is applicable to assistive technology devices and information and communication devices which have an interface for communicating with a user.

Standard's potential relevance to User Modelling
ETSI TR 102 068 is very relevant to User Modelling as: • It describes user sensory, physical and cognitive disabilities and correlates them with assistive devices.

2.1.1.12 EG 202 325 - Human Factors (HF); User Profile Management

ETSI EG 202 325 provides guidelines relevant to users' needs to manage their profiles for personalisation of services and terminals. Effective user profile management will be critical to the uptake and success of new and advanced communication services and it is therefore important to focus on the users' requirements in this area.

Key areas that are addressed in ETSI EG 202 325 are:

- the user profile concept
- the benefits of user profiles to different parties
- scenarios in which user profiles bring benefits
- administering profiles that reflect users' lifestyles and situations
- administering automatic activation of user profiles
- optimizing the presentation of user profile management guidelines to enable easier compliant product development.

Profile solutions considered to be within the scope of ETSI EG 202 325:

- that are provided for the primary benefit of the user;
- where the user has rights to modify the majority of the profile contents;
- where the user has the right to accept or reject proposed changes to the profile.

"User Profiling" is **not** within the scope of ETSI EG 202 325. "User Profiling" employs profiles:

- that are created and owned by other parties, invisible to the user, to enable the other parties to tailor the services they offer to a user
- where the user is given little or no opportunity to check, modify or reject changes made to the profile
- based on the background collection of information about users derived from their actions.

Standard's potential relevance to User Modelling
ETSI EG 202 325 is pretty relevant to User Modelling as: • It provides guidelines relevant to users' needs to manage their profiles for personalisation of services and terminals. The described guidelines take also into account elderly people and persons with disabilities.

2.1.1.13 BS EN 1332-4:2007 (Identification card systems. Man-machine interface. Coding of user requirements for people with special needs)

This European Standard defines the data objects to be stored within an integrated circuit(s) card and exchanged in order to enable integrated circuit(s) card accepting terminals to identify specific user interface preferences. The preference information may be used by terminals to configure appropriate methods of communicating with the user during a transaction process.

Formats and detailed definitions of single data objects are specified, however, the exact method of storage of data within the integrated circuit(s) card is outside the scope of this European Standard.

BS EN 1332-4:2007 is applicable to the scenario where the cardholder operates the card accepting equipment (e.g. a cash dispenser, ticket machine, vending machine) and to integrated circuit(s) cards conforming to ISO/IEC 7816-4 and ISO/IEC 7816-6 and personalised to the individual cardholder.

Standard's potential relevance to User Modelling
BS EN 1332-4:2007 is pretty relevant to User Modelling as: • It provides a set of detailed definitions of user needs (such as preferred speech output rate, requirement for specific type of fonts, etc.), including people with special needs, for example the aged, minors, people with disabilities, those with learning difficulties, first time users, those not conversant with the local language.

2.1.1.14 ISO 11228-2:2007 (Ergonomics -- Manual handling -- Part 2: Pushing and pulling)

This part of ISO 11228 gives the recommended limits for whole-body pushing and pulling. It provides guidance on the assessment of risk factors considered important to manual pushing and pulling, allowing the health risks for the working population to be evaluated. The recommendations apply to the healthy adult working population and provide reasonable protection to the majority of this population. These guidelines are based on experimental studies of push/pull tasks and associated levels of musculoskeletal loading, discomfort/pain, and endurance/fatigue.

Pushing and pulling, as defined in this part of ISO 11228, is restricted to the following:

- whole-body force exertions (i.e. while standing/walking);
- actions performed by one person (handling by two or more people is not part of the assessment);
- forces applied by two hands;
- forces used to move or restrain an object;
- forces applied in a smooth and controlled way;
- forces applied without the use of external support(s);
- forces applied on objects located in front of the operator;
- forces applied in an upright position (not sitting).

This part of ISO 11228 is intended to provide information for designers, employers, employees and others involved in the design or redesign of work, tasks, products and work organization.

Standard's potential relevance to User Modelling
ISO 11228-2:2007 is very relevant to User Modelling as: • It provides a detailed comparative analysis concerning manual pushing/pulling resulting to recommendations for the healthy adult working population, which provide reasonable protection to the majority of this population. This information is extremely useful when there is need for the development of User Models representing the majority of a population.

2.1.1.15 ISO/DIS 24502 (Ergonomics -- Accessible design -- Specification of age-related relative luminance in visual signs and displays)

ISO/DIS 24502 specifies age-related luminance contrast of any two lights of different colour seen by a person at any age by taking into account the age-related change of spectral luminous efficiency of the eye.

This international standard provides a basic method that is applied to the design of visual signs and displays. It applies to lights seen under moderately bright visual environment called photopic vision and whose spectral radiance is known or measurable. It does not apply to lights seen under darker environment called mesopic or scotopic vision.

ISO/DIS 24502 specifies the luminance contrast for people ranged in age from 10 to 70 years old who have had no medical treatment or surgery on their eyes that may affect their spectral luminous efficiency.

ISO/DIS 24502 does not apply to visual signs and displays seen by people with colour defects whose spectral luminous efficiency is different from those with normal colour vision.

This standard is under development.

Standard's potential relevance to User Modelling
ISO/DIS 24502 is quite relevant to User Modelling as: • It provides a comparative analysis concerning the age-related spectral luminous efficiency (age is defined in decade). This information is pretty useful when there is need for the development of User Models representing a population group of specific age.

2.1.1.16 XPDL (version 2.1)

XPDL is the Serialization Format for BPMN (Business Process Modeling Notation). BPMN is a visual process notation standard from the OMG (Object Management Group), endorsed by WfMC (Workflow Management Coalition), and broadly adopted across the industry. But the BPMN standard defines only the look of how the process definition is displayed on the screen. How you store and interchange those process definitions is outside the scope of the standard, and this is where XPDL comes in. XPDL provides a file format that supports every aspect of the BPMN process definition notation including graphical descriptions of the diagram, as well as executable properties used at run time. With XPDL, a product can write out a process definition with full fidelity, and another product can read it in and reproduce the same diagram that was sent.

Standard's potential relevance to User Modelling
XPDL is quite relevant to User Modelling as: • It provides a file format that supports every aspect of the BPMN process definition notation including graphical descriptions of the diagram, as well as executable properties used at run time. Consequently, it can describe the user tasks represented using BPMN.

2.1.1.17 WHO – International Classification of Functioning, Disability and Health (ICF)

The International Classification of Functioning, Disability and Health, known more commonly as ICF, is a classification of health and health-related domains. These domains are classified from body, individual and societal perspectives by means of two lists: a list of body functions and structure, and a list of domains of activity and participation. Since an individual's functioning and disability occurs in a context, the ICF also includes a list of environmental factors.

The ICF is WHO's framework for measuring health and disability at both individual and population levels. The ICF was officially endorsed by all 191 WHO Member States in the Fifty-fourth World Health Assembly on 22 May 2001(resolution WHA 54.21). Unlike its predecessor, which was endorsed for field trail purposes only, the ICF was endorsed for use in Member States as the international standard to describe and measure health and disability.

The ICF puts the notions of 'health' and 'disability' in a new light. It acknowledges that every human being can experience a decrement in health and thereby experience some degree of disability. Disability is not something that only happens to a minority of humanity. The ICF thus 'mainstreams' the experience of disability and recognises it as a universal human experience. By shifting the focus from cause to impact it places all health conditions on an equal footing allowing them to be compared using a common metric – the ruler of health and disability. Furthermore ICF takes into account the social aspects of disability and does not see disability only as a 'medical' or 'biological' dysfunction. By including Contextual Factors, in which environmental factors are listed ICF allows to records the impact of the environment on the person's functioning.

Standard's potential relevance to User Modelling
WHO ICF is quite relevant to User Modelling as: • It provides classifications related to body functions and structure, and a list of domains of activity. Since an individual's functioning and disability occurs in a context, the ICF also includes a list of environmental factors. • VICON: in VICON we define user abilities according to the classifications of the WHO ICF standard. • MyUI: This standard provides the base for the MyUI user modelling approach.

2.1.1.18 WHO - International Classification of Diseases (ICD)

ICD-10 was endorsed by the Forty-third World Health Assembly in May 1990 and came into use in WHO Member States as from 1994. The classification is the latest in a series which has its origins in the 1850s. The first edition, known as the International List of Causes of Death, was adopted by the International Statistical Institute in 1893. WHO took over the responsibility for the ICD at its creation in 1948 when the Sixth Revision, which included causes of morbidity for the first time, was published. The World Health Assembly adopted in 1967 the WHO Nomenclature Regulations that stipulate use of ICD in its most current revision for mortality and morbidity statistics by all Member States.

The ICD is the international standard diagnostic classification for all general epidemiological, many health management purposes and clinical use. These include the analysis of the general health situation of population groups and monitoring of the incidence and prevalence of diseases and other health problems in relation to other variables such as the characteristics and circumstances of the individuals affected, reimbursement, resource allocation, quality and guidelines.

It is used to classify diseases and other health problems recorded on many types of health and vital records including death certificates and health records. In addition to enabling the storage and retrieval of diagnostic information for clinical, epidemiological and quality purposes, these records also provide the basis for the compilation of national mortality and morbidity statistics by WHO Member States.

Standard’s potential relevance to User Modelling
WHO ICD is quite relevant to User Modelling as: • It provides classifications of diseases and other health problems and it is the international standard diagnostic classification for all general epidemiological, many health management purposes and clinical use.

2.1.1.19 Foundational Model of Anatomy ontology (FMA)2

The Foundational Model of Anatomy Ontology (FMA) is an evolving computer-based knowledge source for biomedical informatics; it is concerned with the representation of classes or types and relationships necessary for the symbolic representation of the phenotypic structure of the human body in a form that is understandable to humans and is also navigable, parse-able and interpretable by machine-based systems. Specifically, the FMA is a domain ontology that represents a coherent body of explicit declarative knowledge about human anatomy. Its ontological framework can be applied and extended to all other species.

Standard’s potential relevance to User Modelling
FMA is quite relevant to User Modelling as: • It is a knowledge source for the building of human body models, which are necessary in our context for building of personas.

2.1.1.20 ISO/IEC 19774 — Humanoid Animation (H-Anim)[3]

With the increased interest in 3D graphics over the past decade there has also been a steady emergence of character modeling software to create and animate 3D human figures. During the same period a number of systems have also been developed for tracking the motions of a "real world" human being. The prevalent obstacle encountered when using multiple of these software packages and systems is in the area of

[2] http://sig.biostr.washington.edu/projects/fm/AboutFM.html
[3] http://h-anim.org/Specifications/H-Anim200x/ISO_IEC_FCD_19774

information exchange. The lack of a standardized skeletal system within this community often forces animation houses and motion capture studios to develop their own proprietary solutions to help smooth the transitions between the systems and software they want to use.

This International Standard specifies H-Anim, an abstract representation for modeling three dimensional human figures. This International Standard describes a standard way of representing humanoids that, when followed, will allow human figures created with modeling tools from one vendor to be animated using motion capture data and animation tools from another vendor

Standard's potential relevance to User Modelling
ISO/IEC 19774 — Humanoid Animation is quite relevant to User Modelling as: • It provides an abstract representation for modeling three dimensional human figures, this can be utilized for building of 3D personas.

2.1.1.21 Anthropometric data out of measurement - ANSUR[4]

This is an anthropometric model out of results of an anthropometric survey of the US Army personnel, which are presented in this report in the form of summary statistics, percentile data and frequency distributions.

These anthropometric data are presented for a subset of personnel (1774 men and 2208 women) sampled to match the proportions of age categories and racial/ethnic groups found in the active duty Army of June 1988. Dimensions given in this report include 132 standard measurements made in the course of the survey, 60 derived dimensions calculated largely by adding and subtracting standard measurement data, and 48 head and face dimensions reported in traditional linear terms but collected by means of an automated headboard designed to obtain three-dimensional data. Measurement descriptions, visual indices and a glossary of terms are included to help identify and locate dimensions.) Descriptions of the procedures and techniques used in this survey are also appearing in this report. These include explanations of the complex sampling plan, computer editing procedures, and strategies for minimizing observer error.

Standard's potential relevance to User Modelling
Anthropometric data out of measurement - ANSUR is quite relevant to User Modelling as: • It provides measurements of the human body, which could be used to build persons and to conduct simulations as required in our context. • VICON: the data gathered in this database lies the foundation for creating

4 http://mreed.umtri.umich.edu/mreed/downloads/anthro/ansur/Gordon_1989.pdf

user profiles for the simulation in virtual environments,

2.1.1.22 Rapid Upper Limb Analysis (RULA)

The Rapid Upper Limb Assessment or RULA method has been developed by Dr. Lynn McAtamney and Professor E. Nigel Corlett, ergonomists at the University of Nottingham in England. The Rapid Upper Limb Assessment (RULA) Analysis can be used to analyze many facets of a personas posture based on a combination of automatically detected variables and user data. Using data derived from the RULA equations, this analysis:

- Considers multiple variables such as object weight, lifting distance, lowering distance, task frequency and action duration.
- Gives the option of adding task-specific variables such as whether the persona is externally supported, if the persona's arms are working across the midline of the body during a task, and whether the persona's feet are balanced and well supported.
- Provides a quantified set of results noting whether the task and posture are acceptable, should be investigated further, should be investigated further but changed soon, or should be investigated further but changed immediately.

With the RULA Analysis you can optimize persona posture in the context of a manual task and therefore design better, and more widely accepted, products and workplaces.

Standard's potential relevance to User Modelling
Rapid Upper Limb Analysis is quite relevant to User Modelling as: • It provides equations, which could be used to analyze the performance of the upper limps of a persona by assessments based on observations and annotations. • VICON: the RULA algorithms are very important for the simulation of users in virtual environments especially grasping of things.

2.1.1.23 Rapid Entire Body Assessment (REBA)

REBA is a postural analysis technique, very similar to RULA (explained in the previous paragraph). More technical notes could be founded in [45].

Comparing with RULA, REBA incorporates dynamic and static postural loading factors, humanload interface (coupling) and a new concept of gravity-assisted upper limb position. The procedure to compute a score for the discomfort assessment is very similar to the method.

Standard's potential relevance to User Modelling
REBA is quite relevant to User Modelling as: • It provides a scoring system for muscle activity caused by static, dynamic, rapid changing or unstable postures.

2.1.1.24 Loading on the Upper Body Assessment (LUBA)

Another type of assessment of the postural discomfort is LUBA (postural Loading on the Upper Body Assessment) [56].

The computing of the index is based on an objective evaluation of the discomfort that a specific posture causes. The outcome of this procedure is the maximum time during which a person could maintain the considered posture.

Standard's potential relevance to User Modelling
LUBA is quite relevant to User Modelling as: • It provides a scoring system which enables the quantitative evaluation of postural stresses for varying postures.

2.1.1.25 Ovako Working-postures Analysis System (OWAS)

This procedure allows checking the dangerous phases for the musculoskeletal system in a specific production cycle. It permits also the quantification of the risk level (according to a defined scale).

This is an observational method. Due to this fact the application of the procedure is quite immediate [101].

Standard's potential relevance to User Modelling
OWAS is quite relevant to User Modelling as: • It enables the acquisition of both ergonomic and yield information and the assessment of data in terms of working activity, job phase or single operator.

2.1.1.26 SNOOK equations

The SNOOK equations can be used to help to optimize manual tasks for our target population and design better, and more widely accepted, products and workplaces.

Carrying Analysis

Based on the SNOOK tables, this analysis of the persona carrying an object based on variables such as weight, distance, frequency and duration.

The SNOOK Carrying Analysis can be used to analyze many facets of manual tasks that contain carrying movements. Using data derived from the internationally accepted SNOOK tables, this analysis:

- Considers multiple variables such as object weight, carrying distance, frequency and duration
- Provides a review of percentage age of population that can fulfill the task

Lifting and Lowering Analysis: SNOOK

These equations provide an analysis of the persona lifting and lowering an object based on variables such as weight, distance, frequency and duration. This analysis uses usually data derived from the SNOOK tables.

The SNOOK Lifting and Lowering Analysis can be used to analyze many facets of manual tasks that contain lifting and lowering movements. Using data derived from the internationally accepted SNOOK tables, this analysis should:

- Consider variables such as weight, lifting and lowering distance, task frequency and duration
- Provide a review of a percentage of population that can fulfil the task
- Use an Initial and Final posture to determine the effort required with performing the task

Pushing and Pulling Analysis: SNOOK

Using data derived from the SNOOK equations, this is an analysis of the personas pushing and pulling an object based on variables such as weight, distance, frequency and duration.

Based on internationally accepted data, the SNOOK Pushing and Pulling Analyses analyzes many facets of manual tasks that contain two-handed pushing and pulling movements. Using data derived from the SNOOK tables, this analysis should:

- Consider variables such as object weight, pushing and pulling distance, frequency and duration
- Provide a review of %'age of population that can fulfill the task

- Provide a recommendation for maximum load during a specific taskLifting and Lowering Analysis: NIOSH 91[5]

- Using data derived from the NIOSH 91 tables, this is an analysis of the persona lifting and lowering an object based on variables such as weight, distance, frequency, grasp quality, and duration.

Use the NIOSH 91 Lifting and Lowering Analyses to analyze many facets of manual tasks that contain lifting and lowering movements. Based on internationally accepted data and using data from the NIOSH 91 tables, this lifting and lower analysis:

- Considers variables such as object weight, lifting/lowering distance, frequency and duration

- Provides a review of %'age of population that can fulfill the task

- Uses Initial and Final Postures to determine the effort associated with performing the lifting/lowering task

Use the NIOSH 91 Lifting and Lowering Analysis to help optimize manual tasks for a target population and therefore design better, and more widely accepted, products and workplaces.

Standard's potential relevance to User Modelling
SNOOK equations are quite relevant to User Modelling as: • They provide equations, which could be used to analyze the performance of many tasks performed by a persona like lifting, lowering and carrying of objects. It can be used for assessments based on observations and annotations. • VICON: These algorithms are necessary for the simulation of carrying and lifting of things.

[5] http://www.purdue.edu/rem/injury/Lifting/LP5360-1991RevisedNIOSHLiftingEquation.pdf

2.1.1.27 ISO/FDIS 9241-129:2010(E); Ergonomics of human-system interaction – Part 129: Guidance on software individualization

The ISO/FDIS 9241-129:2010(E) aims to provide guidance on the application of software individualization in order to achieve as high a level of usability as possible including recommendations on where individualization might be appropriate or inappropriate, and how to apply individualization. Individualization refers to modifications that are achievable without reprogramming the application, because individualization capabilities have already been built into the application.

Individualization involves modifying the behaviour of the interactive system and the presentation of its user interface elements, prior to use or while it is in use, to better meet characteristics of its context of use for an individual or a group of users.

The capability to individualize should be built into an application in response to the identification of user requirements, i.e. different users can have different needs and/or individual users can have different needs at different times.

When a single design solution is not sufficiently usable individualization capabilities may be provided to accommodate one or more of the following:

- Variation in user characteristics
- Different user needs and goals
- Variation in task characteristics
- Different equipment used by a single user
- Different environments experienced by a single user

NOTE: The existence of variability in these factors is not usually sufficient to justify providing individualization without objective data that suggest that individualization will result in the improvement of usability.

This part of ISO 9241 deals only with individualization within the context of designing a complete software system. It is intended to be used with ISO 9241-110 and any other parts in the ISO 9241 series applicable to the design of the intended system. Some of its guidance can also be applied to hardware user interfaces and user interfaces that combine software and hardware.

Standard’s potential relevance to User Modelling
ISO/FDIS 9241-129:2010(E) is relevant to User Modelling as: • It provides guidance on individualization of interface components and interaction activities.

2.1.1.28 EMMA: Extensible MultiModal Annotation Markup Language

EMMA is part of a set of specifications for multimodal systems endorsed by the W3C through their recommendation in February 2009 (http://www.w3.org/TR/emma/); and proposed for the W3C Multimodal Interaction Framework (http://www.w3.org/TR/mmi-framework/).

An XML markup language is provided by EMMA in order to contain and annotate the semantic interpretation of user input gained from various input channels. Annotations and interpretations of user input are supported by a set of elements and attributes.

This standard data interchange format is primarily to be used between the components of a multimodal system, especially those responsible for interpretation and integration of user's input.

Generally, an EMMA document comprises

- instance data: application specific input information
- data model: usually defined by the application including content of an instance and constraints on the structure
- meta-data: annotations linked to the instance data; values are added at run-time

Standard's potential relevance to User Modelling
EMMA is quite relevant to User Modelling as: • It can be used to represent information automatically extracted from a user's input by an interpretation component. • It could be also chosen as a basis for a general semantic result that is carried along and filled out during each stage of processing. • Future systems might use it to convey abstract semantic content to be rendered into natural language.

2.1.2 Standards Related to User Modelling - Comparison

A comparative review of the before-mentioned standards has been performed in order to understand their similarities and differences and also to examine their potential use in the user modelling procedures of the cluster projects. Table 1 presents a comparison of the standards cited in section 2.1.1, according to the following dimensions:

- *Focus on accessibility*: indicates if the standard focuses on people with special needs (provides guidelines for developing accessible products/services, analyzes special needs of people with disabilities, etc.).
- *Tasks support*: indicates if the standard introduces new task models or includes guidelines for developing task models.

- *Workflows support*: indicates if the standard introduces new workflow models or includes guidelines for developing workflows.
- *Description of user needs/preferences*: indicates if the standard describes user needs/preferences using models (meta-models, ontology-schema, UML class diagrams, etc.) or includes guidelines for covering user needs/preferences during products and services design and development. User needs/preferences include:
  - General interaction preferences
  - Interaction modality preferences
  - Multicultural aspects
  - Visual preferences
  - Audio preferences
  - Tactile/haptic related preferences
  - Date and time preferences
  - Notifications and alerts
  - Connectivity preferences
- *Description of device characteristics*: indicates if the standard describes device characteristics or provides guidelines to be followed during the design and development of input/output devices.
- *Description of user characteristics*: indicates if the standard describes user characteristics including sensory abilities (seeing, hearing, touch, taste, smell, balance, etc.), physical abilities (speech, dexterity, manipulation, mobility, strength, endurance, etc.) and cognitive abilities (intellect, memory, language, literacy, etc.). A standard may include definitions of user characteristics, changes of these characteristics with age, analysis of user populations and their characteristics, etc.
- *UI definition support*: indicates if the standard provides guidelines for developing user interfaces or introduces a language for defining user interfaces.
- *Guidelines*: indicates if the standard provides guidelines/recommendations that have to be followed by designers and developers of products and services.
- *Implementation*: indicates if the standard provides meta-models, UML diagrams, ontology schemas, XML schemas, and machine-readable formats in general.

Standard	Focus on accessibility	Tasks support	Workflows support	Description of user needs/ preferences	Description of device charac-teristics	Description of user characte-ristics (physi-cal, cognitive, etc.)	UI defini-tion sup-port	Guidelines	Implementation details
ETSI TS 102 747								✓	
ETSI ES 202 746	✓			✓					✓
ISO/IEC 24751-1:2008	✓								✓
ISO/IEC 24751-2:2008	✓			✓					✓
MARIA XML		✓	✓				✓ (Multimodal)		✓
W3C Delivery Context Ontology					✓				✓
W3C CC/PP				✓	✓				✓
URC Standard (ISO/IEC 24752)	✓	✓		✓	✓		✓		✓
IMS Access For All Personal Needs and Preferences Description for Digital Delivery Information Model	✓			✓					✓
ETSI EG 202 116	✓				✓	✓	✓ (Multimodal)	✓	
ETSI TR 102 068	✓				✓	✓		✓	
ETSI EG 202 325	✓ (limited)							✓	
BS EN 1332-4:2007	✓			✓					✓
ISO 11228-2:2007						✓		✓	
ISO/DIS 24502						✓		✓	
XPDL		✓	✓						✓
WHO ICF	✓								

Standard	Focus on accessibility	Tasks support	Workflows support	Description of user needs/ preferences	Description of device charac-teristics	Description of user characte-ristics (physi-cal, cognitive, etc.)	UI defini-tion sup-port	Guidelines	Implementation details
WHO ICD	✓								
FMA				✓		✓			✓
H-Anim		✓		✓		✓	✓		✓
ANSUR						✓			
RULA	✓					✓		✓	✓
REBA	✓					✓		✓	✓
LUBA	✓					✓		✓	✓
OWAS	✓							✓	✓
SNOOK		✓	✓			✓		✓	✓
ISO/FDIS 9241-129:2010				✓			✓	✓	
EMMA									✓

Table 1 Standards related to User Modelling – Comparison

2.2 User Interface Description Languages (UIDLs)

A user interface description language (UIDL) consists of a specification language that describes various aspects of a user interface under development. In this section we present an overview of UIDLs that have been considered for different reasons: they are available for testing; they have been used in some development cases; they are widely used in general.

