



IST Amigo Project
Deliverable D4.3
"First version of software services"
Overview

IST-2004-004182
Public

Project Number	:	IST-004182
Project Title	:	Amigo
Deliverable Type	:	Report

Deliverable Number	:	Deliverable D4.3
Title of Deliverable	:	First version of software services - Overview
Nature of Deliverable	:	Public
Internal Document Number	:	amigo-d4.3-final
Contractual Delivery Date	:	28 February 2006
Actual Delivery Date	:	31 March 2006
Contributing WPs	:	WP4
Author(s)	:	France Telecom, Ikerlan, FhG IPSI, Inria, IMS, Knowledge, PAE, Philips, Telin, VTT

Abstract

This report presents the properties of the Intelligent User Services (IUS) software modules that will be developed in the Amigo project.

Keyword list

Connected home, ambient intelligence, service orientation, context awareness, context management, user modeling and profiling, personalization, awareness and notification, rule based, user interface, speech recognition, gesture recognition, multimodal dialogues, home information and entertainment applications, home care and safety applications, extended home environment, trust, privacy, awareness systems, personal security.

Table of Contents

Table of Contents	3
1 Context Management Service	5
1.1 Context Wrapper	5
1.1.1 Acoustic Position Estimation Sensor	6
1.1.2 Speaker Identification Sensor	7
1.1.3 Topic Recognition Sensor	9
1.1.4 Wireless positioning	10
1.2 Context Interpreter	12
1.3 Context Source	13
1.3.1 Context History Component	14
1.4 Context Broker	16
2 User Modeling and Profiling Service	18
2.1 Reasoning Module	18
2.2 Static Modeler	19
2.3 Feedback Analyzer	21
2.4 Context Module	22
2.5 Dynamic Modeler	23
2.6 Multi-Profile Aggregator	25
3 Awareness and Notification Service	27
3.1 Rule Manager	27
3.2 Controller	28
3.3 Event Monitor	29
3.4 Notification Profile Manager	31
3.5 Notifier	32
4 User Interface Service	34
4.1 Voice Service	34
4.1.1 Signal acquisition and preprocessing	34
4.1.1.1 Speaker Recognition Component	34
4.1.2 Explicit speech interaction	35
4.1.3 Implicit speech input	37
4.1.4 Single Channel Speech Enhancement	38
4.1.5 Multi Channel Speech Enhancement	40
4.1.6 Acoustic Scene Analysis Service	41
4.2 Gesture Service	42
4.2.1 2D Gesture Service	42
4.2.2 3D Gesture Service	44

4.3 GUI Service	46
4.4 Multimodal Dialog Manager	48
4.5 Multimodal Fusion.....	49
4.6 Multidevice Service.....	51
5 Privacy and Personal Security	53
5.1 Perceived Privacy rules	53

1 Context Management Service

1.1 Context Wrapper

Provider

IKER

Introduction

Physical sensors will be encapsulated within a wrapper which will make them view as a ContextSource to context consumers. The wrapper will provide a homogenous interface to the CMS.

Development status

Development has not started yet

Intended audience

Sensor providers

License

The software itself is under LGPL license, but it might make use of proprietary binaries/libraries for which no source code is provided. The sensor may need a proprietary library to access to it.

Language

C#

Environment (set-up) info needed if you want to run this sw (service)

Hardware: Windows PC

Software: Windows XP, .NET 2.0

Visual Studio 2005

Platform

Any computer supporting previous environment

Tools

Visual C# .NET 2005 for compiling the source code

Files

Software code will be available.

Documents

Partially explained in D4.2. More deeply information will be done after implementation.

Tasks

Development

Bugs

None yet

Patches

None yet

1.1.1 Acoustic Position Estimation Sensor**Provider**

PAE

Introduction

Speaker position estimation is computed by evaluating the speech signals captured by multi-channel acoustic sensors. These sensors or microphones are linearly arranged in a microphone array. With each array it will be possible to estimate the “direction-of-arrival”, i.e. the angle towards an active (currently speaking) person. It is not possible to track a non-speaking person with these sensors, since then no input data is available. A group of speaking persons is handled by the sensor, by detecting the loudest speaker.

Development status

A first software version is under development and will be soon available.

Intended audience

Sensor wrapper developer

License

This software is proprietary software by PAE. It will not be made open source since it is not middleware software. Special arrangements can be made with Amigo partners.

Language

C++ / C on Linux operating systems

Environment (set-up) info needed if you want to run this sw (service)

Hardware: multi-channel soundcard; microphones; PC

Software: SuSe Linux 9.2; Jack Audio Connection Kit;

Platform

PC with SuSe Linux operating system (9.2 or higher)

Tools

-

Files

One executable (asreng – main engine), modular based shared libraries and special configuration files

Documents

Installation guide and usage guidelines will be made available.

Tasks

Reduction of computational demands, additional experiments and usage tests.

Bugs

No known bugs

Patches

Intermediate critical patches

1.1.2 Speaker Identification Sensor**Provider**

KNOW, PAE

Introduction

Speaker recognition component will provide two basic functionalities: enrollment of new users and recognition of enrolled users.

- *Enrollment* happens when a new user joins the system – a person speaks and the application builds a *voiceprint model* for that person.
- *Recognition* happens in the future whenever the application wants to ensure the identity of a speaker – the user speaks and the speech is compared to existing voiceprint models. If the comparison scores above a predefined threshold, the identity is validated, otherwise the identity claim is rejected.

Development status

First version 1.0 of engine software is deployed as WIN32 dlls.

Intended audience

Project partners

License

Speaker recognition engine:

KNOWLEDGE owns the IPR for speaker recognition engine software. Licensing negotiable with separate agreement.

Speaker recognition server:

It will be developed as open source software (LGPL)

Language

C#, C++

Environment (set-up) info needed if you want to run this sw (service)

Hardware:

- PC computer with network connection

Software:

- Windows 2K
- .NET 2.0

Platform

.NET

Tools

-

Files

No source code available yet.

Documents

No documentation yet

Tasks

For D4.3 KNOWLEDGE will develop the speaker recognition server (basic functionality).

Bugs

-

Patches

-

1.1.3 Topic Recognition Sensor**Provider**

INRIA

Introduction

This module will provide contextual information about the topic of discussion between users. Obviously, such information will be available only when someone (the user) is talking. A typical scenario is when several people are talking together, either face to face or via teleconferencing or phone. However, the information provided by this module can also be computed when the user is listening to the radio, or when he is listening to a recorded talk.

Development status

Specification stage

Intended audience

Project partners

License

LGPL

Language

JAVA

Environment (set-up) info needed if you want to run this sw (service)

Requires a real-time large vocabulary continuous speech recognizer;

Requires a topic ontology to be defined by the application.

Platform

Hardware: desktop or laptop PC

Software:

Native linux or cygwin

J2EE version 1.4, Sun Java System Application Server

Tools

-

Files

Source code

Documents

None

Tasks

- Complete the interfaces
- Implement the functionalities
- Write the documents
- Test, debug, validate

Bugs

No bugs reported for now

Patches

No patches created for now

1.1.4 Wireless positioning

Provider**Philips****Introduction