UIDLs are very relevant to User Modelling as they provide a formal way of describing various aspects of User Interfaces and useful means for adaptation of User Interfaces according to user needs/preferences.

2.2.1 UIDLs Overview

2.2.1.1 DISL

Dialog and Interface Specification Language (DISL) [89] is a user interface markup language (UIML) subset that extends the language in order to enable generic and modality independent dialog descriptions. Modifications to UIML mainly concerned the description of generic widgets and improvements to the behavioral aspects. Generic widgets are introduced in order to separate the presentation from the structure and behavior, i.e., mainly to separate user- and device-specific properties and modalities from a modalityindependent presentation. The use of generic widget attribute enables to assign each widget to a particular type of functionality it ensures (e.g., command, variable field, text field, etc.). Further, a DISL rendering engine can use this information to create interface components appropriated to the interaction modality (i.e., graphical, vocal) in which the widget will operate. The global DISL structure consists of an optional head element for Meta information and a collection of templates and interfaces from which one interface is considered to be active at one time. Interfaces are used to describe the dialog structure, style, and behavior, whereas templates only describe structure and style in order to be reusable by other dialog components.

2.2.1.2 GIML

The *Generalized Interface Markup Language* (GIML) is used for the generalized Interface Toolkit (GITK) [62]. GIML is used in this context as an interface descriptor.

Following the OMG principles of separation of concerns GIML splits functionality and presentation. While the functionality is preserved in GIML the UI is derived from

XSL files, which come from user and system profiles. This information is merged with the functional descriptions by using XSLT to form a final interface description. The profile data could come directly from a file system or from a remote profile server. GIML avoids the use of concepts such as "push-button", "scrollbar", whereas GIML uses terms such as "action", "data-entry/value- choice/single/limited". The goal is to use interface patterns in the future. These media neutral identifiers are the foundation for an interface object hierarchy.

2.2.1.3 ISML

Interface Specification Meta-Language (ISML) [26] was developed with the intention that metaphors (shared concepts between the user and the computer) be made explicit in design. ISML de-couples that metaphor model from any particular implementation, and express mappings between the concepts shared between the user and the system. It provides a framework that supports mappings between both user-oriented models (such a task descriptions) and software architecture concerns (interactor definitions). The ISML framework composites these concepts within five layers (devices, components, meta-objects, metaphor, interactors), using a variety of mappings to link them together.

2.2.1.4 RIML

Renderer-Independent Markup Language (RIML) [27] is a markup language based on W3C standards that allows document authoring in a device independent fashion. RIML is based on standards such as: XHMTL 2.0 and XFORMS. Special row and column structures are used in RIML to specify content adaptation. Their semantics is enhanced to cover pagination and layout directives in case pagination needs to be done. Due to the use of XForms, RIML is device independent and can be mapped into a XHTML specification according to the target device. RIML semantics is enhanced to cover pagination and layout directives in case pagination needs to be done, in this sense it was possible to specify how to display a sequence of elements of the UI.

2.2.1.5 SeescoaXML

Software Engineering for Embedded Systems using a Component-Oriented Approach (SeescoaXML) [65] consists of a suite of models and a mechanism to automatically produce different final UIs at runtime for different computing platforms, possibly equipped with different input/output devices offering various modalities (e.g. a joystick). This system is context-sensitive as it is expressed first in a modality-independent way, and then connected to a specialization for each specific platform. The context sensitivity of the UI is here focusing on computing platforms variations. An abstract UI is maintained that contains specifications for the different rendering mechanisms (presentation aspects) and their related behavior (dialog aspects). These specifications are written in an XMLcompliant UIDL that is then transformed into platform specific specifications using XSLT transformations. These specifications are then connected to a high-level description of input/output devices. The entry point of this forward engineering approach is therefore located at the level of Abstract UIs.

2.2.1.6 SunML

Simple Unified Natural Markup Language (SunML) [82] is an XML language to specify concrete user interfaces that can be mapped to different devices (PC, PDA, voice). The innovation of this language is the capacity to specify dynamically components. In SunML it is also possible to encapsulate the style and the content of each widget independent of the others. Two different files are used for that purpose. Another interesting feature offered in SunML is widget composition. Some operators have been defined for that purpose: union (semantically-common widgets), intersection, subtraction, substitution, inclusion. Widgets Merging Language (WML) is the extension used for that purpose. SunML presents a reduced set of elements that seems to be not enough, but the composition of widgets is used to specify more complex widgets.

2.2.1.7 TeresaXML

TeresaXML [79] is a UIDL for producing multiple final UIs for multiple computing platforms at design time. They suggest starting with the task model of the system, then identifying the abstract UI specifications in terms of its static structure (the presentation model) and dynamic behavior (the dialog model): such abstract specifications are exploited to drive the implementation. This time, the translation from one context of use to another is operated at the highest level: task and concepts. This allows maximal flexibility, to later support multiple variations of the task depending on constraints imposed by the context of use. Here again, the context of use is limited to computing platforms only. The whole process is defined for design time and not for runtime. For instance, there is no embarked model that will be used during the execution of the interactive system, contrarily to the SEESCOA approach [65]. At the AUI level, the tool provides designers with some assistance in refining the specifications for the different computing platforms considered. The AUI is described in terms of interactors that are in turn transformed into Concrete Interaction Objects (CIOs) once a specific target has been selected.

2.2.1.8 UIML

User Interface Markup Language (UIML) [1] is an XML-based language that provides: (1) a device-independent method to describe a UI, (2) a modality-independent method to specify a UI. UIML allows describing the appearance, the interaction and the connection of the UI with the application logic. The following concepts underlie UIML:

1. UIML is a meta-language: UIML defines a small set of tags (e.g., used to describe a part of a UI) that are modality-independent, target platform-independent (e.g., PC, phone) and target language-independent (e.g., Java, VoiceXML). The specification of a UI is done through a toolkit vocabulary that specifies a set of classes of parts and properties of the classes. Different groups of people can define different vocabularies: one group might define a vocabulary whose classes have a 1-to-1 correspondence to UI widgets in a particular language (e.g., Java Swing API), whereas another group might define a vocabulary whose classes match abstractions used by a UI designer
2. UIML separates the elements of a UI and identifies: (a) which parts are composing the UI and the presentation style, (b) the content of each part (e.g., text, sounds, images) and binding of content to external resources, (c) the behavior of parts expressed as a set of rules with conditions and actions and (d) the definition of the vocabulary of part classes.
3. UIML groups logically the UI in a tree of UI parts that changes over the lifetime of the interface. During the lifetime of a UI the initial tree of parts may dynamically change shape by adding or deleting parts. UIML provides elements to describe the initial tree structure and to dynamically modify the structure.
4. UIML allows UI parts and part-trees to be packaged in templates: these templates may then be reused in various interface designs.

2.2.1.9 UsiXML

USer Interface eXtensible Markup Language (UsiXML) [98] is structured according to different levels of abstraction defined by the Cameleon reference framework [22]. The framework represents a reference for classifying UIs supporting a target platform and a context of use, and enables to structure the development life cycle into four levels of abstraction: task and concepts, abstract UI (AUI), concrete UI (CUI) and final UI (FUI). Thus, the Task and Concepts level is computational-independent, the AUI level is modality-independent (In the cockpit it can be several physical, Vocal, GUI, Tactile) and the CUI level is toolkit independent. UsiXML relies on a transformational approach that progressively moves among levels to the FUI. The transformational methodology of UsiXML allows the modification of the development sub-steps, thus ensuring various alternatives for the existing sub-steps to be explored and/or expanded with new sub-steps. UsiXML has a unique underlying abstract formalism represented under the form of a graph-based syntax.

2.2.1.10 WSXL

Web Service eXperience Language (WSXL) [7] [49] is designed to represent data, presentation and control. WSXL relies on existing standards; in particular, XML based standards such as XPath, XML Events, DOM, XForms and XLink as well as Web Services standards such as SOAP, WSDL and WSFL. WSXL includes an extensible Adaptation Description Language where explicit locations of adaptation points, the permissible operations on adaptation points (e.g. insert, delete, modify), and the constraints on the contents of adaptation (e.g. via an XML Schema) can be specified. The Adaptation Description Language can be used during a post-processing step where the output of a WSXL component can be adapted independently without invoking the component. Finally, a WSXL collection provides an execution and management environment for WSXL components. It calls the lifecycle operations on WSXL components it instantiates, and implements a set of interfaces and a processing model for use by WSXL components and objects external to the collection. An object implementing the WSXL collection interface need not be a WSXL component. The developer can create new and more abstract UI components.

2.2.1.11 XICL

The *eXtensible user-Interface Markup Language* (XICL) [39] is an easy way to develop User Interface Components to Browser-based software. New UI components are created from HTML components and others XICL components. The XICL description is translated into DHTML code. An XICL documents is composed by a UI description composed by HTML or XICL elements and several components (Structure, Properties, Events and Methods. XICL is a language to UI development by specifying its structure and behavior in an abstract level than using only DHTML. It also promotes reuse and extensibility of user interface components.

2.2.1.12 XIML

The *eXtensible Interface Markup Language* (XIML) [29] [30], is a language developed by Redwhale Software, derived from XML and able to store the models developed in MIMIC [84]. MIMIC is a meta-language that structures and organizes interface models. It divides the interface into model components: user-task, presentation, domain, dialog, user, and design models. The design model contains all the mappings between elements belonging to the other models. The XIML is thus the updated XML version of this previous language. The XIML language is mainly composed of four types of components: models, elements, attributes, and relations between the elements. The presentation model is composed of several embedded elements, which correspond to the widgets of the UI, and attributes of these elements representing their characteristics (color, size…). The relations at the presentation level are mainly the links between labels and the widgets that these labels describe. XIML supports design, operation, organization, and evaluation functions; it is able to relate the abstract and concrete data elements of an interface; and it enables knowledge-based systems to exploit the captured data.

2.2.2 UIDLs Comparison

A comparative review of some selected user interface description languages is produced in order to understand their scopes and their differences. Table 2 compares the properties of the different UIDLs according to the following criteria:

- *Models*: This criterion gives the aspects of the UI that can be specified in the description of the UIs.
    - The *task model* is a description of the task to be accomplished by the user,
    - the *domain model* is a description of the objects the user manipulates, accesses or visualizes through the UIs,
    - the *presentation model* contains the static representation of the UI and
    - the *dialog model* holds the conversational aspect of the UI.
- *Tools*: Some of the languages are supported by authoring tools and rendering engines.
- *Supported languages*: Specifies the programming languages to which the XML-based UIDLs can be translated.
- *Supported platforms*: Specifies the computing platform on which the language can be rendered by execution, interpretation or both.

UIDL	Models	Tools	Supported languages	Supported platforms
DISL	Presentation, dialog and control	Rendering engine	VoiceXML, Java MIDP, Java Swing, Visual C++	Mobile and limited devices
GIML	Presentation, dialog, and domain	GITK (Generalized Interface Toolkit)	C++, Java, Perl	Not specified
ISML	Presentation, task, dialog, domain	Under construction	Java, Microsoft foundation class, Java swing classes	Desktop PC, 3D screen
RIML	There is no information	There is no information	XHTML, XFORMS, XEvents, WML	Smart phone, pda, Mobile, Desktop Pc
SeescoaXML	Task, Presentation, dialog	CCOM (BetaVersion 1.0 2002) PacoSuite MSC Editor	Java AWT, Swing, HTML, java.microedition, applet, VoxML, WML Juggler	Mobile, desktop PC, Palm III
SunML	Presentation, dialog, domain	SunML Compiler	Java Swing, voiceXML, HTML, UIML	Desktop Pc
TeresaXML	Presentation, task, dialog	CTTE Tool for task Models Teresa	Markup: Digital TV, VoiceXML, XHTML/SVG, X+V, Programming: C#	DigitalTV, Mobile, Desktop PC
UIML	Presentation, dialog, domain	UIML.net, VoiceXML renderer, WML renderer, VB2UMIL	HTML, Java, C++, VoiceXML, QT, CORBA, and WML	desktop PC, handheld device, tv, mobile
WSXL	Presentation, dialog, domain	Not specified	HTML	PC, Mobile phone
XICL	Presentation, dialog	XICL STUDIO	HTML, ECMAScript, CSS e DOM	Desktop PC
XIML	Presentation, task, dialog, domain	XIML Schema	HTML, java swing, WLM	Mobile, desktop PC, PDA
UsiXML	Presentation, task, dialog, domain	SketchiXML, GraphiXML, FlowiXML, FlasiXML, QtkiXML, InterpiXML	HTML, XHTML, VoiceXML, Java3D, VRML, X3D, XAML, Java, Flash, QTk, WML, XHTML, X+V, C++,	Mobile, Pocket PC, interactive kiosk, wall screen, pda,

Table 2 Properties Comparison of UIDLs [42]

2.3 Task Modelling Techniques

Task Models describe how to perform activities to reach users' goals. The need for modelling is most acutely felt when the design aims to support system implementation as well. If there are only informal representations (such as scenarios or paper mock-ups) available to developers, they would have to make many design decisions on their own, likely without the necessary background, to obtain a complete interactive system. Task models represent the intersection between user interface design and more systematic approaches by providing designers with a means of representing and manipulating an abstraction of activities that should be performed to reach user goals.

Task models can be represented at various abstraction levels. When designers want to specify only requirements regarding how activities should be performed, they consider only the main high-level tasks. On the other hand, when designers aim to provide precise design indications then the activities are represented at a small granularity, thus including aspects related to the dialogue model of a user interface (which defines how system and user actions can be sequenced).

User models are often strongly-related with the tasks that the user is able to perform, so the task models offer the base for the development of new user modelling techniques.

2.3.1 Task Modelling Techniques Overview

2.3.1.1 AMBOSS

The task models developed with AMBOSS [38] describe the hierarchical tree structure of the tasks including the temporal relation between the tasks (formal part of the model) and their description (semi part of the model). On account of this reason that framework shows task models on a semi-formal level. The *task model* is composed of *tasks*, *rooms, roles* and *task relationships*. *Tasks* are, notably, described with attributes such as *name* and *type*. The *name* of the task is generally expressed as a combination of a verb and a substantive (e.g., start decent). The *type* attribute identifies one of the three basic task types:

- interactive, involves an active interaction of the user with the system (e.g., selecting a value, browsing a collection of items)
- system, is an action that is performed by the system (e.g., check a credit card number, display a banner)
- abstract, is an intermediary construct allowing a grouping of tasks of different types

The task also has attributes to determine its *duration* (including its boundaries *minimalDuration* and *maximalDuration* and the time scale used: days, hours, minutes, seconds); the *precondition,* the severity (indicator for the possible damage that arises from this task), the occurrence (the probability that a failure occurs when executing the task), the detection (the likelihood that this failure will be detected), the riskfactor (A riskfactor is an integer that arises of the multiplication of three values severity, occurrence and detection); and additionally it is possible to make a riskfactor write protected (*isWrite-Protected)*.

The following binary *taskRelationship*s are supported in AMBOSS [38]:

- *SEQ*: The subtasks must execute in a fixed sequence from left to right.
- *SER*: The subtasks must execute sequentially but in an arbitrary order.
- *PAR*: The subtasks can start and stop in any order.
- *SIM*: All subtasks have to start in an arbitrary sequence before any task can end. Therefore at least one moment exists where all subtasks are running simultaneously.
- *ALT*: Exactly one randomly selected subtask can execute.
- *ATOM*: The task is the last one in the hierarchy (leaf)

The *decomposition* refers to relationships where a parent task is *decomposed* in subtasks. For each *taskRelationship* the *source* and *target* task must be specified. In addition, *messages* can be transferred from one task to another. A *message* represents the communication between two tasks [38]. It is composed of several attributes like the *name*, the *description*, the *medium* (i.e. electronic, manual, mix)*,* the *contentType* (number, text, graphic, gesture), the *transferType* (synchronous, asynchronous), the *feedback* and the *controlObject* (which ensures the correct delivery of these critical information); also, it contains some flags to determine if the message triggers an action (*isTriggered*), uses a specific protocol *(usesProtocol*), expects feedback (*feedbackNecessary*), is critical (*isCritical*).

A *barrier* determines the protective mechanism of a certain task. It provides the correct execution of the task, which it is assigned to. The assignment is being classified either in safe or unsafe assignment. A barrier consists of an *id*, a *name*, a *description*, a *type* (physical, check, diagnosis, supervision, warning, equipment, procedure, knowledge) and a *purpose* (prevention, control, reduction). Barriers can be active or not, *isActive,* can be activated when the task is started, *isActiveonstart*.

An *objectAMBOSS* is a unity, which is physically available and operating in a running system. An *objectAMBOSS* contains information, which is represented by attributes. It can be either a physical (*isPhysical)* or an informational one (*isInformation)*. Tasks access such objects directly. Also, the objects could be *fixInRoom,* being *allowedinRoom, being forbiddeninRoom,* or has a series of *assignedTask.* A *room* denotes the spatial position of the role involved in the execution of the task. A *room* has different static properties, a unique *id* and *name*, a *description,* a maximum number of persons (*nrPerson)* and a flag that indicates, whether this room isLocked or not, i.e. this area is not available.

Finally, a *role* describes the different actors within the task model. This role is an abstract entity. The expert who starts modelling and does not know at that time who is going to execute the particular task uses this kind of roles. They are responsible for the correct handling of the tasks they are assigned to. There are three predefined roles, the *abstract role* (this means we do not or can not assign to just the task to one role) the *human* and the *system* role. Roles execute tasks and they perform their task in a *room.*

2.3.1.2 ANSI/CEA

ANSI/CEA-2018 is a standard for task model descriptions, which has the potential of significantly improving the usability of computer-controlled electronic products and software interfaces in general. An ANSI/CEA-2018 task model description is an XML document (it is not a graphical formalism) whose syntax and semantics is specified by the standard. The primary use of the XML document is to be interpreted by a device at run-time to guide the user in achieving the described tasks. Detail on this task model can be found in the ANSI/CEA-2018 page[6].

2.3.1.3 ConcurTaskTree (CTT)

In ConcurTaskTrees (CTT) [80] there are five concepts: tasks, objects, actions, operators, and roles. CTT constructors, termed operators, are used to link sibling tasks, on the same level of decomposition. CTT uses a tool (CTTE) for editing the task model used to specify tasks, roles, and objects as well as the task hierarchy with temporal operators. Another feature of CTT is its graphical facility providing means to describe different task types like abstract, cooperative, user, interactive, and application. CTT provides us with means to describe cooperative tasks: a task model will be composed of different task trees: one for the cooperative part and one for each role that is involved in the task. Tasks are further decomposed up to the level of basic tasks defined as tasks that could not be further decomposed. Actions and objects are specified for each basic task. Application objects are mapped onto perceivable objects in order to be presented to the user. Another interesting feature of CTT is the specification of both input and output actions that are associated to an object. Object specification is mainly intended for the specification of UI interaction objects (interactors).

2.3.1.4 Diane +

There are two important points to be made about the way in which Diane+ models a task [9]:

1. The procedures describe only the characteristics specific to an application and do include the standard actions common to most applications, such as quit, cancel, and so on. This assumes that the supposed standard actions, previously defined, really apply to the application of interest. (If a standard action does not apply, this would be indicated.)

2. The described procedures are not mandatory; what is not forbidden is allowed. We note that Diane+ can represent all the constraints of the above specifications. All the algorithmic structures do exist in Diane+, such as ordered sequence, unordered sequence, loop, required choice, free choice, parallelism, default operations, and so on.

[6] http://www.w3.org/2005/Incubator/model-based-ui/wiki/ANSI/CEA-2018

2.3.1.5 GOMS

GOMS developed by [23] is an engineering model for human performance to enable quantitative predictions. By incorporating tables of parameter values that rely on a cognitive architecture, GOMS can be used as an engineering approach to task design [11]. The original GOMS model, referred as CMN-GOMS [23], is the root of a family of models that were elaborated later [51], such as GOMSL (GOMS language) and CPM-GOMS (Critical Path Method GOMS).

Although the first uses a “mental programming language” and is based on a parallel cognitive architecture, the second uses a PERT chart to identify the critical path for computing execution time [10].

In GOMS, the concept of a *method* is essential, as methods are used to describe how tasks are actually carried out. A method is a sequence of operators that describes task performance. Tasks are triggered by goals and can be further decomposed into sub-tasks corresponding to intermediary goals. When several methods compete for the same goal, a selection rule is used to choose the proper one.

Methods describe how goals are actually accomplished. Higher level methods describe task performance in terms of lower level methods, operators, and selection rules. The lowest level of decomposition in GOMS is the unit task, defined by [23] as a task the user really (consciously) wants to perform. Higher level methods use task flow operators that act as constructors controlling task execution.

GOMS makes a clear distinction between tasks and actions. First, task decomposition stops at unit tasks. Second, actions that in GOMS are termed operators are specified by the methods associated with unit tasks. Action modelling varies depending on the GOMS model and the method specification. Operators are cognitive and physical actions the user has to perform in order to accomplish the task goal. Since each operator has an associated execution time (determined experimentally), a GOMS model can help in predicting the time needed to perform a task.

Actions undertaken by the user are specified using external and mental operators. Some special mental operators are flow-control operators that are used to constrain the execution flow. Although the granularity varies according to the purpose of the analysis, it is clear that GOMS is mainly useful when decomposition is done at operational level (i.e., under the unit task level).

2.3.1.6 Groupware task analysis (GTA)

Groupware Task Analysis (GTA) [96] was developed as a means to model the complexity of tasks in a cooperative environment. GTA takes its roots both from ethnography, as applied for the design of cooperative systems and from activity theory adopting a clear distinction between tasks and actions. GTA describes the task world focusing on:

- Agents and roles. Specifying roles and sub-roles that agents play, the relation of responsibility between roles and tasks.

- Work. Involving the decomposition of tasks, the goals and sub-goals, the events that trigger the tasks, and the different strategies used to perform them. A task could be performed by an agent or a role.
- Situation. Specifying the objects used in the task world as well as their structure, the history of past relevant events, and the work environment.

Its framework describes a task world ontology that specifies the relationships between the concepts on which the task world is modeled. Based on this ontology a supporting tool to model task knowledge was also developed: EUTERPE [97].

2.3.1.7 Hierarchical Task Analysis (HTA)

Hierarchical Task Analysis (HTA) [6] was a pioneering method of task analysis. It was primarily aimed at training users to perform particular tasks. On the basis of interviews, user observation, and analysis of existing documents (e.g., manuals, documentation), HTA describes tasks in terms of three main concepts: tasks, task hierarchy, and plans. Tasks are recursively decomposed into subtasks to a point where subtasks are allocated either to the user or the user interface, thus becoming observable. The task hierarchy statically represents this task decomposition. The decomposition stopping criterion is a rule of thumb referred to the $p \times c$ rule. This criterion takes into account the probability of a no satisfactory performance and the cost of a no satisfactory performance (i.e., the consequences it might produce).

Since the task hierarchy does not contain any task ordering, any task should be accomplished according to a plan describable in terms of rules, skills, and knowledge. A plan specifies an ordering in which subtasks of a given task could be carried on, thus acting as a constraint on task performance.

A plan is provided for each hierarchic level. Although the plan is an informal description of temporal relationships between tasks, it is one of the most attractive features of HTA, as it is both simple and expressive. Plans are very close to textual description or to the activity list of traditional task analysis. One advantage of plans is that they do not create any artificial tasks, as some formal notations force analysts' to do to avoid ambiguous specification.

On the other hand, because plans are informal, it is not possible to apply automatic checking of properties such as consistency and reachability.

Any task can be expressed in terms of goals that are reached when the corresponding task is accomplished. Each goal has a status (i.e., latent or active) and conditions to be satisfied. The advantage here in HTA is that goals are independent of the concrete means of reaching them. Therefore, for each goal at any level of decomposition, For each goal, several different operations for reaching the goal can be imagined and specified. Each operation is consequently related to a goal (or goals) and is further specified by the circumstances in which the goal is activated (the input), the activities (action) that contribute to goal attainment, and the conditions indicating the goal has been attained (feedback).

HTA provides a graphical representation of labeled tasks and a plan for each hierarchic level explaining the possible sequences of tasks and the conditions under which each sequence is executed. HTA also supports task analysis for teamwork, as described in [5].

2.3.1.8 Task knowledge structure (TKS)

In Task Knowledge Structure (TKS) method [51], [53], the analysts manipulate a TKS, which is a conceptual representation of the knowledge a person has stored in her memory about a particular task. TKS focuses on:

- *Roles:* A role is assumed to be defined by the particular set of tasks for each an individual is responsible. A person may take on a number of roles and there are tasks associated with each of these roles; or a person could perform similar tasks under different roles.
- *Goal structure:* It identifies the goal and sub-goals contained within the TKS. The goal structure also includes the enabling and conditional states that must prevail if a goal of sub-goal is to be achieved. In this way the goal structure represents a plan for carrying out the task; the plan is carried out through a procedural structure. A procedure is a particular element of behaviour, at the lowest level it can be an action or an object.
- *Taxonomic structure:* Involves action(s) and object(s) knowledge. This includes the representativeness of the object, the class membership, and other attributes such as the procedures in which it is commonly used; its relation to other objects and actions, and its features [51].

TKS was not developed on supporting more than one task at a time, but Johnson and Hyde [51] adapted the basic model and extended it to analyze the collaboration work structure. In order to accommodate collaborative tasks, they considered the mechanics proposed by Pinelle and Gutwin [83]. Their approach is called *Fundamental Knowledge Structures* (FKSs). Metaknowledge and mental models constitute the keystone to an FKS for collaboration. It is postulated that there are three different kinds of knowledge that collaborators possess: 1) general knowledge about what makes for an effective collaboration, 2) individual collaborator's specific knowledge of how they will collaborate to complete the task and an understanding of each collaborator's contribution to the task, and 3) collaborator's knowledge of another collaborator's knowledge. The FKS for collaboration necessarily models high-level knowledge across tasks and consequently is able to generate a set of general requirements for tools to support collaboration across a range of tasks [51].

2.3.1.9 Task Object-Oriented Description (TOOD)

Task Object-Oriented Description (TOOD) consists of an object-oriented method for modelling tasks in the domain of control processes and complex interactive systems, such as those used in air traffic control [66]. The method consists of four steps: hierarchical decomposition of tasks, identification of descriptor objects and world objects, definition of elementary and control tasks, and integration of concurrency (Figure 4). Each task is treated as an instance of a task class identified by a name and an identifier and characterized by a goal, a type (i.e., human, automatic, interactive, and cooperative), the level in the hierarchy, and the total amount of task components. The task body represents a task hierarchy organized using three logical constructors (i.e., AND, OR, and XOR). Each task is then associated with a task control structure (TCS) made up of six classes of descriptor objects that are consumed when the task is carried out and they are aggregated:

1. The *triggering* class has four types of events: formal and informal events, events occurring outside and inside the system.

2. The *condition* class contains contextual conditions governing the performance of the task.

3. The *resource* class describes resources (human or system) required for the task to be performed.

4. The *input data* class specifies information items required for performance of the task. To initialize a task, an input transition expresses logical conditions on these data by sending rules and benefits from various checking functions to ensure that all conditions required to perform the task are fulfilled. For instance, the completeness function checks that all input data are available and satisfy related constraints.

5. The *output data* class specifies information items produced by the task performance. To terminate a task, an output transition expresses logical conditions on these data through synchronization rules and benefits from various checking functions.

6. The *reaction* class describes physical and cognitive results resulting from the task performance.

2.3.1.10 UsiXML

USer Interface eXtensible Language (UsiXML) is a XML-compliant markup language that describes the user interface for multiple contexts of use, it describes a task model [43] where tasks are organized in a high-level of abstraction called processes. A process consists of a number of tasks and a set of relationships among them. The definition of a process indicate which tasks must be performed and in what order. A task can be: user, abstract, interaction or application task. It is decomposed into subtasks to consider hierarchical structure of a task tree; operators are used to link them on the same level of decomposition. A task may manipulate objects through actions. It introduces the concept of Job instead of role. Jobs are the total collection of tasks, duties, and responsibilities assigned to one or more positions which require work of the same nature and level. The job concept allows assembling tasks under a same umbrella in a way that is independent of individual resources in the organization unit. In this way, several individuals could play a particular job, and jobs could be interchanged dynamically. Typically, only resources and their roles within organizations are modeled in most task models, we consider that the place where the tasks are executed is an important aspect in the environment where the collaboration is developed. Thus, the concept of organizational unit is considered, it is a formal group of people working together with one or more shared goals or objectives. It could be composed of other organizational units. Resources are characterized thanks to the notion of user stereotype. But a same task could require other types of resources such as material resources (e.g., hardware, network) or immaterial resources (e.g., electricity, power). The agenda is a list of tasks that are assigned to user stereotype. A user stereotype has one and only one agenda and an agenda belongs to one and only one user stereotype. The concept of agenda is useful to cope with the cooperative aspects. It is also possible to allocate or offer tasks to user stereotypes through the agendas.

2.3.1.11 Statecharts (SCXML)

SCXML stands for State Chart XML: State Machine Notation for Control Abstraction. It is an XML-based markup language which provides a generic state-machine based execution environment based on Harel statecharts. SCXML is able to describe complex state-machines. For example, it is possible to describe notations such as sub-states (hierarchy), parallel states, synchronization, or concurrency, in SCXML. Despite its simplicity, the pluggable and extensible data (XML, JSON) and expression languages (XPath, EcmaScript) along with the event-based communication model provide powerful primitives for representing the behavior of complex reactive systems. For example states may (conditionally) map to domain tasks and expose interfaces and behavior by means of transitions (inherited) in their context. Changes to sub-states may drive the execution status of these task-level states etc.

2.3.1.12 YAWL

YAWL[7] (Yet Another Workflow Language) is a workflow language based on extended Petri nets implementing a variety of workflow patterns. Such it represents a procedural paradigm for describing task flows, their data exchange and automated or manual handling. The language is supported by a software system that includes an execution engine, a graphical editor and a default worklist handler. Production-level uses of the YAWL system include a deployment by first:utility and first:telecom in the UK to automate front-end service processes, and by the Australian film television and radio school to coordinate film shooting processes.

2.3.2 Task Modeling Techniques Comparison

A comparative review of the task modelling techniques cited in section 2.3.1 is presented in Table 3. In order for Table 3 to be fully understandable, it is essential to introduce the available temporal relationships between tasks (some temporal relationships appear in the cells of Table 3).

The temporal relationships between tasks include:

- *Binary relationships* – Binary relationships connect two different tasks.
    - *Enabling* – Enabling relationships specify that a target task cannot begin until the source task is finished.
    - *Disabling* – Disabling relationships refer to a source task that is completely interrupted by a target task.
    - *Suspend/resume* – Suspend resume relationships refer to a source task that can be partially interrupted by a target task and after the target task is completed, the source task will be concluded.

[7] www.yawlfoundation.org

  - *Order independence* – Order independence relationships define that two tasks are independent of the order of execution.

  - *Concurrency with information passing* – Concurrency with Information Passing relationships define that two tasks are executed concurrently and there is passing information between them.

  - *Independent concurrency* – Independent concurrency relationship is a type of temporal relationships where two tasks are executed concurrently but independently and there is no information interchange between them.

  - *Enabling with information passing* – Enabling with information passing relationships specify that a target task cannot be executed until the source task has been executed, and that information produced by the source task is used as an input for the target task.

  - *Cooperation* – Specifies the cooperative relationship between two or more tasks.

  - *Inclusive choice* – An inclusive choice relationship specifies that either both tasks or just one of them or none could be executed.

  - *Deterministic choice* – Deterministic choice relationships refer to two source tasks that could be executed but once one task is initiated, the other cannot be accomplished anymore.

  - *Undeterministic choice* – Undeterministic choice relationships define that two tasks could be both started but once one task is finished, the other cannot be accomplished anymore.

  - *Disabling with information passing* – Disabling with information passing relationships define that one task is completely interrupted by another task and the information produced in the first task is used as an input for the second task.

- *Unary relationships* – Unary relationships are temporal relationships that connect several instances of the same task.

  - *Optional* – Option relationships refer to tasks that are optional.

  - *Iteration* – Iteration relationships indicate tasks that may be iterated.

  - *Finite iteration* – Finite iteration tasks indicate tasks that may be iterated *n* times.

The task modelling techniques cited in section 2.3.1are compared according to the following criteria:

- *Decomposition*: The *decomposition* refers to relationships where a parent task is *decomposed* in subtasks.
- *Sequence*: Defines the type of sequence between different tasks.
- *Iteration*: Iteration relationships indicate tasks which may be executed iteratively.
- *Choice*: Refers to the selective execution of tasks.
- *Optionality*: Optionality refers to tasks whose execution could be optional.
- *Interruption*: Refers to the interruption of a task's execution.
- *Concurrency*: Refers to the concurrent execution of tasks.
- *Cooperation*: Refers to the cooperation between tasks.
- *Parallel*: Refers to the parallel execution of tasks.

The last column ("Relevance") of Table 3 indicates the importance that the criterion of each row has for the task modeling technique.

	AMBOSS	ANSI/CEA	CTT	Diane +	GOMS	GTA	HTA	TKS	TOOD	UsiXML	SCXML	YAWL	Relevance
Decomposition	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Hierarchy	Must
Sequence	Sequence	Ordered sequence, enabling with information passing	Enabling, enabling with information passing	Ordered sequence	Sequence	Sequence	Fixed sequence	Sequence	Sequence	Enabling, enabling with information passing	state (implicit XOR-decomposition)	Sequence	Must
Iteration	X	MinOccurs/ MaxOccurs	Iteration, finite iteration	Loop	Loop (If, then, else)	X	Stop rules	X	X	Iteration, finite iteration	implicit while-loop	implicit while-loop	Should
Choice	ALT	Precodition	Choice	Required choice, free choice	Or (If, then, else)	Or	Selective rule	Or	Choice	Deterministic choice, undeterministic choice, inclusive choice	XOR via guard conditions on transitions	OR, XOR-split	Should
Optionality	Barrier	MinOccurs / MaxOccurs	Optional	Optional	Optional (If, then, else)	Start condition	X	X	X	Optional	via guard conditions on transitions	via branching	Should
Interruption	X	X	Suspend-resume, disabling	X	Interruption (If, then, else)	Stop condition	Stop rules	X	Interruption	Suspend-resume, disabling, disabling with information passing	via outgoing transitions	status change to suspended, canceled etc.	Should

Concur-rency	SER	Ordered	Concur-rent, con-current communi-cating tasks, indepen-dence	unorder ed sequen ce	Concurre ncy (If, then, else)	X	Selectiv e rule	X	Concurren cy	Independ-ent concur-rency, con-currency with infor-mation passing, order inde-pendence	X	multiple task instances	Should
Cooperation	Preconditi on	X	Cooperati ve	X	X	Cooperati on	Teamw ork	Collaborati on	Collaborati on		implicit via events	X	Should
Parallel	PAR, SIM	X	X	Parallel	X	And	Dual task (time sharing)	And	Simultanei ty		parallel	or-split, join with multiple paths enabled	Should

Table 3 Task modeling techniques – Comparison [43]

2.4 Physical Modeling

2.4.1 Multibody Methods

Multibody Methods is a large variety of methods developed to model chains of interconnected bodies. In origin these methods were developed to model machines: for example suspension systems for cars, or a robot arm. Multibody systems (MBS) in their first incarnations were made of rigid bodies and a restricted set of joints (spherical, cylinder, rotary, prismatic etc). However MBS have been later extended including flexible bodies (e.g. the Space Shutlle robot arm is a flexible manipulator), general types of constraints (for example a human knee is not a rotary joint but rather a one degree of freedom constraint, which imposes a neraly rotary motion but actually with differences that cannot be neglected in orthpaedic prothesis [78]). Further later MBS have been extended with models of actuators, sensors and control loops generating methods known as Extended Multibody Systems.

Models of the human body, the mobility of the human body or limbs, strength, models of muscles as actuators have been developed with MBS methodology.

The study of human body motion as a multibody system is a challenging research field that has undergone enormous developments over the last years [91, 70]. Computer simulations of several human capabilities have shown to be quite useful in many research and development activities, such as: (i) analysis of athletic actions, to improve different sports performances and optimization of the design of sports equipment, (ii) ergonomic studies, to assess operating conditions for comfort and efficiency in different aspects of human body interactions with the environment; (iii) orthopaedics, to improve the design and analysis of prosthesis; (iv) occupant dynamic analysis for crashworthiness and vehicle safety related research and design; (v) and gait analysis, for generation of normal gait patterns and consequent diagnosis of pathologies and disabilities [85, 4, 86, 2]. In general, most of the research works developed for simulation of human tasks is based on the assumption that the joints that constrain the relative motion of the system components are considered as ideal or perfect joints. Thus, the physical and mechanical properties of the natural human joints including the effects of friction, lubrication, and intra contact force joints are neglected. Over the last few years, a good number of studies considering the phenomena associated with real joints has been presented [34, 35, 36]. These methodologies are valid for both planar and spatial systems, and have been developed for the most general multibody systems. Consequently, in order to better understand the realistic performance of human body biomechanical systems, it is important to accurately describe the characteristics of the natural human joints, from the simple ones, such as the hip joint, to the most complex ones, such as the knee joint.

2.4.2 Kinematic and Dynamic Analysis of Multibody Systems

In dealing with the study of multibody system motion, two different types of analysis can be performed, namely, the kinematic analysis and the dynamic analysis. The kinematic analysis consists in the study of the system's motion independently of the forces that produce it, involving the determination of position, velocity and acceleration of the system components. In kinematic analysis, only the interaction between the geometry and the motions of the system is analyzed and obtained. Since the interaction between the forces and the system's motion is not considered, the motion of the system needs to be specified to some extent, that is, the kinematic characteristics of some driving elements need to be prescribed, while the kinematic motion characteristics of the remaining elements are obtained using the kinematic constraint equations, which describe the topology of the system.

The dynamic analysis of multibody systems aims at understanding the relationship between the motion of the system parts and the causation of the motion, including external applied forces and moments. The motion of the system is, in general, not prescribed, its calculation being one of the principal objectives of the analysis. The dynamic analysis also provides a way to estimate external forces that depend on the relative position between the system's components, such as the forces exerted by springs, dampers and actuators. Furthermore, it is also possible to estimate the external forces that are developed as a consequence of the interactioni between the system components and the surrounding environment, such as contact-impact and friction forces. The internal reaction forces and moments generated at the kinematic joints are also obtained in the course of the dynamic analysis. These reaction forces and moments prevent the occurrence of the relative motions, in presvribed directions, between the bodies connected via kinematic joints.

2.5 User Behaviour Simulation

Research on simulating user behaviour to predict machine performance was originally started during the Second World War. Researchers tried to simulate operators' performance to explore their limitations while operating different military hardware. During the same time, computational psychologists were trying to model the mind by considering it as an ensemble of processes or programs. McCulloch and Pitts' model of the neuron and subsequent models of neural networks, and Marr's model of vision are two influential works in this discipline. Boden [18] presents a detailed discussion of such computational mental models. In the late 70s, as interactive computer systems became cheaper and accessible to more people, modelling human computer interaction (HCI) also gained much attention. However, models like Hick's Law [44] or Fitts' Law [33] which predict visual search time and movement time respectively were individually not enough to simulate a whole interaction.

The Command Language Grammar [72] developed by Moran at Xerox PARC could be considered as the first HCI model. It took a top down approach to decompose an interaction task and gave a conceptual view of the interface before its implementation. However it completely ignored the human aspect of the interaction and did not model the capabilities and limitations of users. Card, Moran and Newell's Model Human Processor (MHP) [23] was an important milestone in modelling HCI since it introduced the concept of simulating HCI from the perspective of users. It gave birth to the GOMS family of models [23] that are still the most popular modelling tools in HCI.

There is another kind of model for simulating human behaviour that not only works for HCI but also aims to establish a unified theory of cognition. These types of models originated from the earlier work of computational psychologists. Allen Newell pioneered the idea of unifying existing theories in cognition in his famous paper "You can't play 20 questions with nature and win" at the 1973 Carnegie Symposium [76]. Since then, a plethora of systems have been developed that are termed as cognitive architectures and they simulate the results of different experiments conducted in psychological laboratories. Since these models are capable (or at least demanded to be capable) of simulating any type of user behaviour, they are also often used to simulate the behaviour of users while interacting with a computer. Gray and colleagues [41] assert that cognitive architectures ensure the development of consistent models over a range of behavioural phenomena due to their rigorous theoretical basis.

So there are two main approaches of user modelling: the GOMS family of models was developed only for HCI while the models involving cognitive architectures took a more detailed view of human cognition. Based on the accuracy, detail and completeness of these models, Kieras [58] classified them as low fidelity and high fidelity models respectively. These two types of model can be roughly mapped to two different types of knowledge representation. The GOMS family of models is based on goal-action pairs and corresponds to the Sequence/Method representation while cognitive architectures aim to represent the users' mental model [24]. The Sequence/Method representation assumes that all interactions consist of a sequence of operations or generalized methods, while the mental model representation assumes that users have an underlying model of the whole system.

In the following sections, we briefly describe these different types of user model. Then, we present a critical review of existing models and set out the objectives of this research.

2.5.1 The GOMS family of models

GOMS stands for Goals, Operators, Method and Selection. It was inspired by the GPS system [74] developed by Newell. It assumes that people interact with a computer to achieve a goal by selecting a method, which consists of a sequence of basic operations. The GOMS model enables a designer to simulate the sequence of actions of a user while undertaking a task by decomposing the task into goals and sub goals [51]. There are many variations of the original GOMS model.

The Keystroke Level Model (KLM model) [23] simplifies the GOMS model by eliminating the goals, methods, and selection rules, leaving only six primitive operators. They are:

- Pressing a key
- Moving the pointing device to a specific location
- Making pointer drag movements
- Performing mental preparation
- Moving hands to appropriate locations, and
- Waiting for the computer to execute a command.

The durations of these six operations have been empirically determined. The task completion time is predicted by the number of times each type of operation must occur to accomplish the task.

Kieras developed a structured language representation of GOMS model, called NGOMSL (Natural GOMS Language) [60]. Originally, it was an attempt to represent the content of a CCT model at a higher level of notation. CCT is a rule-based system developed by Bovair and colleagues [19] to model the knowledge of users of an interactive computer system. In NGOMSL, the methods of the original GOMS model are represented in terms of production rules of the CCT model. Kieras and colleagues [59] also developed a modelling tool, GLEAN (GOMS Language Evaluation and Analysis), to execute NGOMSL. It simulates the interaction between a simulated user and a simulated device for undertaking a task.

John and Kieras [51] proposed a new version of the GOMS model, called CPM-GOMS, to explore the parallelism in users' actions. This model decomposes a task into an activity network (instead of a serial stream) of basic operations (as defined by KLM) and predicts the task completion time based on the Critical Path Method.

2.5.2 Cognitive architectures

Allen Newell [75] developed the **SOAR (State Operator And Result)** architecture as a possible candidate for his unified theories of cognition. According to Newell [75] and Johnson-Laird [54], the vast variety of human response functions for different stimuli in the environment can be explained by a symbolic system. So the SOAR system models human cognition as a rule-based system and any task is carried out by a search in a problem space. The heart of the SOAR system is its chunking mechanism. Chunking is "a way of converting goal-based problem solving into accessible long-term memory (productions)" [75]. It operates in the following way. During a problem solving task, whenever the system cannot determine a single operator for achieving a task and thus cannot move to a new state, an impasse is said to occur. An impasse models a situation where a user does not have sufficient knowledge to carry out a task. At this stage SOAR explores all possible operators and selects the one that brings it nearest to the goal. It then learns a rule that can solve a similar situation in future. Other studies successfully explained the power law of practice through the chunking mechanism.

However, there are certain aspects of human cognition (such as perception, recognition, motor action) that can better be explained by a connectionist approach than a symbolic one [77]. It is believed that initially conscious processes control our responses to any situation while after sufficient practice, automatic processes are in charge for the same set of responses. Lallement and Alexandre [64] have classified all cognitive processes into synthetic or analytical processes. Synthetic operations are concerned with low level, non decomposable, unconscious, perceptual tasks. In contrast, analytical operations signify high level, conscious, decomposable, reasoning tasks. From the modelling point of view, synthetic operations can be mapped on to connectionist models while analytic operations correspond to symbolic models. Considering these facts, **the ACT-R (Adaptive Control of Thought- Rational) system** [3] does not follow the pure symbolic modelling strategy of the SOAR, rather it was developed as a hybrid model, which has both symbolic and sub symbolic levels of processing. At the symbolic level, ACT-R operates as a rule-based system. It divides the long-term memory into declarative and procedural memory. Declarative memory is used to store facts in the form of 'chunks' and the procedural memory stores production rules. The system works to achieve a goal by firing appropriate productions from the production memory and retrieving relevant facts from the declarative memory. However the variability of human behaviour is modelled at the sub-symbolic level. The long-term memory is implemented as a semantic network. Calculation of the retrieval time of a fact and conflict resolution among rules is done based on the activation values of the nodes and links of the semantic network.

The EPIC (Executive-Process/Interactive Control) [57] architecture pioneers to incorporate separate perception and motor behaviour modules in a cognitive architecture. It mainly concentrates on modelling the capability of simultaneous multiple task performance of users. It also inspired the ACT-R architecture to install separate perception and motor modules and developing the ACT-R/PM system. A few examples of their usage in HCI are the modelling of menu searching and icon searching tasks [46][21].

The CORE system (Constraint-based Optimizing Reasoning Engine) [47][95][31] takes a different approach to model cognition. Instead of a rule-based system, it models cognition as a set of constraints and an objective function. Constraints are specified in terms of the relationship between events in the environment, tasks and psychological processes. Unlike the other systems, it does not execute a task hierarchy; rather prediction is obtained by solving a constraint satisfaction problem. The objective function of the problem can be tuned to simulate the flexibility in human behaviour.

The COSPAL project [93] introduces a model involving both symbolic and sub symbolic processing like the ACT-R system. It combines symbolic reasoning and learning of artificial neural networks (ANN) for association of percepts and states in a bidirectional way to make learning complex task easier. It can model simple arm movements based on object recognition. It also has graphical user interfaces to design and test models. However it has not yet published results on quantitative validation of prediction like previously discussed cognitive architectures.

There exist additional cognitive architectures (such as **Interactive Cognitive Subsystems** [8], **Apex, DUAL, CLARION** [25] etc.), but they are not yet as extensively used as the previously discussed systems.

2.5.3 Advanced models involving GOMS and Cognitive Architectures

The GOMS family of models is mainly suitable for modelling the optimal behaviour (skilled behaviour) of users [51]. These models assume that for each instance of a task execution, the goal and the plan of a user are determined before the execution is started. During execution of a task, a novice first time user or a knowledgeable intermittent user may not have a fixed plan beforehand and can even change goals (or subgoals) during execution of the task. Even expert users do not follow a fixed sequence of actions every time. So the assumptions of the GOMS model may not hold true for many real life interactions. In actuality, these models do not have probabilistic components beyond the feature of selecting the execution time of primitive operators from a statistical distribution in order to model the uncertainty involved in the sub-optimal behaviour of users. As it fails to model the sub-optimal behaviour, it cannot be used to predict the occurrences of different errors during interaction. These problems are common for any Sequence/Method representations since these ways of representations overlook the underlying mental models of users [24].

On the other hand, cognitive architectures model the uncertainty of human behaviour in detail but they are not easily accessible to non psychologists and this causes problem as interface designers are rarely psychologist as well. For example, the ACT-R architecture models the content of a long-term memory in the form of a semantic network, but it is very difficult for an interface designer to develop a semantic network of the related concepts of a moderately complex interface. Developing a sequence of production rules for SOAR or a set of constraints for CORE is equally difficult. The problem in usability issues of cognitive architectures is also supported by the development of the X-PRT system [95] for the CORE architecture. Additionally, Kieras [58] has shown that a high fidelity model cannot always outperform a low fidelity one though it is expected to do so.

Researchers have already attempted to combine the GOMS family of models and cognitive architectures to develop more usable and accurate models. Salvucci and Lee [2003] developed the ACT-Simple model by translating basic GOMS operations (such as move hand, move mouse, press key) into ACT-R production rules. However they do not model the 'think' operator in detail, which corresponds to the thinking action of users and differentiates novices from experts. The model works well in predicting expert performance but does not work for novices.

Blandford and colleagues [17] implemented the Programmable User Model (PUM) [100] by using the SOAR architecture. They developed a program, STILE (SOAR Translation from Instruction Language made Easy), to convert the PUM Instruction Language into SOAR productions. However, this approach also demands good knowledge of SOAR on the part of an interface designer. Later, the PUM team identified additional problems with runnable user models and they are now investigating abstract mathematical models [20].

There also exist some application specific models that combine GOMS models with a cognitive architecture. For example, Gray and Sabnani [40] combined GOMS with ACT-R to model a VCR programming task, while Peck and John [81] used SOAR to model interaction with a help-browser, which ultimately turned out to be a GOMS model.

2.5.4 Models for users with disabilities

There is not much reported work on systematic modelling of assistive interfaces. McMillan [69] felt the need to use HCI models to unify different research streams in assistive technology, but his work aimed to model the system rather than the user. The **AVANTI** project [92] modelled an assistive interface for a web browser based on static and dynamic characteristics of users. The interface is initialized according to static characteristics (such as age, expertise, type of disability and so on) of the user. During interaction, the interface records users' interaction and adapts itself based on dynamic characteristics (such as idle time, error rate and so on) of the user. This model works based on a rule based system and does not address the basic perceptual, cognitive and motor behaviour of users and so it is hard to generalize to other applications. A few researchers also worked on basic perceptual, cognitive and motor aspects. The **EASE** tool [15] simulates effects of interaction for a few visual and mobility impairments. However the model is demonstrated for a sample application of using word prediction software but not yet validated for basic pointing or visual search tasks performed by people with disabilities. Keates and colleagues [55] measured the difference between able-bodied and motor impaired users with respect to the Model Human Processor (MHP) [23] and motor impaired users were found to have a greater motor action time than their able-bodied counterparts. The finding is obviously important, but the KLM model itself is too primitive to model complex interaction and especially the performance of novice users. Gajos, Wobbrock and Weld [37] developed a model in the **SUPPLE** project [37] to estimate the pointing time of disabled users by selecting a set of features from a pool of seven functions of movement amplitude and target width, and then using the selected features in a linear regression model. This model shows interesting characteristics of movement patterns among different users but fails to develop a single model for all. Movement patterns of different users are found to be inclined to different functions of distance and width of targets. The **CogTool** system [50, 52] combines GOMS models and ACT-R system for providing quantitative prediction on interaction. The system simulates expert performance through GOMS modelling, while the ACT-R system [3] helps to simulate exploratory behaviour of novice users [52]. The system also provides GUIs to quickly prototype interfaces and to evaluate different design alternatives based on quantitative prediction [50]. However it does not yet seem to be used for users with disability or assistive interaction techniques. Serna and colleagues [90] used ACT-R cognitive architecture [3] to model progress of Dementia in Alzheimer's patient. They simulated the loss of memory and increase in error for a representative task at kitchen by changing different ACT-R parameters [3]. The technique is interesting but their model still needs rigorous validation through other tasks and user communities.

2.5.5 Comparative Analysis

We have compared different modeling techniques (Table 4) with the help of the following criteria:

Fidelity signifies how detailed the model is. A high fidelity model simulates human behaviour uses more detailed theories of psychology than a low fidelity model.

Ease of use signifies the usability of the model itself.

Perception and Motor action models signify whether the model has separate modules to model perception and motor action in detail. These models are important for simulating performance of visual and mobility impaired users.

Supporting disability signifies whether the model is used to simulate performance of people with disabilities.

Validation for disabled users indicates whether the model has been validated with user trials involving people with disabilities.

	GOMS	SOAR	ACT-R / PM	EPIC	CORE	COG-TOOL	AVANTI	EASE	SUP-PLE	COS-PAL
Fidelity	Low	High	High	High	High	Low	Low	Low	High	High
Ease of use	Easy	Diffi-cult	Diffi-cult	Diffi-cult	Diffi-cult	Easy	Not known	Easy	Not known	Easy
Percep-tion and Motor action models	No	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes
Support-ing disa-bility	Yes	No	Yes	No	No	Just started	Yes	Yes	Yes	No
Validation for dis-abled users	Yes	No	No	No	No	No	Yes	No	Yes	No

Table 4 Comparison of different modelling techniques

3 VERITAS methodology

The VERITAS user modelling methodology is being built based on four major building blocks that are the following:

- Task Models
- Abstract User Models
- Generic Virtual User Models
- Virtual User Models (implemented in VERITAS as personas)

Based on these major components a methodology has been synthesized in the first months of the project that is described in the following sections.

The correspondence between VERITAS terms and VUMS terms is depicted in Table 5.

VERITAS term	VUMS term
Abstract User Model	User Model
Generic Virtual User Model	Similar to User Profile (referring to a set of users having a specific disability)
Virtual User Model	User Profile

Table 5 Correspondence between VERITAS terms and VUMS terns

3.1 VERITAS user modelling methodology outline

Abstract User Models

The Abstract User Models refer to a high level description of potential users. They are developed with respect to several specific disabilities and are broken down according to the disability category, i.e. cognitive user models, physical user models, behavioural & psychological user models. An abstract user model, that will be stored in ontologies, will include several disability related parameters like disability description, disability metrics, ICF functional abilities, etc.

Generic Virtual User Models

A Generic Virtual User Model describes a set of users having a specific set of disabilities. In a Generic Virtual User Model the description is also augmented with actions (primitive tasks) that are affected by the specific set of disabilities. For instance, for users with hemiplegia actions that are affected by the disability could include gait, grasping, etc.

Virtual User Models

A Virtual User Model describes an instance of a virtual user (e.g. Persona) that will be synthesized in the VERITAS “Virtual User Model Generator”. All the disabilities of the user are included in the Virtual User Model as well as the affected actions (primitive tasks). They also include several disability-related parameters describing the severity of the disorder. For instance, the value of the gait cycle for a specific virtual user who suffers from spinal cord injuries, is 2.1 sec, etc.

Task Models

The actions that are being systematically performed in the context of the five VERITAS application areas are described within the task model. Moreover, these tasks are developed using a hierarchical approach. Thus, high level tasks are related to more complex abstract actions, e.g. driving, and are broken down into simpler tasks, e.g. steering, and primitive tasks, e.g. grasping.

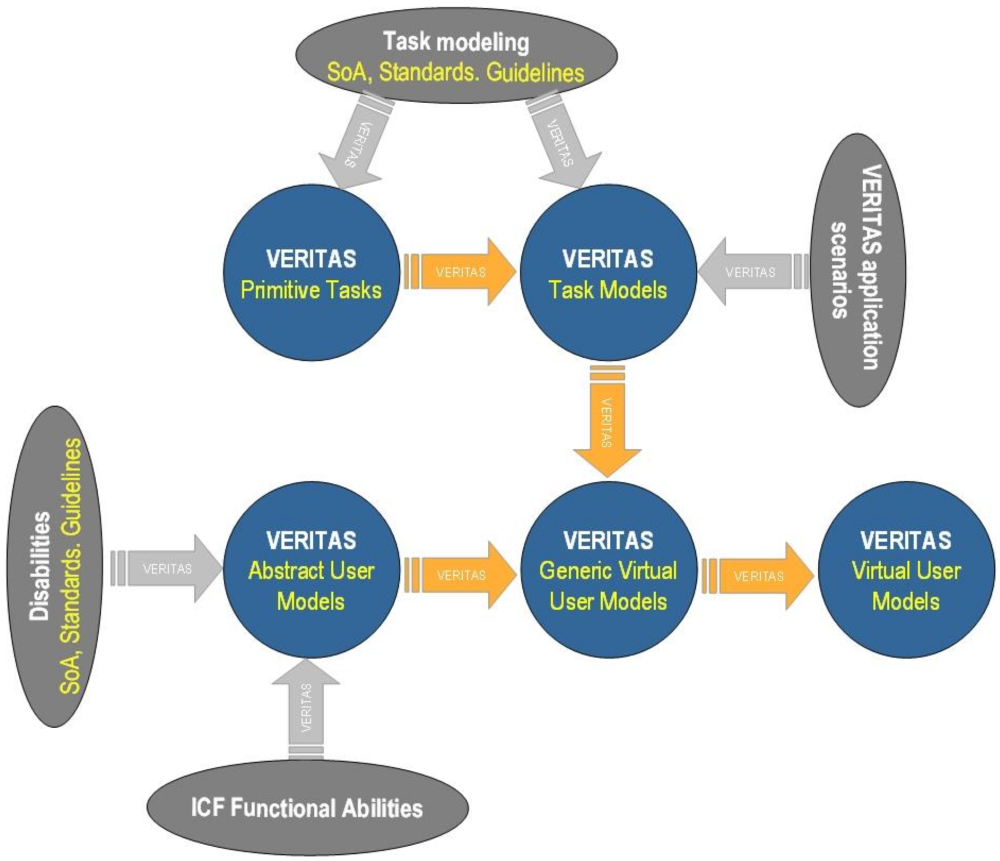


Figure 1 VERITAS User/Task Modeling Methodology

Figure 1 illustrates an outline of the VERITAS user modeling methodology. The development of the user models can be performed in four distinct but interrelated steps:

1. The VERITAS Abstract User Models are initially formed by examining the current state-of-the-art, existing standards and guidelines related to several disabilities. Moreover, this information is augmented utilizing the WHO ICF functional abilities framework.

2. The VERITAS Task Models are developed for the five VERITAS application scenarios and are reflecting the actions that are systematically performed by the users in the context of these scenarios. They follow a hierarchical structure from high level

tasks to low-level primitive tasks. It is very important to have a limited but sufficient number of primitive tasks, since they will be related to disabilities.

3. The Generic Virtual User Models refer to a specific category of virtual users and can be comprised from one or more Abstract User Models, e.g. a Generic Virtual User Model can include the propanopia and hemiplegia disabilities. They also include description on how the specific disabilities affect the execution of specific tasks (primitive or not) that are described in the task models.

4. Finally, the Virtual User Models are instances (virtual users, Personas) of the Generic Virtual User Models and describe a specific virtual user with specific disability related parameters. They will be synthesized in the VERITAS "Virtual User Model Generator" and will follow the needs and requirements of specific accessibility evaluation application scenarios.

As it can be assumed by the above description and methodology, VERITAS follows a top-down approach, where initially at the top-level the Abstract User Models refer to descriptions of specific disabilities, described via ontologies that can be potentially used-extended by the broader community. In the mid-level of the hierarchy the Generic Virtual User Models refer to description of specific classes of disabled user exhibiting the same kind of disability. Here the User Models are also related to the Task Models that refer to actions performed in the five VERITAS application scenarios. The results of this level could also be potentially used by the broader community for applications that are however relevant to the five VERITAS applications scenarios. On the bottom-level of the hierarchy, the Virtual User Models refer to instances generated for a specific application scenario and for specific accessibility evaluation needs and requirements.

In the following a brief description on the definition of the above four basic element is provided, along with simple indicative examples so as to clarify potentially vague issues.

Definition of the primitive tasks

The primitive tasks define the *primitive human actions* and are related to the disability category (physical, cognitive, behavioral). The number of primitive tasks should be limited but also sufficient so as to efficiently model all systematically performed actions in the five VERITAS application scenarios. The degree of primitiveness that will be adopted will be carefully chosen within VERITAS.

Concerning implementation, each primitive task should contain a name as well as the category in which it belongs to. The list of primitive tasks that will be defined within VERITAS will include tasks of different categories such as: motor, cognitive, perceptual, visual, hearing, speech, etc. The following table lists some indicative primitive tasks.

Primitive task's category	Primitive task
Motor	Push
Motor	Grasp
Motor	Pull

Primitive task's category	Primitive task
Motor	Walk
Motor	Sit
Cognitive	Select
Cognitive	Wait
Cognitive	Read
…	…

Table 6 Primitive tasks - Example

Definition of the task models

The Task Models that will be implemented within VERITAS will be based on the five application scenarios. They will refer to user actions/interactions with a specific environment (e.g. car, workplace, user interface) and will also follow a hierarchical structure ranging from abstract high-level task to primitive tasks. They will also support the use of assistive devices to perform a specific task, through multiple instances of a specific action. The Task Models will be developed based on existing relevant stat-of-the-art, standards and guidelines but also based on domain knowledge and relevant attributes with respect to the contents of the VERITAS application scenarios such as automotive, smart living spaces and building, domotics, infotainment, health. The following table lists an indicative instance of a task model.

Sector	Task	Subtask	Primitive tasks	Primitive task – Object
Automotive	Getting in a car	Open door	Grasp	Door handle
			Pull	Door handle
			Pull	Door
		Enter in car	Walk	To car seat
			Sit	On car seat
		Close door	Grasp	Door handle
			Pull	Door
Computer Applications	…	…	…	…
Smart Living Spaces	…	…	…	…
Office Workplace	…	…	…	…
Personal Health-care	…	…	…	…
Infotainment	…	…	…	…

Table 7 Task Models - Example

Definition of the Abstract User Models

The VERITAS Abstract User Models are initially formed by examining the current state-of-the-art, existing standards and guidelines related to several disabilities. In particular, the definition of the Abstract User Models (see D1.3.1) was based on:

- Analysis of existing physical, cognitive and behavioural/psychological user models with disabilities from the state-of-the-art.

- Analysis of accessibility guidelines, methodologies and existing practices such as Human Factors (HF); Guidelines for ICT products and services; "Design for All" methodologies, etc. More specifically, VERITAS will take into account appropriate principles, guidelines and standards for accessibility and universal design for various types of applications, services, goods and infrastructures, which are available from standardization organizations such as Mandate 376 "Accessibility requirements for public procurement of products and services in the ICT domain", the Mandate 420 "Accessibility of the Built Environment", ETSI EG 202 116 V1.2.1 (2002-09) ETSI Guide Human Factors (HF), "Guidelines for ICT products and services - Design for All", and effectively contribute to the simulation of accessibility and usability features.

Ontologies will be used to provide a powerful interoperable and extensible description of the Abstract User Models. An Abstract User Model stored in the ontology will include the type of user disability, user capabilities according to the ICF functional abilities framework, user needs, characteristics from cognitive user models, physical user models, behavioural & psychololological user models, guidelines and standards.

Also, the ontology approach to be followed allows VERITAS to take advantage of the relevant work of ASK-IT, ACCESSIBLE and OASIS projects (developed by partner CERTH/ITI) and which will provide a very good starting point for the project. Additionally, knowledge will be transferred by the PERSONA project (through common partners UPM, ITACA) as well as by the VAALID project.

The advantage of using ontology for specification purposes gives a great potential to the models. Ontology can provide a common basis for communication and collaboration between heterogeneous artefacts and AmI environments. The ontology can describe the basic conceptual terms, the semantics of these terms, and define the relationships among them. A fully semantic description of the Abstract User Models will allow the use of inference engines in the simulation environment. The use of ontologies to describe user models and their interrelationship will also ensure the openness and the accuracy of the models specified. In case that new Abstract User Models should be added, designers can take profit of existing Abstract User Models inheriting the properties of similar ones.

The following table (Table 8) lists an indicative instance of an Abstract User Model.

Disability category	Block (ICD classification)	Code and Pathology (ICD classification)	Short description	Priority L (low) or H (high)	Age related (Yes/No)	Functional limitations (ICF Classification)	Quantitative disability metrics		
							Type	Parameter	Value
Motor Impairments	G20 – G26 Extrapyramidal and movement disorders	G20 – G22 Parkinson's disease	Parkinson's disease is an age-related, progressive, degenerative brain disorder, related to insufficient generation of dopamine in the midbrain. It affects several functions.	H	Yes	b760 Control of voluntary movement functions b7651 Tremor b770 Gait pattern functions	1. Gait parameters Note- Table 4.30 reports data for patients with and without L-dopa.	1.1 Gait speed [m/s]	1.05 (0.05) Table 4.30 Reference: [74] 0.94 (0.20) Table 4.31 Reference: [75]
								1.2 Stride length [m]	1.11 (0.05) Table 4.30 Reference: [74] 0.96 (0.19) Table 4.31 Reference: [75]
								1.3 Cadence [steps/min]	118.3 (15.3) Table 4.31

Disability category	Block (ICD classification)	Code and Pathology (ICD classification)	Short description	Priority L (low) or H (high)	Age related (Yes/No)	Functional limitations (ICF Classification)	Quantitative disability metrics		
							Type	Parameter	Value
									Reference: [75]
								1.4 Gait cycle in double limb support [%]	24.7 (4.8) Table 4.31 Reference: [75]
								1.5 Stride time variability [%]	2.09 (0.16) Table 4.30 Reference: [74]
								1.6 Gait asymmetry	5.17 (1.64) Table 4.30 Reference: [74]
								1.7 Hoehn & Yahr stage	2.3 (0.1) Table 4.30 Reference: [74]
								1.8 UPDRS score	35.8 (2.6) Table 4.30

Disability category	Block (ICD classification)	Code and Pathology (ICD classification)	Short description	Priority L (low) or H (high)	Age related (Yes/No)	Functional limitations (ICF Classification)	Quantitative disability metrics		
							Type	Parameter	Value
									Reference: [74]
							2. Control/dexterity parameters	2.1 Speed [contacts/s]	0.968 (0.399) Table 4.29 Reference: [66]
								2.2 Precision [fraction of hits in A zone]	0.728 (0.257) Table 4.29 Reference: [66]
								2.3 Imprecision [fraction of hits in B, C, D zones]	0.823 (0.204) Table 4.29 Reference: [66]
								2.4 Sureness [contacts/event]	2.288 (0.807) Table 4.29 Reference: [66]
								2.5 Tremor [extra-contacts/event]	0.578 (0.616) Table 4.29 Reference: [66]

Disability category	Block (ICD classification)	Code and Pathology (ICD classification)	Short description	Priority L (low) or H (high)	Age related (Yes/No)	Functional limitations (ICF Classification)	Quantitative disability metrics		
							Type	Parameter	Value
								2.6 Transit time [s]	0.939 (0.462) Table 4.29 Reference: [66]
								2.7 Contact time [s]	0.280 (0.220) Table 4.29 Reference: [66]
								2.8 Fitts' constant [ms]	0.168 (0.071) Table 4.29 Reference: [66]
								2.9 Irregularity [s]	0.181 (0.135) Table 4.29 Reference: [66]
							3. Kinematic parameter of upper body	3.1 Arm/shoulder angles during grasp [deg]	To see reference [68], Fig. 4, pp. 635
								3.2 Wrist	To see reference

Disability category	Block (ICD classifi-cation)	Code and Pathology (ICD classi-fication)	Short description	Priority L (low) or H (high)	Age re-lated (Yes/No)	Functional limitations (ICF Classifi-cation)	Quantitative disability metrics		
							Type	Parameter	Value
									[69] Movement delay 127 ms Time to max grip aperture 631 ms
							4. Fingers	4.1 Flexion amplitude [mm]	44.08 Reference: [70]
								4.2 Extension amplitude [mm]	42.52 Reference: [70]
								4.3 Flexion duration [ms]	190.13 Reference: [70]
								4.4 Extension duration [ms]	206.44 Reference: [70]
							5. Clinical scales and tests	5.1 Unified Parkin-son's Disease Rat-ing Scale UPDRS	Score [52-56]

Disability category	Block (ICD classifi-cation)	Code and Pathology (ICD classi-fication)	Short description	Priority L (low) or H (high)	Age re-lated (Yes/No)	Functional limitations (ICF Classifi-cation)	Quantitative disability metrics		
							Type	Parameter	Value
								5.2 Spiral Analysis	Score [58-60] First order cross-ing, Second order smoothness, mean speed, degree of severity [61,63] WSTS SDDV scores [63]

Table 8 VERITAS Abstract User Models – Example (taken from D1.3.1)

Definition of the Generic Virtual User Models

A Generic Virtual User Model describes the class of Virtual User Models exhibiting one or more specific disabilities. The Generic Virtual User Models will also describe the tasks and primitive tasks affected by the specific disabilities and their associated disability-related parameters.

The following table lists an indicative instance of a Generic Virtual User Model.

Disability category	Disability	Affected primitive tasks (*)	Affected primitive tasks' parameters (*)
Motor	Hemiplegia	Grasp	The user is able to grasp objects, with limited size
		Pull	The user can pull an object with max_Force: 5N
		Walk	Gait velocity ranges from 0.18 to 1.03 m/s
			Abnormal step rhythm
Motor	Hand Osteoarthritis	Grasp	Finger extension and flexion is limited. Grip strength is reduced. (see D1.3.1 table 4.24). Could use alternative grasping (other fingers).
Motor	Muscular dystrophy	…	…
Motor	Multiple sclerosis	…	…
Motor	Rett Syndrome	…	…
Vision	…	…	…
Hearing	…	…	…
Cognitive	…	…	…
Speech	…	…	…

Table 9 Generic Virtual User Models - Example

Definition of the Virtual User Models

The Virtual User Models are instances (virtual users, Personas) of the Generic Virtual User Models and describe a specific virtual user with specific disability related parameters including disabilities, affected primitive tasks and specific affected primitive tasks' parameters for the specific user.

The following table lists an indicative instance of a Virtual User Model.

User ID	Disability category	Disability	Affected primitive tasks	Affected primitive tasks' parameters (*)
User 1	Motor	Hemiplegia	Grasp	The user is able to grasp objects, with size <= *2.5cm* x *2.5cm* x *2.5cm*
			Pull	The user can pull an object with max_Force: 3N
			Walk	Gait velocity : 0.9 m/s
				Abnormal step rhythm
User 2	…	…	…	…

Table 10 VERITAS Virtual User Models – Example

3.2 VERITAS requirements from existing standards

VERITAS aims to utilize, whenever possible, existing standards for the formal and valid implementation of the user models. Specifically, detailed principles, guidelines and recommendations as well as machine-readable formats (meta-models, ontology-schema, etc.) are needed concerning the definition and development of user and task models.

Based on the aforementioned methodology, the following list of requirements of VERITAS from existing standards, has been defined:

- **Accessibility support:** Structured description of abilities and functional limitations for all disability categories and sub-categories. Guidelines and specifications for designers, manufacturers and service providers, for ensuring that their products and environments take into account accessibility limitations (in order to support design for all).

- **User profile description:** Detailed and structured description of user profiles for supporting personalization.

- **Disabled user preferences structured description**: Description of user preferences of the disabled users and the elderly, in the context of Human-Computer interaction.

- **User Interface Modelling**: A formal and efficient way for describing user interfaces. It should be noticed that the user interfaces are not confined within the context of ICT applications but should also describe interactions for non-ICT domains.

- **Multimodal Interfaces Support**: Provision of guidelines and recommendations for the design of multimodal interfaces. Also, provision of a formal way for describing multimodal use interfaces.

- **Assistive devices**: Detailed and structured representation and modelling of assistive devices and their relevance to different types of disabilities.

- **Task modeling support**: Detailed modeling of primitive and complex user tasks, while interacting with ICT and non-ICT interfaces that can be performed in the context of a workflow.

- **Workflow support:** Workflow representation guidelines and methods for describing use interaction while performing specific tasks.

- **Disability-related user interaction characteristics and parameters**: Detailed description of disabled user physical, cognitive, behavioral and psychological characteristics. Includes guidelines, metrics, bounds and limitations that the parameters follow as an effect of the functional limitation of the user.

4 VERITAS in the context of existing standards

The following table summarizes the most relevant to User Modelling standards and methodologies (extensively presented in sections 2.1, 2.2 and 2.3). It also describes their potential use within VERITAS along with potential contributions of VERITAS so as to fill existing standardization gaps.

Standard/Methodology	Potential use in VERITAS	Potential contribution of VERITAS
ETSI TS 102 747; Human Factors (HF); Personalization and User Profile Management; Architectural Framework	VERITAS may follow some of the recommendations provided by this standard (ex. those concerning privacy) for the management of the Virtual User Models.	
ETSI ES 202 746; Human Factors (HF); Personalization and User Profile Management; User Profile Preferences and Information	VERITAS may include the user preferences specified by the standard, especially those related to disabilities to the structure of the Virtual User Models.	VERITAS could possibly extend the user preferences related to disabilities.
ISO/IEC 24751-1:2008 (Information technology - Individualized adaptability and accessibility in e-learning, education and training - Part 1: Framework and reference model)	This standard is not too relevant with the goals of VERITAS. However, it proposes a process for matching digital resources to user's needs and preferences. Some of the steps of the proposed process could provide guidance to VERITAS on how the Virtual User Models could be used by a simulation platform.	
ISO/IEC 24751-2:2008 (Information technology - Individualized adaptability and accessibility in e-learning, education and training - Part 2: "Access for all" personal needs and preferences for digital delivery)	VERITAS may include a subset of the stated in the standard user needs and preferences related to disabilities to the structure of the Virtual User Models.	VERITAS could possibly extend the user preferences related to disabilities.
MARIA XML	One of the requirements of VERITAS is the formal and detailed description of User Interfaces of ICT and non-ICT applications. On the other hand, MARIA XML provides meta-models that describe in detail the design of user interfaces with focus on applications	

Standard/Methodology	Potential use in VERITAS	Potential contribution of VERITAS
	based on web services. Consequently, MARIA XML could be used in VERITAS for the description of user interfaces of such ICT applications.	
W3C Delivery Context Ontology	The Delivery Context Ontology itself constitutes a vocabulary of terms describing different types of devices. This vocabulary can be used in the Abstract User Models as well as in the Virtual User Models wherever special requirements have to be defined for a user concerning the interaction with various devices.	VERITAS could possibly extend the vocabulary of the Delivery Context Ontology in the context of assistive devices.
W3C Composite Capability/Preference Profiles (CC/PP)	VERITAS may use the user preferences specified by the standard, especially those related to disabilities, for the development of the Abstract User Models and Virtual User Models.	VERITAS could possibly extend the user preferences related to disabilities.
IMS Access For All Personal Needs and Preferences Description for Digital Delivery Information Model	This standard provides a machine-readable method of stating user needs and preferences with respect to digitally based education or learning. Some of the stated user needs may also be used out of the scope of e-learning, describing the limitations in interaction of the user with the environment (ex. there are attributes like “speech-rate”, “pitch” and “volume” concerning screen readers). The description of such user needs, as it is proposed by the standard, may be used in the development of VERITAS Abstract and Virtual User Models.	VERITAS could possibly extend the definitions of user needs and link them with specific disabilities.
ETSI EG 202 116; Human Factors (HF); Guidelines for ICT products and services; "Design for All"	ETSI EG 202 116 contains definitions of user characteristics, including sensory, physical and cognitive abilities. These definitions may be used in the development of the VERITAS Abstract User Models, in order	VERITAS could possibly extend the definitions of user characteristics and even create machine-readable formats (ex. XML-schemas) of

Standard/Methodology	Potential use in VERITAS	Potential contribution of VERITAS
	to express how the disabilities are connected with the sensory, physical and cognitive abilities of the user. Additionally, ETSI EG 202 116 describes how user abilities are changing over years. This information could be used in the development of VERITAS Generic Virtual User Models representing users of different age groups.	these characteristics.
ETSI TR 102 068; Human Factors (HF); Requirements for assistive technology devices in ICT	ETSI TR 102 068 describes user sensory, physical and cognitive disabilities and correlates them with assistive devices. VERITAS may use this information in the development of the Abstract User Models, which describe the disabilities, in order to correlate the disabilities with assistive devices.	
EG 202 325 - Human Factors (HF); User Profile Management	EG 202 325 provides guidelines relevant to users' needs to manage their profiles for personalisation of services and terminals. VERITAS may follow some of these guidelines for the better management of the Virtual User Models.	
BS EN 1332-4:2007 (Identification card systems. Man-machine interface. Coding of user requirements for people with special needs)	BS EN 1332-4:2007 provides a set of detailed definitions of user needs (such as preferred speech output rate, requirement for specific type of fonts, etc.), including people with special needs, for example the aged, minors, people with disabilities, those with learning difficulties, first time users, those not conversant with the local language. These user needs may be used in the VERITAS Abstract User Models, presenting this way how user needs are connected with the disabilities.	VERITAS could possibly extend the list of requirements concerning the interaction of elderly and disabled people with a user interface.

Standard/Methodology	Potential use in VERITAS	Potential contribution of VERITAS
ISO 11228-2:2007 (Ergonomics -- Manual handling -- Part 2: Pushing and pulling)	ISO 11228-2:2007 provides structured information illustrating the maximum acceptable forces concerning pushing/pulling for the 90% of the healthy adult working population, according to different parameters (gender, pushing/pulling distance, frequency, etc.). Additionally, many other comparative tables are provided (ex. population subgroup profiles varying in age and gender and reflecting elderly working population (50-64 years)). This information could be taken into account during the development of the VERITAS Generic Virtual User Models.	
ISO/DIS 24502 (Ergonomics -- Accessible design -- Specification of age-related relative luminance in visual signs and displays)	ISO/DIS 24502 provides a comparative analysis concerning the age-related spectral luminous efficiency (age is defined in decade). This information is pretty useful when there is need for the development of User Models representing a population group of specific age and could be taken into account during the development of the VERITAS Generic Virtual User Models.	
XPDL	XPDL may be used for the definition of the VERITAS Task Models.	
WHO ICF	WHO ICF provides classifications related to body functions and structure, and a list of domains of activity, which may be used during the development of the VERITAS Abstract User Models.	

Standard/Methodology	Potential use in VERITAS	Potential contribution of VERITAS
WHO ICD	WHO ICD provides classifications of diseases and other health problems, which may be used during the development of the VERITAS Abstract User Models.	
UsiXML	One of the requirements of VERITAS is the formal and detailed description of multimodal interfaces as well as user tasks. On the other hand, UsiXML provides meta-models that describe in detail the design of multimodal user interfaces, while it also supports task modeling. Consequently, UsiXML could be used in VERITAS for the development of multimodal user interfaces and task models.	VERITAS could possibly extend UsiXML with meta-models describing the users with particular emphasis on elderly and disabled.

Table 11 VERITAS in the context of existing standards

5 VICON methodology

VICON will utilize mainly the User Profiles. The focus in VICON lies on the User Profiles as our direct target users are designers, who will utilize personas to validate their designs of mobile devices and washing machines at various phases of the design process. The following steps explain where guidelines and standards will be utilized in VICON:

- Build a web based virtual user recommendation system on inclusive design for mobile phones and washing machines to assist designers in the scratch phase utilizing existing guidelines, standards and materials. Here guidelines about design for all will be synthesized and made available for the designer.
- Create virtual human environments, where the designers could configure a virtual human "persona". The User Profiles will be used for this purpose.
- Design products in a CAD system, where the Designer can invoke a context sensitive recommendation system. Standards about Device characteristics will be used o create meta device models.
- Build a virtual environment. Context profiles will be used to describe the environment.
- Import the CAD design into the virtual environment.
- Position the persona in the environment
- Assign tasks to the persona. Task standards and workflow standards will be used to define the tasks and the workflow of their conduction.
- Analyze how the human performs. Analysis equations of the assigned tasks will be utilized for this purpose.

The correspondence between VICON terms and VUMS terms is depicted in Table 12.

VUMS terms	VICON correspondence
User Model	Almost same though it is limited to elderly users with vision, hearing and dexterity disabilities
User Profile	Virtual user model
Virtual user	Same as in VUMS
Environmental Model	Almost Same as VUMS.
Device Model	Same as VUMS
User Agent	Has not used explicitly yet
Interaction Model	Task model
Context Model	Same as VUMS
Simulation	Same as VUMS

VUMS terms	VICON correspondence
Validation	Same as VUMS
Adaptive User Interfaces	Same as VUMS
User Interface Design Pattern	Has not used explicitly yet

Table 12 Correspondence between VICON terms and VUMS terns

5.1 VICON user modelling methodology outline

Similar to section 3.1 of VERITAS, VICON will use at most the User Profiling approach.

Additionally VICON will use environments models to describe the devices and contexts, where the persona will reside and interact with.

The actions that are being systematically performed in the context of the two VICON application areas are described within the task model. Moreover, these tasks are developed using an hierarchical approach. Thus, high level tasks are related to more complex abstract actions, e.g. driving, and are broken down into simpler tasks, e.g. steering, and primitive tasks, e.g. grasping.

Figure 2 illustrates an outline of the VICON user modeling methodology. The development of the user models can be performed in many interrelated steps:

- The user models will describe the mild to moderate physical disabilities, which are targeted by the VICON project
- The Generic User Profiles will build the base for the creation of the user categories of the elderly people with the disabilities (namely vision hearing and dexterity disabilities) in the focus of the VICON project. Furthermore it will contain related tasks, which could be performed by the user categories.
- The User Profiles are instances (Virtual users, Personas) of the Generic User Profiles and describe a specific virtual user with specific disability related parameters. It will contain as well kinematic parameters. They will be synthesized in the VICON and be available to be integrated into the virtual environment.

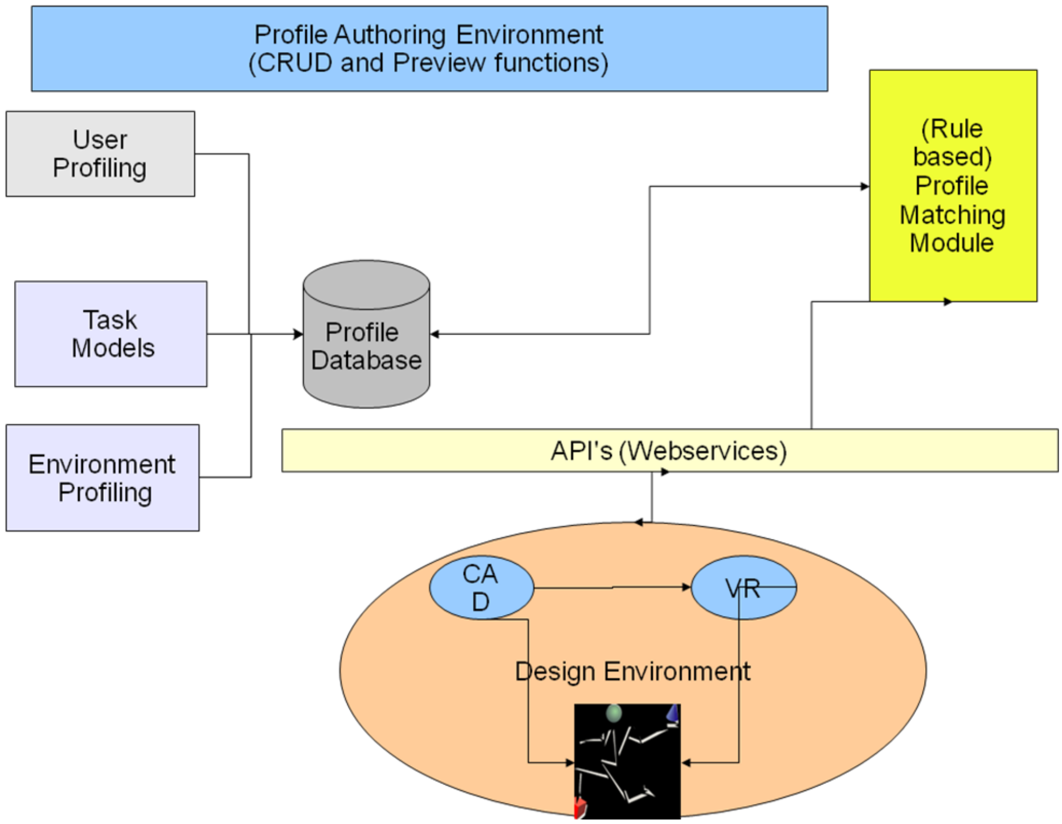


Figure 2: Illustration of the models required in the VICON context

5.1.1 Definition of the primitive tasks

The primitive tasks define the *primitive human actions* and are related to physical disabilities. The number of primitive tasks should be limited but also sufficient so as to efficiently model all systematically performed actions in the two VICON application scenarios.

Concerning implementation, each primitive task should contain a name as well as the category in which it belongs to. The list of primitive tasks that will be defined within VICON will include tasks of different categories such as: motor, visual and hearing. Table 13 lists some indicative primitive tasks.

5.1.2 Definition of the task models

The Task Models that will be implemented within VICON will be based on the two application scenarios. They will refer to user actions/interactions with a specific environment (e.g. washing machine and mobile phone) and will also follow a hierarchical structure ranging from abstract high-level task to primitive tasks. They will also support the use of assistive devices to perform a specific task, through multiple instances of a specific action. The Task Models will be developed based on existing relevant state-of-the-art, standards and guidelines but also based on domain knowledge and relevant attributes with respect to the contents of the VICON application scenarios. The following table lists an indicative instance of a task model.

Sector	Task	Subtask	Primitive tasks	Primitive task – Object
Washing machines	Conduct a washing programme	Opening the door	Grasp	Door handle
			Pull	Door handle
			Pull	Door
		Loading the washing machine	Lift / Lower	Clothes
			Carry	Clothes basket
		Close door	Grasp	Door handle
			Pull	Door
Mobile phones	Turning the phone on and off	Push button	Push	On/Off button of a mobile phone

Table 13 VICON Task Models - Example

5.1.3 Definition of the User Models

The VICON User Models are initially formed by examining the current state-of-the-art, existing standards and guidelines related to several disabilities. In particular, the definition of the User Models will be based on a similar approach as described in the VERITAS approach see Section 3.1 especially table Table 8. VICON focuses on mild to moderate physical disabilities.

5.1.4 Definition of the Generic User Profiles

See definition in Section 3.1

5.1.5 Definition of the User Profiles

See Section 3.1.

5.2 VICON requirements from existing standards

Similar to VERITAS, VICON aims to utilize, whenever possible, existing standards for the formal and valid implementation of the user models.

Based on the aforementioned methodology, the following list of requirements of VICON from existing standards has been defined:

- **Accessibility support:** Structured description of abilities and functional limitations for all disability categories and sub-categories. Guidelines and specifications for designers, manufacturers and service providers, for ensuring that their products and environments take into account accessibility limitations (in order to support design for all).

- **Multimodal Interfaces Support**: Provision of guidelines and recommendations for the design of multimodal interfaces. Also, provision of a formal way for describing multimodal use interfaces.

- **Task modeling support**: Detailed modeling of primitive and complex user tasks, while interacting with ICT and non-ICT interfaces that can be performed in the context of a workflow.

- **Workflow support:** Workflow representation guidelines and methods for describing use interaction while performing specific tasks.

- **Disability-related user interaction characteristics and parameters**: Detailed description of disabled user physical characteristics. Includes guidelines, metrics, bounds and limitations that the parameters follow as an effect of the functional limitation of the user.

6 VICON in the context of existing standards

The following table summarizes the most relevant to User Modelling standards and methodologies (extensively presented in sections 2.1, 2.2 and 2.3). It also describes their potential use within VICON along with potential contributions of VICON so as to fill existing standardization gaps.

Standard/Methodology	Potential use in VICON	Potential contribution of VICON
ETSI ES 202 746; Human Factors (HF); Personalization and User Profile Management; User Profile Preferences and Information	VICON may include the user preferences specified by the standard, especially those related to disabilities to the structure of the User Profiles.	
W3C Delivery Context Ontology	The Delivery Context Ontology itself constitutes a vocabulary of terms describing different types of devices. This vocabulary can be used in the User Models as well as in the User Profiles wherever special requirements have to be defined for a user concerning the interaction with various devices.	VICON could possibly extend the vocabulary of the Delivery Context Ontology in the context of assistive devices.
W3C Composite Capability/Preference Profiles (CC/PP)	VICON may use the user preferences specified by the standard, especially those related to disabilities, for the development of the User Models and User Profiles. The device characteristics could as well be used by VICON.	
ETSI EG 202 116; Human Factors (HF); Guidelines for ICT products and services; "Design for All"	ETSI EG 202 116 contains definitions of user characteristics, including sensory, physical and cognitive abilities. These definitions may be used in the development of the VICON User Models, in order to express how the disabilities are connected with the sensory, physical and cognitive abilities of the user. Additionally, ETSI EG 202 116 describes how user abilities are changing over years. This information could be used in the development of VICON Generic User Profiles representing users of different age	VICON could possibly extend the definitions of user characteristics and even create machine-readable formats (ex. XML-schemas) of these characteristics.

Standard/Methodology	Potential use in VICON	Potential contribution of VICON
	groups.	
BS EN 1332-4:2007 (Identification card systems. Man-machine interface. Coding of user requirements for people with special needs)	BS EN 1332-4:2007 provides a set of detailed definitions of user needs (such as preferred speech output rate, requirement for specific type of fonts, etc.), including people with special needs, for example the aged, minors, people with disabilities, those with learning difficulties, first time users, those not conversant with the local language. These user needs may be used in the VICON User Models, presenting this way how user needs are connected with the disabilities.	VICON could possibly extend the list of requirements concerning the interaction of elderly and disabled people with a user interface.
ISO 11228-2:2007 (Ergonomics -- Manual handling -- Part 2: Pushing and pulling)	ISO 11228-2:2007 provides structured information illustrating the maximum acceptable forces concerning pushing/pulling for the 90% of the healthy adult working population, according to different parameters (gender, pushing/pulling distance, frequency, etc.). Additionally, many other comparative tables are provided (ex. population subgroup profiles varying in age and gender and reflecting elderly working population (50-64 years)). This information could be taken into account during the development of the VICON Generic User Profiles.	
Foundational Model of Anatomy ontology (FMA)	FMA is concerned with the representation of classes or types and relationships necessary for the symbolic representation of the phenotypic structure of the human body in a form that is understandable to humans and is also navigable, parseable and interpretable by machine-based systems.	
ISO/IEC 19774 — Humanoid Animation (H-Anim)	This standard may be used in VICON for the creation of the User Profiles software to create and animate 3D human figures.	VICON may contribute to this standard by providing our experience out of the project.

Standard/Methodology	Potential use in VICON	Potential contribution of VICON
The Rapid Upper Limb Assessment (RULA)	RULA Analysis can be used in VICON to analyze many facets of a personas posture based on a combination of automatically detected variables and user data.	
SNOOK equations	The SNOOK equations can be used in VICON to help to optimize manual tasks for our target population and design better, and more widely accepted, products and workplaces	
ISO/DIS 24502 (Ergonomics -- Accessible design -- Specification of age-related relative luminance in visual signs and displays)	ISO/DIS 24502 provides a comparative analysis concerning the age-related spectral luminous efficiency (age is defined in decade). This information is pretty useful when there is need for the development of User Models representing a population group of specific age and could be taken into account during the development of the VICON Generic User Profiles.	
WHO ICF	WHO ICF provides classifications related to body functions and structure, and a list of domains of activity which may be used during the development of the VICON User Models.	
UsiXML	UsiXML provides meta-models that describe in detail the design of multimodal user interfaces, while it also supports task modeling. Consequently, UsiXML could be used in VICON for the development of multimodal user interfaces and task models.	
Hierarchical Task Analysis (HTA)	HTA provides a task analysis methodology, which will be used in VICON to describe the tasks of the Virtual User interacting with washing machines and mobile phones.	

Table 14 VICON in the context of existing standards

7 GUIDE Methodology

The GUIDE project aims to develop a toolbox of adaptive, multi-modal user interfaces that target the accessibility requirements of elderly and impaired users in their home environment, making use of TV set-top boxes as processing and connectivity platform. For this purpose, the toolbox not only provides the technology of advanced multi-modal user interface (UI) components, but also the adaptation mechanisms necessary to let the UI components interoperate with legacy and novel applications, including the capability to self-adapt to user needs.

The GUIDE user model simulates basic perceptual, cognitive and motor capabilities of users. The user model is a set of mathematical models which takes parameters on users' and devices' properties and capabilities. The models are calibrated, validated and used through a simulator, which is explained in the next section. The following table (Table 15) relates VUMS terms to GUIDE terms used in next section.

VUMS terms	GUIDE correspondence
User Model	Almost same though it also includes models of basic perception-cognition-motor action
User Profile	Same as VUMS
Virtual user	Has not used explicitly
Environmental Model	Almost Same as VUMS
Device Model	Same as VUMS
User Agent	Has not used explicitly yet
Interaction Model	Has not used explicitly yet
Context Model	Same as VUMS
Simulation	Same as VUMS
Validation	Same as VUMS
Adaptive User Interfaces	Same as VUMS
User Interface Design Pattern	Has not used explicitly yet

Table 15 Correspondence between GUIDE terms and VUMS terns

7.1 The simulator

Figure 3 below shows the architecture of the simulator. It consists of the following modules:

The Environment model contains a representation of the application and context of use. It consists of:

- **The Application model** containing a representation of interface layout and application states.
- **The Task model** representing the current task undertaken by a user that will be simulated by breaking it up into a set of simple atomic tasks following the KLM model [23]..
- **The Context model** representing the context of use like background noise, illumination and so on.

The Device model decides the type of input and output devices to be used by a particular user and sets parameters for an interface.

The User model simulates the interaction patterns of users for undertaking a task analysed by the task model under the configuration set by the interface model. It uses the sequence of phases defined by Model Human Processor [23].

- The perception model simulates the visual perception of interface objects. It is based on the theories of visual attention.
- The cognitive model determines an action to accomplish the current task. It is more detailed than the GOMS model [51] but not as complex as other cognitive architectures.
- The motor behaviour model predicts the completion time and possible interaction patterns for performing that action. It is based on statistical analysis of screen navigation paths of disabled users.

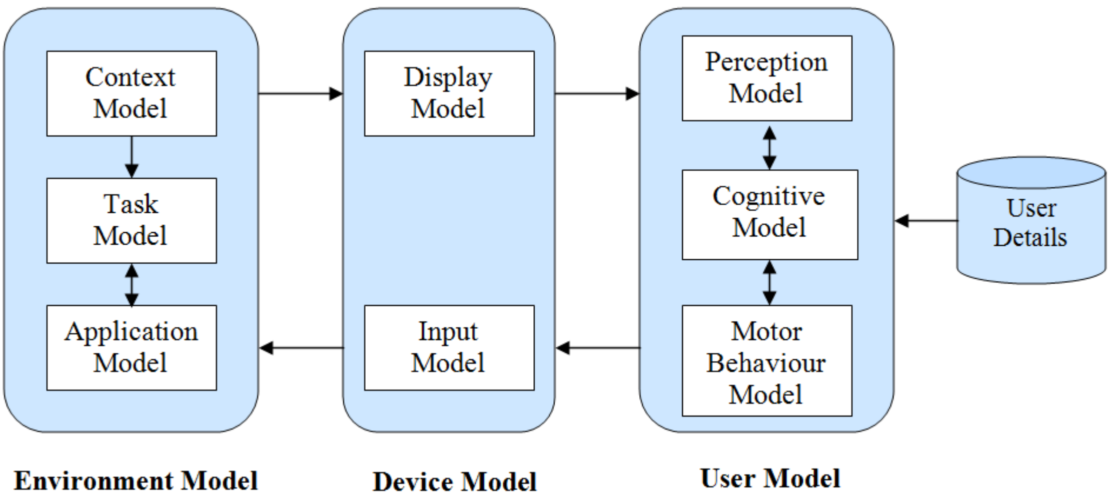
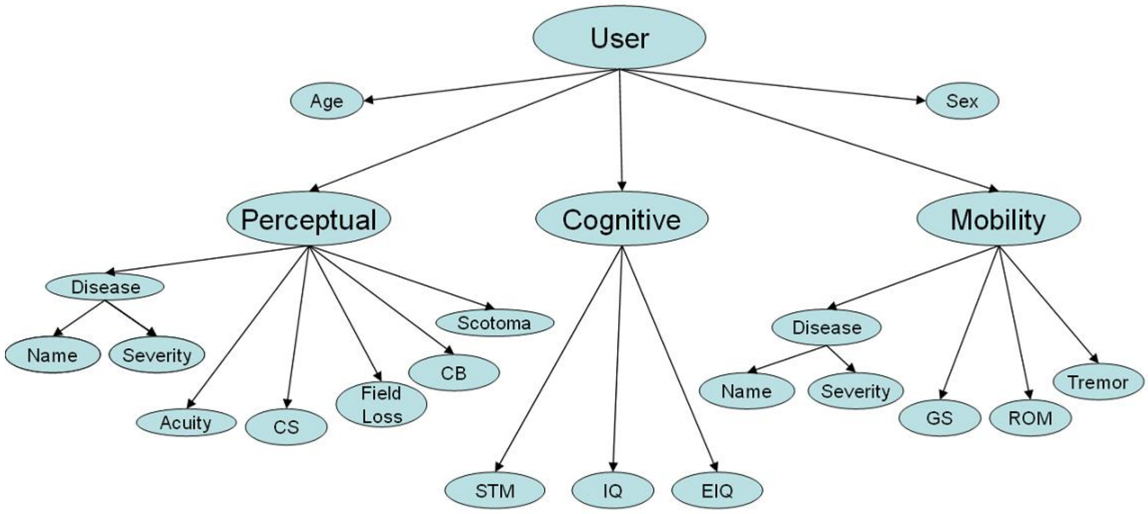


Figure 3. Architecture of the Simulator

The details about users are stored in xml format in the user profile following the ontology shown in Figure 4 below. The ontology stores demographic detail of users like age and sex and divide the functional abilities in perception, cognition and motor action. The perception, cognitive and motor behaviour models take input from the respective functional abilities of users.



STM: Short Term Memory, IQ: Intelligent Quotient, EIQ: Emotional Intelligent Quotient

Figure 4. User Ontology

The visual perception model [15] simulates the phenomenon of visual perception (like focussing and shifting attention). We have investigated eye gaze patterns (using a Tobii X120 eye tracker) of people with and without visual impairment. The model uses a backpropagation neural network to predict eye gaze fixation points and can also simulate the effects of different visual impairments (like Maccular Degeneration, colour blindness, Diabetic Retinopathy and so on) using image processing algorithms. Figure 5 shows the actual and predicted eye movement paths (green line for actual, black line for predicted) and points of eye gaze fixations (overlapping green circles) during a visual search task. The figure shows the prediction for a protanope (a type of colour blindness) participant and so the right hand figure is different from the left hand one as the effect of protanopia was simulated on the input image.

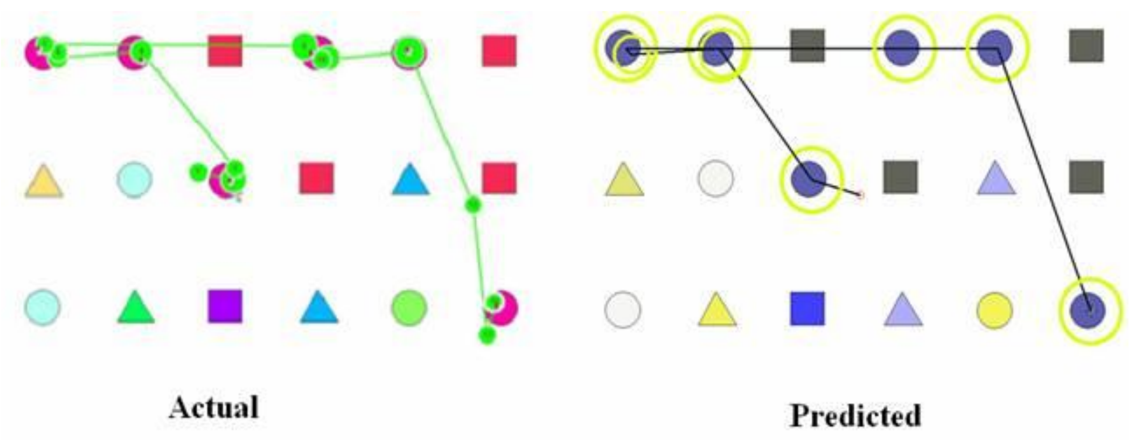


Figure 5. Eye movement trajectory for a user with colour blindness

The auditory perception model is under development. It will simulate effect of both conductive (outer ear problem) and sensorineural (inner ear problem) hearing impairment. The models will be developed using frequence smearing algorithm [73] and will be calibrated through audiogram tests.

The cognitive model [13] breaks up a high level task specification into a set of atomic tasks to be performed on the application in question. This operation is illustrated in Figure 6. At any stage, users have a fixed policy based on the current task in hand. The policy produces an action, which in turn is converted into a device operation (e.g. clicking on a button, selecting a menu item and so on). After application of the operation, the device moves to a new state. Users have to map this state to one of the states in the user space. Then they again decide a new action until the goal state is achieved.

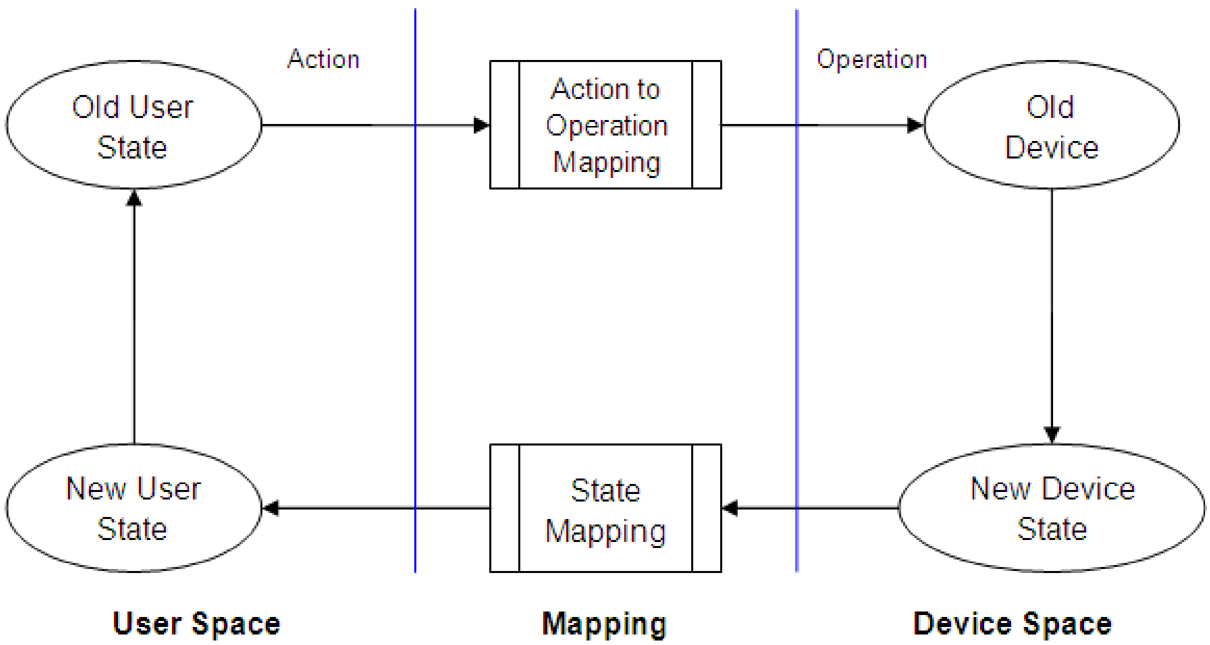


Figure 6. Sequence of events in an interaction

The model also has the ability to learn new techniques for interactions. Learning can occur either offline or online. The offline learning takes place when the user of the model (such as an interface designer) adds new states or operations to the user space. The model can also learn new states and operations itself. During execution, whenever the model cannot map the intended action of the user into an operation permissible by the device, it tries to learn a new operation. To do so, it first asks for instructions from outside. The interface designer is provided with the information about previous, current and future states and he can choose an operation on behalf of the model. If the model does not get any external instructions then it searches the state transition matrix of the device space and selects an operation according to the label matching principle [88]. If the label matching principle cannot return a prospective operation, it randomly selects an operation that can change the device state in a favourable way. It then adds this new operation to the user space and updates the state transition matrix of the user space accordingly. In the same way, the model can also learn a new device state.

Whenever it arrives in a device state unknown to the user space, it adds this new state to the user space. It then selects or learns an operation that can bring the device into a state desirable to the user. If it cannot reach a desirable state, it simply selects or learns an operation that can bring the device into a state known to the user.

The model can also simulate the practice effect of users. Initially the mapping between the user space and the device space remains uncertain. It means that the probabilities for each pair of state/action in the user space and state/operation in the device space are less than 1. After each successful completion of a task the model increases the probabilities of those mappings that lead to the successful completion of the task and after sufficient practice the probability values of certain mappings reach one. At this stage the user can map his space unambiguously to the device space and, thus, behave optimally. We are presently working on simulating effect of a few cognitive impairments like dyslexia, dementia and so on.

The motor behaviour model [16] is developed by statistical analysis of cursor traces from motor impaired users. We have evaluated hand strength (in terms of grip strength, range of motion of wrist and forearm and static tremor using a Baseline 7-pc Hand Evaluation Kit and Comby® Stressometer) of able-bodied and motor impaired people and investigated how hand strength affects human computer interaction. Based on the analysis, we have developed a regression model to predict pointing time. Application of Fitts' Law for motor impaired users is a debatable issues [16], our model is an alternative to Fitts' Law though it conforms to the basic prediction of Fitts' law in terms of relationship between movement time and distance and width of target. Figure 7 shows an example of the output of the model. The thin purple line shows a sample trajectory of mouse movement of a motor impaired user. It can be seen that the trajectory contains random movements near the source and the target. The thick red and black lines encircle the contour of these random movements. The area under the contour has a high probability of missed clicks as the movement is random there and thus lacks control.

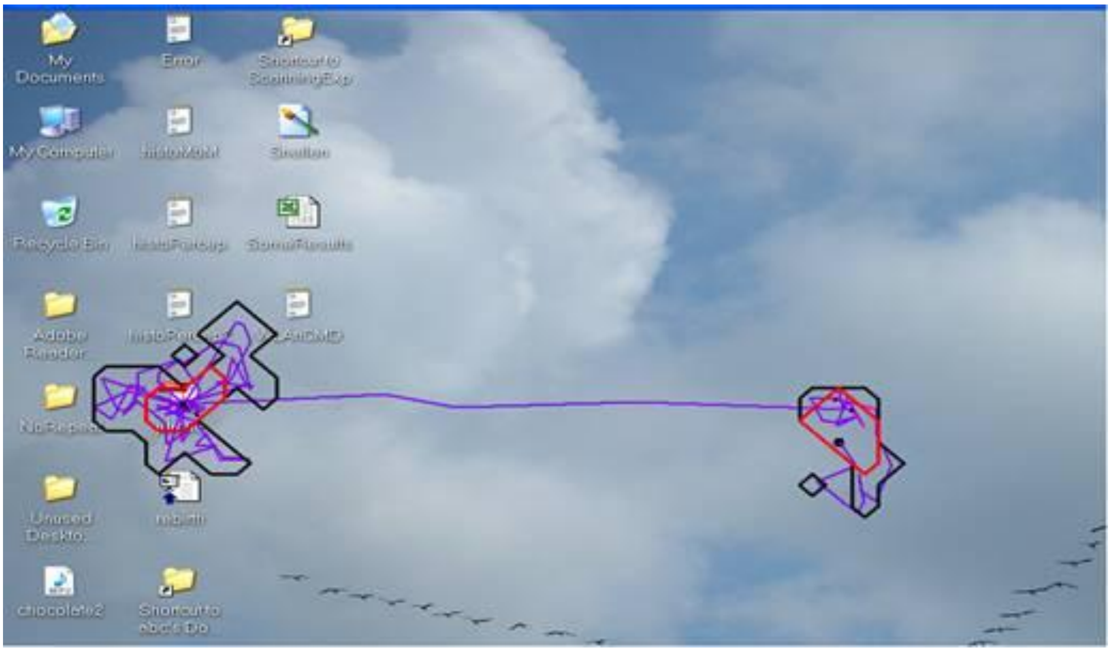
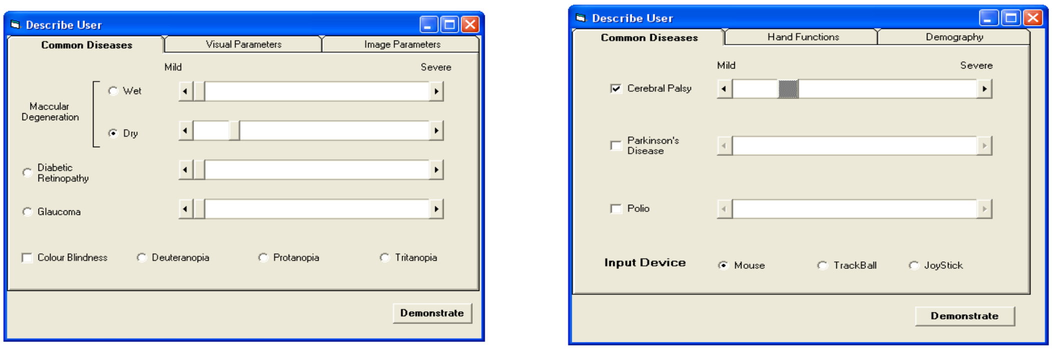
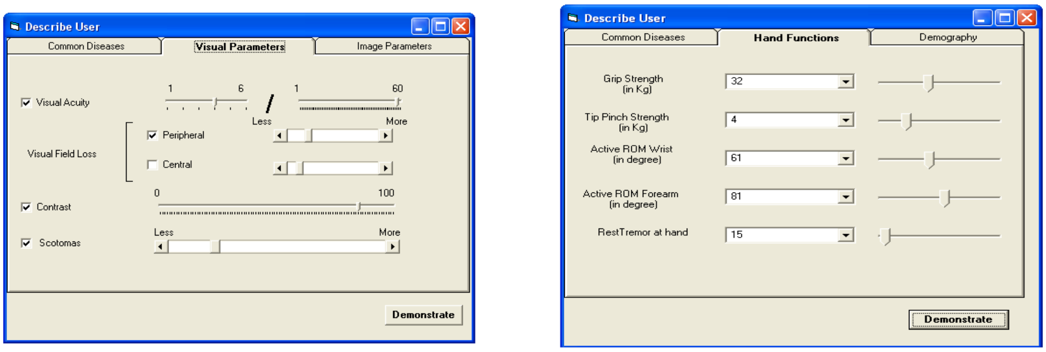


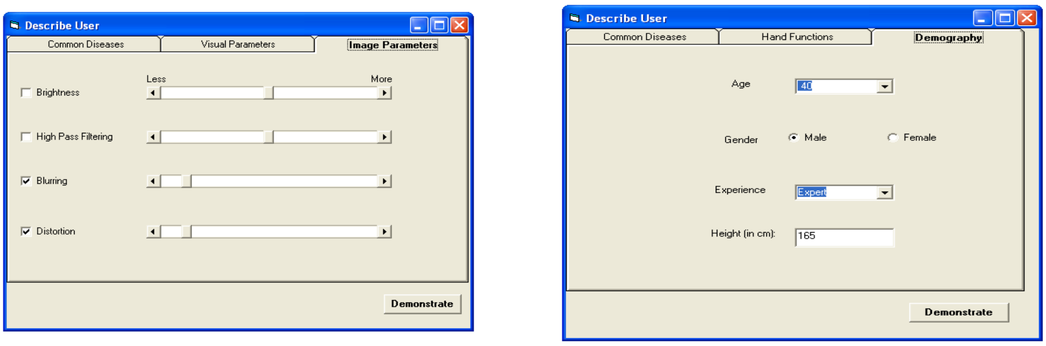
Figure 7. Mouse movement trajectory for a user with cerebral palsy



a. Interfaces to simulate the effects of different diseases



b. Interfaces to simulate the effects of different visual functions and hand strength metrics



c. Interfaces to run image processing algorithms and set demographic detail of users

Figure 8. A few interfaces of a prototype of the toolbox

These models do not need detailed knowledge of psychology or programming to operate. They have graphical user interfaces to provide input parameters and to show the output of the simulation. Figure 8 shows some screenshots of the simulator.

At present it supports a few types of visual and mobility impairments. For both visual and mobility impairments, we have developed the user interfaces in three different levels:

- In the first level (figure 8a) the system simulates different diseases. With increasing level of the severity of the diseases the parameters of the models change accordingly, for example with increasing level of severity of cerebral palsy or polio the grip strength and Range of Motion of wrist and forearm decreases exponentially.
- In the next level (figure 8b) the system simulates the effect of change in different visual functions (like Visual acuity, Contrast sensitivity, Visual field loss and so on), hand strength metrics (like Grip Strength, Range of Motion of forearm, wrist and so on) and auditory parameters (like audiogram, loudness and so on).
- In the third level (figure 8c), the system allows different image processing and digital filtering algorithms to be run (such as high/low/band pass filtering, blurring etc.) on input images and demographic details of the users to be set.

The simulator can show the effects of a particular disease on visual functions and hand strength metrics and in turn their effect on interaction. For example, it can demonstrate how the progress of dry macular degeneration increases the number and sizes of scotoma (dark spots in eyes) and converts a slight peripheral visual field loss into total central vision loss. Similarly, it can show the perception of an elderly colourblind user, or in other words the combined effect of visual acuity loss and colour blindness. We have modelled the effects of age and gender on hand strength and the system can show the effects of Cerebral Palsy or Parkinson's disease for different age group and gender. Some sample screenshots can be found at http://www.cl.cam.ac.uk/~pb400/Demo.htm.

Validation of the models

Each of the perception, cognitive and motor behaviour models were calibrated and validated separately involving people with and without visual and mobility impairments [14].

The visual perception model was validated through an eye gaze tracking study for a visual search task. We compared the correlation between actual and predicted visual search time, eye gaze and also investigated the error in prediction. The actual and predicted visual search time correleated statistically significantly with less than 40% error rate for more than half of the trials [15].

The cognitive model was used to simulate interaction for first time users and to simulate the effect of learning as well [13].

The motor behaviour model was validated through ISO 9241 pointing task. The actual and predicted movement time correleated statistically significantly with less than 40% error rate for more than half of the trials [16].

The whole simulator has been validated [14] for an icon searching task involving people with and without visual and mobility impairments.

7.2 Run time system

We will use the simulator to develop the GUIDE profile that will help to dynamically adapt interfaces based on the user, context and particular application in use. It works in the following way (Figure 9): The user can provide input through multiple devices like motion sensors (like Wiimote) and speech recognizers, meaning that he can use multiple modalities like pointing, gesture and speech simultaneously. The signals from recognition based modalities are processed by interpreter modules like a series of points from the motion sensor go through a gesture recognition engine, in order to detect gestures. Signals corresponding to pointing modalities go through input adaptation modules (e.g. in order to smooth tremors from the user's hand or to guess the intention of the user). Both interpreter and adaptation modules base their decisions on knowledge stored in the GUIDE profiles achieving noise reduction in the input signals. The multimodal fusion module analyzes the raw input signals and the outputs of input interpreters and input adaptation and combines these multiple streams into a single interpretation based on the user, context and application models. The interpretation resulting from the input signals is sent to the dialog manager who decides which will be the application's response. This response is fed to the multimodal fission module, which again takes input from the user, context and application models and prepares the output appropriately (like embedding a HTML page in a video with subtitle and voice output) to be rendered in the output devices. The user perceives this output and provides further input.

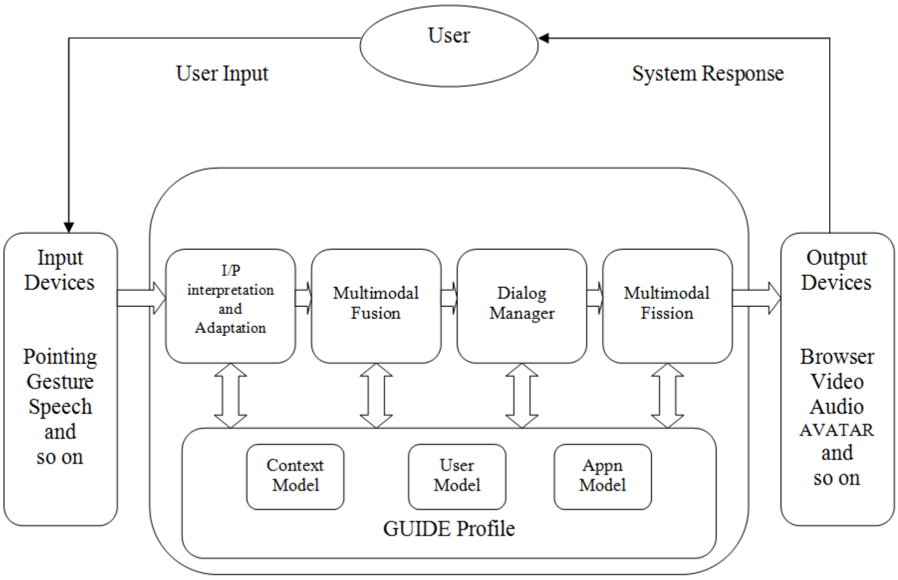


Figure 9. Run time adaptation system

7.3 GUIDE requirements from existing standard

This section highlights the requirements of GUIDE from existing standards

1. **Description of functional abilities of users:** We need a standard hierarchy to describe basic perceptual, cognitive and motor ability of users.

2. **Storing details of user, interfaces and devices:** We have to store detail of users, interfaces and devices in an interoperable machine readable form, and a standard for each of these details can help in seamless integration of our models to different platforms and devices.

3. **Calibration and Validation of models:** GUIDE requires existing standards on running user trials to calibrate and validate the models.

4. **Accessibility issues:** GUIDE will try to follow the existing accessibility guidelines and recommendations in designing application interfaces.

5. **Application development:** During application development, GUIDE will follow standards regarding hybrid TV and digital TV in general.

7.4 GUIDE in context of VUMS

We have adapted the GUIDE user model according to VUMS definition. As a starting point, we have added all the VUMS terms in GUIDE glossary of terms. Though the concept of initial GUIDE user model was different than VUMS definition (as it contains a dynamic behavioural simulation module), we have adapted the XML structure for the storage of user details using tags used by othe VUMS projects. As a result, we can import and export our user profiles among other projects. We are also trying to follow a similar approach for the other models, like Application model and Device Model. Additionally, we are defining a standard for conducting user trials to calibrate and validate the models as well as storing data, so that projects can share user trial data among themselves.

8 GUIDE in the context of existing standards

The following table summarizes relevant standards and their relevance to the GUIDE user model.

Standards	Potential use in GUIDE		Potential contribution of GUIDE
	Part of GUIDE User Model	Relevance	
ACT-R/PM, EPIC	Architecture	The architecture, calibration and validation of the simulator follow similar principles such as those used in the cognitive architectures (mainly in ACT-R/PM and EPIC systems).	GUIDE user model could contribute to develp a new inclusive cognitive architecture standard.
GOMS models	Task model	The task model uses the task decomposition technique used by the KLM GOMS model.	
XForm/UIML/XIML/UsiXML	Interface Model	We shall use one of XForm/UIML/XIML/UsiXML to describe interfaces.	
ISO IEC 24752 – Part 4		This standard can be used to describe interfaces.	
ETSI TS 102 747	User Model	This standard can be used to develop the user profile.	
ISO IEC 24751		This standard can be used to model cognition and cognitive impairments.	
WHO ICF		The classification technique will be used for requirement analysis and user profile generation.	
XML		We have used XML to store user characteristics.	
ISO 9241 Ergonomics standard	Calibration	It is used to conduct user studies for calibrating and validating the simulator and also to evaluate new interaction devices.	
WCAG and other Guidelines	Validation	Initially, the output of the simulation will be com-	GUIDE user model could

Standards	Potential use in GUIDE		Potential contribution of GUIDE
	Part of GUIDE User Model	Relevance	
		pared to the existing guidelines to validate the results and in a next step the simulation output will augment the existing guidelines in the form of the GUIDE handbook.	develop new guidelines in the form of the GUIDE Handbook.

Table 16 GUIDE in the context of existing standards

9 MyUI methodology

9.1 MyUI terms and their correspondence with the VUMS terms

Relations between the VUMS terms, as presented in the common VUMS glossary in section 2, and the terms used to further detail the MyUI methodology in this document are outlined in the following table (Table 17).

VUMS terms	Corresponding terms in MyUI
User Model	User Model (as defined in the VUMS glossary)
User Profile	User Profile (as defined in the VUMS glossary)
Virtual user	No separate term Not relevant in MyUI as simulation plays a minor role since the 2nd amendment of the MyUI DoW from 23 May 2011
Environmental Model	Environment Model (as defined in the VUMS glossary)
Device Model	Device Model (as defined in the VUMS glossary)
User Agent	Browser (focus on web-based user agents)
Interaction Model	MyUI Abstract Interaction Model
Context Model	Context Model, encompassing User Model and Environment Model
Simulation	Simulation In MyUI simulation is limited to the WYSIWOG representation for customized user interface aspects, e.g. how will a colour-blind person see in a specific colour scheme.
Validation	No separate term As MyUI is based on user profiles which

	are built and refined during run-time, user profile validation is an ongoing process. User models are not directly validated in MyUI.
Adaptive User Interfaces	Adaptive User Interfaces (as defined in the VUMS glossary)
User Interface Design Pattern	User Interface Design Pattern (as defined in the VUMS glossary)

Table 17 Correspondence between MyUI terms and VUMS terns

9.2 MyUI user, environment and device modelling methodology outline

The MyUI user, environment and device model is described by an ontology including all information on sensor, device, location and user characteristics as well as their relationships. In contrast to the user model approaches used in VERITAS and VICON, MyUI aims to focus on a function-based modelling approach rather than on a diagnosis-based approach. This means that the underlying ontology in MyUI is based on interaction constraints that individual users might have and not on detailed medical impairments and limitations of the individual user.

Individual user profiles, derived from the general user model, form the basis for the individualised user interface generation by composing those design patterns that are most-fitting to the current user profile characteristics (Figure 10). User profiles are continuously updated during runtime based on information gained from different software and hardware sensors as well as by the user's interaction with the system. Sensor information is processed by means of aggregation, augmentation and conflict resolution. Calculated changes in the user profile are made as needed. In this way, adaptive user interfaces at runtime are provided in MyUI.

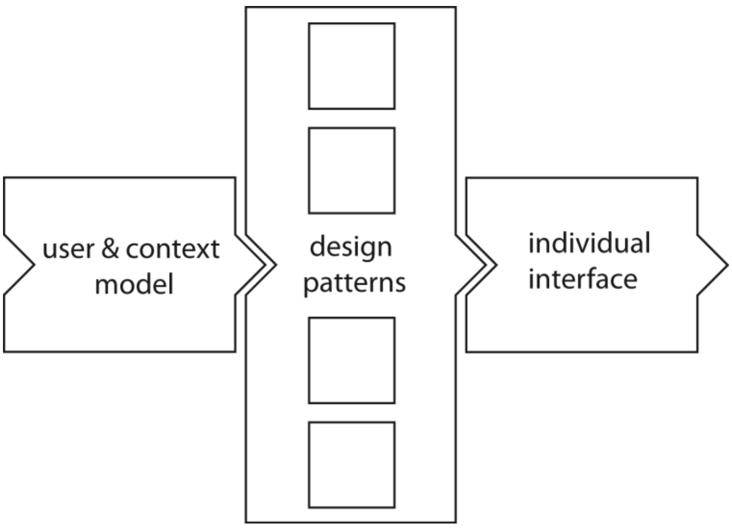


Figure 10: User and context model as basis for the individual user interface generation

9.2.1 MyUI user, device and environment model - general approach

MyUI aims at providing support for adaptive user interfaces that dynamically adapt to the needs and the characteristics of the user and the user's environment i.e., the **context** of a user. More specifically, context can be defined as

any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.(taken from [28])

The role of context management in adaptive systems is better understood by referring to the general model of adaptive system described in D2.1. The model describes the three typical stages involved when providing adaptive behavior:

1. Afference – collection of observational data about the user.
2. Inference – creating or updating a user model based on that data.
3. Efference – deciding how to adapt the system behavior.

Typically, context management integrates the stages of afference and inference with the goal of providing a useful representation of contextual information that enables efference. Seen from the viewpoint of context management, we can characterize afference as providing a shared sensor data repository where sensor events are collected in terms of a sensor or device model. On top of this, inference is the integration and management of different context augmentation services. This augmentation services may incorporate environmental information about, for example, location properties of the user. The goal of this augmentation is to gather information about the user. This information must be useful to enable adaptive behavior in the phase of afference. Information about the user is structured according to the user model and stored in the user profile. While the user model describes the information collected about users in terms of a schema, the user profile is the instantiation of the user model for a concrete user. Both, the user model and the user profile, will be discussed in the course of the subsequent sections. The next three subsections will also further examine the role of context management in the light of this 3 stage model.

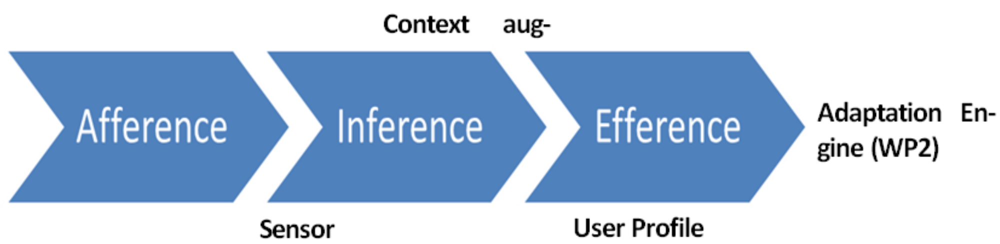


Figure 11: Approach to an adaptive system in the light of context management

9.2.1.1 Collection of Sensor Information

Looking at the problem of afference in more detail, we can distinguish implicit afference, i.e. collecting data in the background without direct involvement of the user, and explicit afference, i.e. actively involving the user in data collection. Since unobtrusive collection of user-related data without influencing the user in his or her normal interaction with the system cannot be discarded as a requirement in MyUI, the architecture of the context management solution has to take into account both cases.

Implicit afference can be achieved by using sensors to unobtrusively detect observational data about the user. Unfortunately, individual sensors seldom deliver information that can be used directly to make useful statements about the user. Usually, sensor events must be connected, aggregated and augmented to arrive at information that can be used to populate a user profile.

This leads us to a generic context management architecture for adaptive systems where sensor events are collected in a sensor data repository. This shared data repository is filled by virtual and physical sensors that store their sensor events into the data repository. Physical sensors are pieces of hardware equipment placed in the surrounding of the user that are able to detect certain characteristics that are useful to infer user profile information. On the other hand, virtual sensors are pure software based sensors that detect information by analyzing the user’s interaction with the MyUI system.

In the inference stage context augmentation algorithms are then accessing this shared sensor data repository via a blackboard-like API and they are trying to derive user profile information from sensor events, additional background knowledge and historic context information.

9.2.1.2 Context Augmentation

As described in [61] rule-based approaches, probalistic reasoning, data mining, predictive reasoning, and other machine-learning approaches might be used to derive useful information about the user. Each of these different approaches has different drawbacks and benefits, e.g. probabilistic methods might have benefits in activity recognition, whereas rule-based approaches are easier to maintain and implement. To account for different possibilities the general context management architecture should be able to incorporate different reasoning mechanisms.

This can be achieved by a data-centric integration architecture. As analyzed by Winogard in [99], blackboards are the architecture of choice for this problem. In such blackboard architecture, sensor events and other useful information are published on a shared blackboard. Context augmentation services that represent different reasoning and inference techniques and algorithms can then access the information on the blackboard and derive useful user profile information.

The blackboard itself provides an event-based API to a data repository. The data repository is specified by a data model that incorporates the sensor or device model and additional environmental information that might be useful for context augmentation, like additional information about the location of the user.

Compared to other approaches such architecture focuses on a separation of the shared descriptive data that is captured in sensor data repository and user profile from the algorithmic and processual knowledge that is needed to infer additional information. Consequently, context augmentation services and therefore reasoning algorithms can be replaced independently. This enables fast prototyping, by easily experimenting with different reasoning approaches.

9.2.1.3 User Model

Capabilities and limitations of humans change over time, i.e. during the process of aging. As described in the MyUI-deliverable D2.1 (section 5.2.2.1) one example is the ability of hearing, which decreases over time. Although it is common that abilities decrease during aging, there is no uniform pattern of the steady cognitive decline. Sometimes it is also possible that a loss of cognitive abilities can be reversed. This induces that the user profile is never fixed at a certain point in time. Rather the user profile has to be continuously adapted to the user it is associated to.

In MyUI, context management enables adaptive user interfaces by providing a continuously updated user profile. Of course, to enable efference the user model must provide information that is able to trigger and drive adaptation. Since MyUI is dealing with the adaptation of user interfaces, MyUI's user model is focusing on impairments and disabilities with regards to human computer interaction.

In general, a MyUI user profile is a collection of information about an end-user of the MyUI system. This includes personal information like email address, first name, last name, etc. as well as information about user capabilities and characteristics as far as they are relevant to determine the human computer interaction (HCI) abilities of a user. The general idea is to collect and continuously adapt HCI-relevant user information, so that human computer interfaces (also called user interfaces) can be dynamically adapted to the current capabilities, needs and limitations of a user.

In more technical terms, the user profile is based on the RDF data model. RDF describes relationships in terms of resources. Resources can be differentiated into subjects, predicates, objects and statements. A statement is a triple consisting of subject, predicate and object. An RDF-statement roughly translates to a data representation of a simple natural language sentence that captures statements about the subjects. In RDF resources are referenced by Uniform Resource Identifiers URIs. In relation to MyUI a statement always addresses a user capability, characteristic or short property of a user. The user is the subject. The property is the predicate. The value that is stated for this property is the object of the statement.

When it comes to adaptation, we are using the term "user profile variable". This corresponds to a statement where the subject is the specific user this variable belongs to. The property, which can be seen as the predicate in a RDF-statement, is also called "user profile variable name", and the object is called "user profile variable value". In short, one could say that the property must be the name of a user profile variable.

The range of a property defines which values can be used as an object in a statement (i.e. it defines the type of the corresponding user profile variable). We can distinguish the following three cases:

- *Numerical Literal:* We use numerical literals to represent user profile variables which are ratio-scale. In this case, one value is selected from a predefined, continuous interval for a user profile variable of this type. In the current ontology this interval is considered to be the same for all ratio-scaled user profile variables (in contrast to nominal-scaled user profile variables where a different set of possible values is defined for each user profile variable). The interval is specified to be [0, 4] (i.e. all possible values between 0 and 4 including 0 and 4 can be assigned). Although, we want to avoid an exact medical definition, 0 can roughly be mapped to “not limited or normal”, 2 can roughly be mapped to “mildly limited” and 4 can roughly be mapped to “severely limited”. This kind of type is typically assigned to user profile variables that represent limitations and capabilities of the user like hand precision, visual acuity. Assigning a value of 0 to hand-precision, therefore, maps to the statement that this user’s hand-precision is not limited or normal. The default value is 0. However the above mapping exists, this doesn’t have to be applicable for all user profile variables. The user profile variable “ambient light”, for example, uses another mapping (0 translates to no ambient light meaning absolute darkness, 4 translates to very high ambient light). Concluding it can be said that the meaning of the values of a numerical literal depends on the concrete user profile variable. Using a continuous, ratio-scaled interval has the advantage that there exists a strong order between the distinct values for a user profile variable. Furthermore using continuous values is helpful by providing the possibility to have a fine-grained differentiation.

- *String Literal:* This corresponds to a free-text user profile variable. This means that any value can be assigned to this kind of user profile variable as long as it is a string. This user profile variable type is used for user characteristics like first name, last name and email address. Although, in principle any value is accepted (as opposed to a set or interval of values), the format of the string might be constrained in a clearly defined way. For example, the email address must be a syntactically correct email-address; or the first name must not exceed a certain number of letters. The default value for a user profile variable of this kind is an empty string.
- *Enumeration:* This means a subset from a set of predefined values (represented as concept instances) can be assigned to a user profile variable which is nominal-scaled. This user profile variable type is used to specify the languages that are accepted by a person. In a collection of statements, an “assigned subset” maps to a set of statements with the corresponding property for that user profile variable. The collection must not be empty if it is explicitly assigned. Enumerations also support default values, which are used when there is no corresponding statement. The default value is one specifically marked value of the set of possible values (since only subsets can be assigned, technically it would be a set containing only this single element).

9.2.2 Properties and User Profile Variables

In MyUI, a function-based user modelling approach is used. Hence, user profile variables are connected to specific user interaction abilities and constraints subdivided into perceptual, cognitive and motor attributes. Variables relevant for MyUI have been initially selected from the WHO ICF guidelines. Further sources, e.g. the ISO 22411 standard and major requirements determined in the MyUI-deliverable D2.1 (Requirements for User Interface Adaptation), are also considered.

The following key questions have been underlain the selection process:

1. Does a certain attribute affect the interaction with ICT products?
2. Can user interface adaptation overcome or weaken the interaction constraints?

If both questions can be answered with “yes” the attribute has been included as additional user profile variable to the initial MyUI user model. Based on appropriate feedback from the end-user studies, further refinements on user model variables might be done in the future.

9.2.3 The User Profile as basis for the individual user interface generation

Generic design patterns refer to certain user profile variables in order to set global user interface features as font size. In this way, an individual user interface profile is created. The individual user interface is composed by different user interface elements. Each user interface element is described by a related interaction design pattern which in turn is based on specific global variables from the user interface profile and the current interaction situation.

In general, design patterns (generic and interaction) are selected if preconditions for a certain combination of user profile characteristics and user interface profile characteristics are fulfilled. The preconditions are illustrated in the following on the example of generic design patterns.

The majority of the user profile variables are ratio-scaled, some are nominal-scaled. For these two data types, examples are presented for simple preconditions:

- *Ratio-scaled user profile variables:* A simple precondition of a generic design pattern could define an interval restricting allowed values for a certain ratio-scaled user profile variable. If a user’s individual assignment of this ratio-scaled user profile variable is included in the given interval, then the precondition is met and the generic design pattern is selected. For example, considering a pattern suitable for people having normal or slightly limited hand precision. So, the precondition is “IF 0<= hand_precision <=2”. In consequence, for all users that have a value less-than or equal-to 2 assigned to this user profile variable this generic design pattern is applied.
- *Nominal-scaled user profile variables:* here a simple precondition might check the presence or absence of a specific user profile variable. If a user’s individual assignment of this user profile variable equals the hypothesis, the generic design pattern is selected.

Concerning the interaction design patterns, similar preconditions are used. The only difference is that they refer to global user interface profile variables and a specific interaction situation.

9.3 MyUI requirements from existing standards

This section lists important abstract requirements derived from the previous analysis of existing standards and methodologies. Concerning MyUI, this mainly addresses the area of user modeling. Therefore, the analysis also focuses on user-modeling specific requirements.

- **Modelling of functional limitations and disabilities:** Functional abilities and limitations, separated into perceptual, cognitive and motor categories, are described in the MyUI user model. In this way, the MyUI user model allows to establish individual user profiles over time providing the base for individual user interface adaptations according to the user's needs.

- **Accessibility support:** Developers and designers are supported by a development toolkit providing access to the patterns repository and helping to develop their own self-adaptive application to offer accessibility to a larger number of people.

- **User profile description:** MyUI user profiles are detailed and structured as described above. Personalization is supported by distinguishing perceptual, cognitive and motor functional abilities and limitations.

- **User Interface Modelling:** The user interface modeling approach in MyUI is based on the individual user characteristics, stored in the MyUI user profiles. Specific design patterns are selected accordingly and composed to individualized, self-adapting user interfaces.

- **Assistive devices:** Input devices are selected to meet the requirements of MyUI end users, i.e. people having limitations that are usually assigned to older people and stroke survivors.

- **Machine-readable format:** MyUI user profile and user model capture well-defined knowledge in a machine-readable format that can be automatically processed and interpreted by the MyUI-system.

10 MyUI in the context of existing standards

The pragmatic user modelling approach in MyUI is based on modelling functional limitations of the user. Therefore our user model is based on the ICF list of functional constraints. Other user modelling standards that are capturing static preferences of the user are not suitable for triggering adaptive system behaviour. More precise modelling standards that capture very specific medical knowledge about the user are very hard to detect in real-time and they typically result in user profiles that contain complex medical knowledge. Application developers usually have difficulties to assess this kind of knowledge. Our model is based on an easy to understand severity scale which makes interpretation of the information easier. Considering all this, standard user modelling approaches are not suitable in the MyUI context.

However, single aspects of the following standards seem to have the potential to be integrated in MyUI. Table 18 also includes potential contributions of MyUI to some existing standards.

Standard/Methodology	Potential use in MyUI	Potential contribution of MyUI
ETSI TS 102 747; Human Factors (HF); Personalization and User Profile Management; Architectural Framework		MyUI's user profile management framework for dynamic detection of functional impairments related to use of ICT could be considered as an extension for highly dynamic scenarios.
ETSI ES 202 746; Human Factors (HF); Personalization and User Profile Management; User Profile Preferences and Information	MyUI user profile manager focuses on the dynamic detection of functional disabilities. The preference-centred approach of this standard can therefore not be incorporated easily. However, many of the profile items listed in the Annex "Preferences related to disabilities" might be included in MyUI's generic design patterns.	
W3C Delivery Context Ontology	The Delivery Context Ontology itself constitutes a vocabulary of terms describing different types of devices. MyUI's model can incorporate those descriptions when more devices are incorporated.	

Standard/Methodology	Potential use in MyUI	Potential contribution of MyUI
WHO ICF	WHO ICF's classifications of body functions and structures have been used as the main source of MyUI's modelling of functional impairments in relation to ICT.	

Table 18 MyUI in the context of existing standards

Conclusions

The main aim of this document was to provide an analysis of the existing standards, techniques and methodologies, in order to drive the work towards user modelling and standardisation of the ***"Virtual User Modelling and Simulation"*** cluster of EC co-funded projects VERITAS, GUIDE, VICON and MyUI.

More specifically, in the present document, an overview of the most relevant to User Modelling standards and a comparative analysis between them were presented. An overview and a comparative analysis of the most widely known User Interface Description languages and Task Modelling techniques were also presented. Furthermore, existing methods for physical and cognitive user modelling were investigated. The Modelling Methodologies followed by VERITAS/VICON/GUIDE/MyUI were analyzed and the requirements of each project from the existing standards were examined. Moreover, the document presented the relevance of the VERITAS/VICON/GUIDE/MyUI User Modelling approach with the existing standards, methodologies and techniques related to User Modelling, while it also analyzed areas, where each project could potentially contribute. The document concluded with an action plan of the VUMS cluster towards standardization.

References

1. Abrams, M., Phanouriou, C., Batongbacal, A.L., Williams, S. & Shuster, J. (1999), UIML: An Appliance-Independent XML User Interface Language. In A. Mendelzon, editor, Proceedings of 8th International World-Wide Web Conference WWW'8 (Toronto, May 11-14, 1999), Amsterdam, 1999. Elsevier Science Publishers.

2. Ambrósio, J. and Dias, J. A Road Vehicle Multibody Model for Crash Simulation based on the Plastic Hinges Approach to Structural Deformations, International Journal of Crashworthiness, 2007, 12(1), 77-92.

3. Anderson J. R. and Lebiere C. The Atomic Components of Thought. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1998.

4. Andriacchi, I. and Hurwitz D. Gait Biomechanics and the Evolution of Total Joint Replacement, Gait and Posture, 1997, 5, 256-264.

5. Annett, J., Cunningham, D., & Mathias-Jones, P. (2000). A method for measuring team skills. *Ergonomics, 43*, 1076–1094.

6. Annett, J., & Duncan, K. (1967). Task analysis and training design. *Occupational Psychology, 41*, 211–227.

7. Arsanjani, A., Chamberlain, D. and et al. (2002), (WSXL) web service experience language version, 2002. Retrieved from: http://www-106.ibm.com/developerworks/library/ws-wsxl2/.

8. Barnard P. The Emotion Research Group Website, MRC Cognition and Brain Sciences Unit. Available at: http://www.mrc-cbu.cam.ac.uk/~philb , Accessed on 1st July, 2007

9. Barthet, M.-F., & Tarby, J.-C. (1996). The Diane+ method. In J. Vanderdonckt, (Ed.), Computer-aided design of user interfaces (pp. 95–120). Namur, Belgium: Presses Universitaires de Namur.

10. Baumeister, L. K., John, B. E., & Byrne, M. D. (2000). A comparison of tools for building GOMS models: Tools for design. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI 2000)* (Vol. 1; pp. 502–509). New York: ACM Press.

11. Beard, D. V., Smith, D. K., & Denelsbeck, K. M. (1996). QGOMS: A direct-manipulation tool for simple GOMS models. In Proceedings of ACM Conference on Human Factors in Computing Systems (CHI '96) (Vol. 2; pp. 25–26). New York: ACM Press.

12. Benyon D. and Murray D. Applying User Modeling to Human Computer Interaction Design. Artificial Intelligence Review 7.3 (1993): 199-225.

13. Biswas P. and Robinson P. Automatic Evaluation of Assistive Interfaces. ACM International Conference on Intelligent User Interfaces (IUI) 2008, pp. 247-256.

14. Biswas P. and Robinson P. Evaluating the design of inclusive interfaces by simulation, Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI) 2010, pp. 277-280

15. Biswas P. and Robinson P. Modelling Perception using Image Processing Algorithms. 23rd British Computer Society Conference on Human-Computer Interaction (HCI 09).

16. Biswas P. and Robinson P. Predicting Pointing Time from Hand Strength. USAB 2009, LNCS 5889, pp. 428–447.

17. Blandford A., Butterworthb R. and Curzonb P. "Models of interactive systems: a case study on programmable user modelling." International Journal of Human-Computer Studies 60 (2004): 149-200.

18. Boden M. A. Computer Models of Mind: Computational Approaches in Theoretical Psychology. Cambridge, UK: Cambridge University Press, 1985.

19. Bovair S., Kieras D. E., and Polson P. G. The acquisition and performance of text-editing skill: A cognitive complexity analysis. Human-Computer Interaction 5 (1990): 1-48.

20. Butterworth R. and Blandford A. "Programmable user models: The story so far." Available at: http://www.cs.mdx.ac.uk/puma/wp8.pdf, Accessed on 30th June, 2007

21. Byrne M. D. ACT-R/PM And Menu Selection: Applying A Cognitive Architecture To HCI. International Journal of Human Computer Studies 55 (2001): 41-84.

22. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers, Vol. 15, No. 3 (June 2003) 289–308.

23. Card S., Moran T. and Newell A. The Psychology of Human-Computer Interaction. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1983.

24. Carroll J. M. and Olson J.M. Mental Models In Human-Computer Interaction. Handbook of Human-Computer Interaction Ed. Helander M. Amsterdam, Netherlands: Elsevier Ltd., 1990. 135-158.

25. Cognitive Architectures. Available at: http://en.wikipedia.org/wiki/Cognitive_ architecture, Accessed on 1st July, 2007

26. Crowle, S., Hole, L., ISML: An Interface Specification Meta-Language. DSV-IS 2003, Funchal, Madeira Island, Portugal. June 11-13, 2003. Lecture Notes I Computer Science 2844 Springer 2003, ISBN 3-540-20159-9.

27. Demler, G., Wasmund, M., Grassel, G., Spriestersbach, A. & Ziegert, T. (2003), Flexible pagination and layouting for device independent authoring, WWW2003 Emerging Applications for Wireless and Mobile access Workshop.

28. Dey, A., Abowd, G.; Toward a better understanding of context and context-awareness; In HUC ’99: Proceedings of the 1st international symposium on Hand-held and Ubiquitous Computing

29. Eisenstein J., Vanderdonckt J., Puerta A. (2000), Adapting to Mobile Contexts with User-Interface Modeling, Proceedings of 3rd IEEE Workshop on Mobile Computing Systems and Applications WMCSA’2000 (Monterey, 7-8 December 2000), IEEE Press, Los Alamitos, 2000, pp. 83-92.

30. Eisenstein J., Vanderdonckt J., Puerta A. (2001), Model-Based User-Interface Development Techniques for Mobile Computing, Proceedings of 5th ACM Int. Conf. on Intelligent User Interfaces IUI’2001 (Santa Fe, 14-17 January 2001), Lester, J. (Ed.), ACM Press, New York, 2001, pp. 69-76.

31. Eng K., Lewis R. L., Tollinger I., Chu A., Howes A. and Vera A. Generating Automated Predictions of Behavior Strategically Adapted To Specific Performance Objectives. ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI) 2006. 621-630.

32. Fabio Paterno' , Carmen Santoro , Lucio Davide Spano, MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments, ACM Transactions on Computer-Human Interaction (TOCHI), v.16 n.4, p.1-30, November 2009

33. Fitts P.M. The Information Capacity of The Human Motor System In Controlling The Amplitude of Movement. Journal of Experimental Psychology 47 (1954): 381-391.

34. Flores, P., Ambrósio, J., Claro, J.C.P. and Lankarani, H.M. Dynamic behaviour of planar rigid multibody systems including revolute joints with clearance, Proceedings of the Institution of Mechanical Engineers, Part-K Journal of Multi-body Dynamics, 2007, 221(2), 161-174.

35. Flores, P., Ambrósio, J., Claro, J.C.P. and Lankarani, H.M. Spatial revolute joints with clearance for dynamic analysis of multibody systems, Proceedings of the Institution of Mechanical Engineers, Part-K Journal of Multi-body Dynamics, 2006, 220(4), 257-271.

36. Flores, P., Ambrósio, J., Claro, J.C.P., Lankarani, H.M., and Koshy, C.S. Lubricated revolute joints in rigid multibody systems, Nonlinear Dynamics, 2009, 56(3), 277-295.

37. Gajos K. Z., Wobbrock J. O. and Weld D. S. Automatically generating user interfaces adapted to users' motor and vision capabilities. ACM Symposium on User Interface and Software Technology (UIST) 2007. 231-240.

38. Giese, M., Mistrzyk, T., Pfau, A., Szwillus, G. and von Detten, M. AMBOSS: A Task Modelling Approach for Safety-Critical Systems. Proc. of 7th Int. Workshop on TAsk MOdels and DIAgrams TAMODIA'2008 (Pisa, 25-26 September 2008), Lecture Notes in Computer Science, Springer-Verlag, Berlin. Available at: http://wwwcs.uni-paderborn.de/cs/ag-szwillus/lehre/ws05_06/PG/PGAMBOSS/index.php

39. Gomes de Sousa, L. & Leite, J. C. (2003), XICL: a language for the user's interfaces development and its components. Proceedings of the Latin American conference on Human-computer interaction (Rio de Janeiro, Brazil, August 17 - 20, 2003), ACM Press pages, New York, pp. 191-200.

40. Gray W. D. and Sabnani H. "Why you can't program your VCR, or, predicting errors and performance with production system models of display-based action." Conference Companion On Human Factors In Computing Systems in ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI) 1994. 79-80.

41. Gray W., Young R.M. and Kirschenbaum S. Introduction to this special issue on cognitive architectures and human-computer interaction. Human-Computer Interaction 12 (1997): 301-309.

42. Guerrero García, J., González Calleros, J.M., Vanderdonckt, J., Muñoz Arteaga, J., A Theoretical Survey of User Interface Description Languages: Preliminary Results, Proc. of Joint 4th Latin American Conference on Human-Computer Interaction-7th Latin American Web Congress LA-Web/CLIHC'2009 (Merida, November 9-11, 2009), E. Chavez, E. Furtado, A. Moran (Eds.), IEEE Computer Society Press, Los Alamitos, 2009, pp. 36-43.

43. Guerrero Garcia, J., Vanderdonckt, J., Gonzalez Calleros, J.M. (2008), Towards a Multi-Users Interaction Meta-Model. IAG Working Paper 08/28, Université catholique de Louvain, 2008.

44. Hick W.E. On the rate of gain of information. Journal of Experimental Psychology 4 (1952): 11-26.

45. Hignett S. and McAtamney L., "Rapid Body Entire Assessment (REBA)," Applied Erghonomics, vol. 31, pp. 201-205, 2000.

46. Hornof A. J. and Kieras D. E. Cognitive Modeling Reveals Menu Search Is Both Random And Systematic. ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI) 1997. 107-114.

47. Howes A., Vera A., Lewis R.L. and Mccurdy, M. Cognitive Constraint Modeling: A Formal Approach To Reasoning About Behavior. Annual meeting of the Cognitive Science Society Lawrence Erlbaum Associates, 2004.

48. Hwang F. et. al. Cursor Characteristics And Haptic Interfaces For Motor Impaired Users. Cambridge Workshop on Universal Access and Assistive Technology 2002. 87-96.

49. IBM (2002), WSXL specification, April 2002, retrieved on Janury 2nd 2009.

50. John B. E.(2010), Reducing the Variability between Novice Modelers: Results of a Tool for Human Performance Modeling Produced through Human-Centered Design. Behavior Representation in Modeling and Simulation (BRIMS), 2010.

51. John B. E. and Kieras D. The GOMS Family of User Interface Analysis Techniques: Comparison And Contrast. ACM Transactions on Computer Human Interaction 3 (1996): 320-351.

52. John B. E., Prevas, Salvucci & Koedinger (2004), Predictive Human Performance Modeling Made Easy, ACM/SIGCHI Conference on Human Factors in Computing Systems 2004

53. Johnson, P. and Johnson, H., *Knowledge analysis of task: Task analysis and specification for human-computer systems*. In A. Downton, (Ed.). Engineering the human-computre interface (pp. 119-144), London: McGraw-Hill (1989).

54. Johnson-Laird P.A. The Computer and The Mind. Cambridge, MA, USA: Harvard University Press, 1988.

55. Keates S., Clarkson J. and Robinson P. Investigating The Applicability of User Models For Motion Impaired Users. ACM/SIGACCESS Conference on Computers and Accessibility (ASSETS) 2000. 129-136.

56. Kee, D. and Karwowski, W., "LUBA: an assessment technique for postural loading on the upper body based on joint motion discomfort and maximum holding time," Applied Ergonomics, vol. 32, pp. 357-366, 2001.

57. Kieras D. and Meyer D. E. An Overview of The EPIC Architecture For Cognition And Performance With Application to Human-Computer Interaction. Human-Computer Interaction 12 (1990): 391-438.

58. Kieras D. E. Fidelity Issues In Cognitive Architectures For HCI Modelling: Be Careful What You Wish For. International Conference on Human Computer Interaction 2005.

59. Kieras D. E., Wood S. D., Abotel K. and Hornof A. GLEAN: A Computer-Based Tool For Rapid GOMS Model Usability Evaluation of User Interface Designs. ACM Symposium on User Interface and Software Technology (UIST) 1995. 91-100.

60. Kieras D. GOMS Modeling of User Interfaces Using NGOMSL. Conference Companion On Human Factors In Computing Systems in ACM/SIGCHI Conference on Human Factors in Computing Systems 1994. 371-372.

61. Kobsa, A. (2004): Adaptive Methoden – Benutzermodellierung. Grundlagen der Information und Dokumentation (5th completely revised edition), München, K. G. Saur, S.299-302

62. Kost, S. (2004), Dynamically generated multi-modal application interfaces. Ph.D. Thesis, Technical University of Dresden and Leipzig University of Applied Sciences, Germany.

63. Laird J.E., Rosenbloom P.S. and Newell A. Towards chunking as a general learning mechanism. National Conference on Artificial Intelligence at Austin, TX: Morgan 1984. 188-192.

64. Lallement Y. and Alexandre F. Cognitive Aspects of Neurosymbolic Integration. Connectionist-Symbolic Integration Ed. Sun R. and Alexandre F. London, UK: Lawrence Erlbaum Associates, 1997.

65. Luyten, K., Abrams, M., Vanderdonckt, J. & Limbourg, Q. (2004) Developing User Interfaces with XML: Advances on User Interface Description Languages, Sattelite workshop of Advanced Visual Interfaces 2004, Gallipoli, Italy.

66. Mahfoudhi, A., Abed, M.,&Tabary,D. (2001). From the formal specifications of user tasks to the automatic generation of the HCI specifications. In A. Blandford, J. Vanderdonckt, & P. Gray, (Eds.), *People and computers XV* (pp. 331–347). London: Springer.

67. Mankoff J., Fait H. and Juang R. Evaluating accessibility through simulating the experiences of users with vision or motor impairments. IBM Sytems Journal 44.3 (2005): 505-518.

68. Marr D. C. Visual Information Processing: the structure and creation of visual representations. Philosophical Transactions of the Royal Society of London 290.1038 (1980): 199-218.

69. Mcmillan W. W. Computing For Users With Special Needs And Models of Computer-Human Interaction. ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI) 1992. 143-148.

70. Meireles, F., Silva, M., and Flores, P. Kinematic analysis of human locomotion based on experimental data, Proceedings of the VipImage2007, Thematic Conference on Computational Vision and Medical Image Processing, Porto, Portugal, October 17-19, 2007, 6p.

71. Meyer D. E., Abrams R. A., Kornblum S., Wright C. E. and Smith J. E. K. Optimality in human motor performance: Ideal control of rapid aimed movements. Psychological Review 95 (1988): 340-370.

72. Moran T.P. Command Language Grammar: A Representation For The User Interface of Interactive Computer Systems. International Journal of Man-Machine Studies 15.1 (1981): 3-50.

73. Nejime Y. and Moore B.C.J., Simulation of the effect of threshold elevation and loudness recruitment combined with reduced frequency selectivity on the intelligibility of speech in noise, J. Acoust. Soc. Am.(1997) 102: 603-615

74. Newell A. and Simon H. A. GPS, A Program That Simulates Human Thought. Cambridge, MA, USA: MIT Press, 1995.

75. Newell A. Unified Theories of Cognition. Cambridge, MA, USA: Harvard University Press, 1990.

76. Newell A. You can't play 20 questions with nature and win. Projective comments on the papers of this symposium. Pittsburgh, Pa: Carnegie Mellon University, Department of Computer Science. 1973.

77. Oka N. Hybrid cognitive model of conscious level processing and unconscious level processing. IEEE International Joint Conference on Neural Networks 1991. 485-490.

78. Parenti-Castelli V., Di Gregorio R., Parallel mechanisms applied to the human knee passive motion simulation, 7th ARK, International Symposium on Advances in Robot Kinematics Piran-Portoroz, Slovenia, 26–30 June 2000.

79. Paternò, F. & Santoro, C. (2003), A Unified Method for Designing Interactive Systems Adaptable to Mobile and Stationary Platforms, Interacting with Computers, Elsevier, 15, pp. 349-366.

80. Paternò, F., Model Based Design and Evaluation of Interactive Applications. Springer Verlag, Berlin (1999).

81. Peck V. A. and John B. E. "Browser-Soar: a computational model of a highly interactive task." ACM/SIGCHI conference on Human factors in computing systems (CHI) 1992. 165-172.

82. Picard, E., Fierstone, J., Pinna-Dery, A-M. & M. Riveill (2003). Atelier de composition d'ihm et évaluation du modèle de composants. Livrable l3, RNTL ASPECT, Laboratoire I3S, mai.

83. Pinelle, D., Gutwing, C., and Greenberg, S. Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. ACM Transactions on Computer-Human Interaction10, 4 (2003), 281-311.

84. Puerta A.R. (1996), The Mecano Project: Comprehensive and Integrated Support for Model-Based Interface Development, Proceedings of 2nd Int. Workshop on Computer-Aided Design of User Interfaces CADUI'96 (Namur, 5-7 June 1996), Presses Universitaires de Namur, 1996, pp. 19-35.

85. Raasch, C., Zajac, F., Ma, B. and Levine, S. Muscle Coordination of Maximum-Speed Pedaling, Journal of Biomechanics, 1997, 30(6) 595-602.

86. Rasmussen, J., Damsgaard, M., Christensen, S. and Surma, E. Design Optimization with Respect to Ergonomic Properties, Structural and Mu1tidisciplinan Optimization, 2002, 24(2), 89-97.

87. RFC 2533: A Syntax for Describing Media Feature Sets; G. Klyne; IETF Request for Comments: ftp://ftp.isi.edu/in-notes/rfc2533.txt

88. Rieman J. and Young R. M. "A dual-space model of iteratively deepening exploratory learning." International Journal of Human-Computer Studies 44 (1996): 743-775.

89. Schaefer, R., Steffen, B., Wolfgang, M., Task Models and Diagrams for User Interface Design, Proceedings of 5th International Workshop, TAMODIA'2006 (Hasselt, Belgium, October 2006), Lecture Notes in Computer Science, Vol. 4385, Springer Verlag Berlin, 2006, pp. 39-53.

90. Serna A., Pigot H. and Rialle V. Modeling the progression of Alzheimer's disease for cognitive assistance in smart homes. User Modeling and User-Adapted Interaction 17 (2007): 415-438.

91. Silva, M.P.T. and Ambrósio, J.A.C. Kinematic Data Consistency in the Inverse Dynamic Analysis of Biomechanical Systems, Multibody System Dynamics, 2002, 8(2), 219-239.

92. Stephanidis C., Paramythis A., Sfyrakis M., Stergiou A., Maou N., Leventis A.,Paparoulis G. and Karagiannidis C. Adaptable And Adaptive User Interfaces for Disabled Users in the AVANTI Project. Intelligence in Services and Networks, LNCS-1430, Springer-Verlag 1998. 153-166.

93. The COSPAL Project, Available at: http://www.cospal.org/, Accessed on: 6th July 2011

94. Tobii Eye Tracker. Available at http://www.imotionsglobal.com/Tobii+X120+ Eye-Tracker.344.aspx Accessed on 12th December 2008

95. Tollinger I., Lewis R. L., McCurdy M., Tollinger P., Vera A., Howes A. and Pelton L. Supporting Efficient Development of Cognitive Models At Multiple Skill Levels: Exploring Recent Advances In Constraint-Based Modeling. ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI) 2005. 411-420.

96. van der Veer, G.C., van der Lenting, B.F., Bergevoet, B.A.J., GTA: Groupware Task Analysis – Modeling Complexity. Acta Psychologica 91 (1996) 297–322

97. van Welie, M. Task-Based User Interface Design. SIKS Dissertation Series 2001-6. Vrije Universiteit, Amsterdam. Nl.(2000).

98. Vanderdonckt, J. A MDA-Compliant Environment for De-veloping User Interfaces of Information Systems. In Proc. of 17th Conf. on Advanced Information Systems Engineering CAiSE'05 (Porto, 13-17 June 2005), LNCS, Vol. 3520, Springer-Verlag, Berlin, 2005, pp. 16-31.

99. Winograd T. Architectures for Context. HUMAN-COMPUTER INTERACTION, 2001, Volume 16, pp. 401–419.

100. Young R.M., Green T.R.G. and Simon T. "Programmable User Models For Predictive Evaluation of Interface Designs." ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI) 1989. 15-19.

101. Zanuttini, R., Cielo, P. and Poncino, D., "Il metodo OWAS. Prime applicazioni nella valutazione del rischio di patologie muscolo-scheletriche nel settore forestale in Italia", Forest@, vol. 2, no. 2, pp. 242-255, 2005.

102. Zimmerman G., Vanderheiden G. A dream... The Universal Remote Console: http://trace.wisc.edu/docs/2010-URC-ISOFocus/ (visited 29.05.2011)

ANNEX

VUMS cluster meetings and actions

Cluser Workshop in Pargue November 2011

A VUMS cluster meeting was held in November 29 in Prague in conjunction with the VERITAS first international workshop. A list on the necessary actions that should follow has been defined and is analyzed in the following:

Action 1: Definition of Terms, Vocabulary:

A list of terms to be defined by each project of the VUMS cluster has been extracted. Each project should provide a glossary of these terms in their respective deliverables on user models that are due on M12 (December 2010 for VICON and VERITAS and January 2011 for GUIDE and MyUI).

List of Terms to be defined in the glossary

- User model
- Classification of user models
- Application domains covered
- User Profile
- Virtual User
- Virtual human
- Simulation
- Model Validation (Metrics and Process)
- Interface Adaption
- Profile Adaptation
- (User Interface) Design Pattern
- User Interaction
- Disablility, capabilities and functional limitations

Moreover, the following task and related deliverables should be included/ modified in the DoW of all projects of the VUMS cluster.

Tx (Joint Task) Standardisation concertation actions

This is a joint task of FP7 projects VERITAS, VICON, MyUI, GUIDE that aims at supporting the harmonisation of research in standardisation area in order to guarantee efficient and in-time exchange of work done in this field in the aforementioned projects and avoid overlaps and potential fragmentations between projects.

A special Task Force will be organised from the beginning of the project that will be coordinated by VERITAS and will be comprised by two representatives from each project (the Project Manager of the project and the representative of the organisation responsible for standards in each project).

This Task Force will meet in all workshops held in the context of VERITAS, after the 2nd year of life (a special session dedicated to this will be organised in each case by

VERITAS), in order to update each other on the status of the work and outcomes of each project in standardisation. In the context of these sessions, common intended actions on standardisation will be scheduled, related, for example, to the common approach to be followed for projects' active involvement in international mainstream standardisation organisations such as W3C, ISO/IEC, ITU, ETSI, CEN, or CENELEC and the collaboration with standardisation organisations like the ETSI Special Task Force 342 on Personalization and User Profile Standardization.

Further VUMS standardisation concertation meetings will be held at m13 and m26 together with representatives from related EU-projects (e.g. VAALID) in order to analyse and compare the respective user modelling approaches and prepare common statements and deliverables on the levels of terminology, concepts and technologies. The outcome of this work will be a series of policy and standards recommendations that will be summarized in an interim report (D6.4) and a final standardisation report (D6.5).

D2.6 (GUIDE & MyUI) Interim Report on VUMS cluster standardisation (M17)

The interim report will document the standardisation activities of the first period of the VUMS cluster and a strategic action plan for further standardisation activities. Moreover, a draft version of potential standardisation input to be fed directly into standardisation bodies will be prepared. This includes common definitions of the most important terminology in the field of user modelling and adaptive user interfaces.

D8.5 (GUIDE & MyUI) Input to standardisation and report on VUMS cluster standardisation activities (M30)

The final standardisation report documents the joint standardisation activities throughout the common lifecycle of the VUMS cluster projects. It includes also concrete definitions, requirements and specifications in formats which can be directly fed into standardisation bodies.

Output of the action*:* List of terms and glossaries for all VUMS cluster projects (M12, December 2010 for VICON and VERITAS and January 2011 for GUIDE and MyUI)

Action 2: Workshop on common definition of a virtual user model

A VUMS cluster workshop will be held during 15-17.03.2010 so as to come to common definition on the concepts defined in each of the VUMS cluster projects. The workshop will also compare the human modelling approaches documented in the respective deliverables and identify clues for standardization. It will be held in Stuttgart and representatives from all VUMS cluster projects will participate. Moreover, the "D2.2 Concept and services ontology description" of VAALID will be considered.

Output of the action*:* Analysis of the virtual user models and modeling technologies used (M15, March 2010).

Action 3: Common VUMS cluster virtual user models and extensions:

The common part of the virtual user models developed within the VUMS cluster projects will be defined based on the analysis of “Action 2”. The common part will refer to general, interoperable parts and properties of the virtual user models and will be the main result subject to standardization. Moreover, each project of the VUMS cluster will further describe extensions of this main, common virtual user model for specific purposes and application scenarios related to their project objectives.

Output of the action: Definition of the common virtual user model and extension in a separate document to be provided by June 2011.

Action 4: Data format standardization:

Based on the result of “Action 3”, the initialization of a new “working group” or the participation in an existing “working group” of a specific standardization body will be decided. This action will start on July 2011 and by the end of the year 2011 standardization activities in the context of a specific working group should have been started.

Output of the action: Initialization of the standardization process as defined in “Action 3”.

In the following figure, a simple Gantt chart is depicted, illustrating the timing of the major actions towards standardization planned by the VUMS cluster.

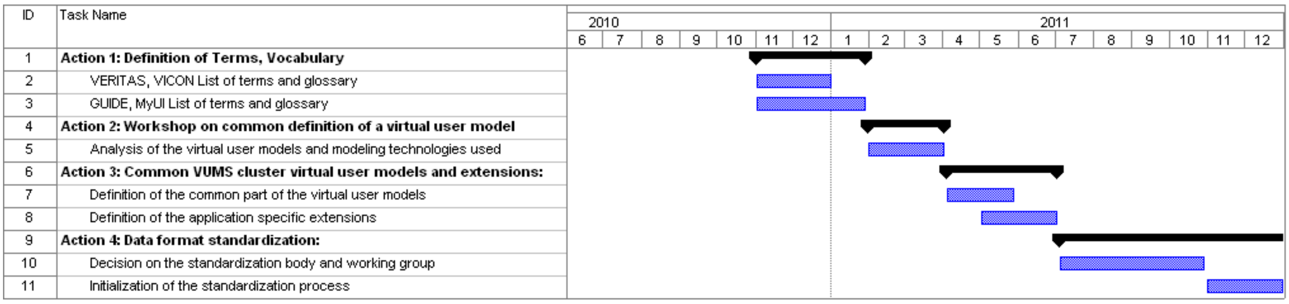


Figure 12. Timeline for the VUMS cluster standardization actions

Cluster Workshop in Stuttgart 17th and 18th of March 2011

This workshop was conducted after the annual reviews of the cluster projects and the annual review of the VUMS cluster itself. The main suggestions from the reviewers regarding the VUMS cluster where the following:

- Resubmit D1.6.4
  - Each Project has to provide input
  - Add more standards
  - Include common terminology
  - Sections on user models per project
  - Common section on status after Stuttgart
- Establish contact with responsible persons for ISO/IEC 24751
  - Standard is currently being revised
- Collaboration with future projetcs
- Add references to VAALID (each project’s and VUMS website)

We have already taken actions in response to the reviewers' suggestions prior to this workshop:

- Cluster contacted Jutta Treviranus from ISO/IEC 24751 and reported about the results.
- We will offer our services to future projects and will be open for collaboration. VUMS is not interested in a big cluster starting from the scratch again. It would hinder work more than support it.
- References to VAALID have been added.

Short presentations of user models

During the first part of the workshop, the user models of all the projects were presented. Similarities and differences between projects' user modeling approaches were identified.

Workshop on terminology

In this part of the workshop, we have started a collaborative effort to harmonize the different taxonomies and definitions within the cluster projects. There we have identified the scope and context of our models, which is the following:

Adaptation of a human-machine-interface to the needs of the real user or simulation of the interaction between a human and a product, in order to design the product.

In the further course of the workshop we have worked on harmonizing the terminology between all projects which was already started from the beginning of the cluster and intensified in the workshop in Prague (see section 0). The started harmonization process was scheduled to the participants of cluster for finalization after the workshop to create a glossary as a abasement for all projects. The final glossary has been incorporated into this deliverable (see section 1)

What to standardize?

In this part oft the workshop we tried to identify what to standardize, here we agreed on the following:

- Taxonomy
- User model variables

The main goal of the standardization will be the interoperability of user profiles, which will enable their exchanging, import, export and sharing.

Discussion

We have discussed many aspects and agreed on the following ideas:

- Extensible user models

    - Extensibility is essential, because completeness cannot be assumed and different purposes of use.
    - One way to handle an extensible taxonomy could be the supply of a user model repository, which will define namespaces and will use the URI concept.
- Recommendations
    - Further developement of teminology and taxonomy of the user model
    - Prepare a white paper
    - Organize an open workshop to discuss our ideas/concepts with other experts
    - Send the prepared white paper together with an invitation
    - Collect feedback from experts and discuss about cluster's standardisation efforts.

User Models Taxonomy

In order to define the taxonomy of the user models, we first defined the hierarchy structure and started with some example items at different levels of the hierarchy.

After the Stuttgart workshop several teleconferences have taken place so as to coordinate activities related to the stnadardization actions and contributions to 24751 "ISO IEC".

Plan for standardization

As mentioned in the sections above we have contacted 24751 "ISO IEC" to check if our contribution from the VUMS cluster would fit in the frame of this standard. After exchanging of mutual information, we agreed on the fact that we can contribute to this standard. The next step was to elaborate on how we can contribute to the standard and the committee of the 24751 "ISO IECadvised us that the best way for collaboration and participation could go through our national body - that would grant us all rights.

Plan to achieve our goal to contribute to the 24751 standard:

- First register as national experts with the related national body. For Germany that is DIN. For UK through BSI and the SC36 mirror committee.
- Usually each mirror committee appoints their delegation to different WG's and the plenary meeting. The suggestion was to be part of WG7, and WG4.
- For UK, BSI will nominate experts to participate at the different working groups, so no invitation is necessary.
- It is also important to get access to the document registries for SC36 and the relevant working groups - this is managed by the National Bodies.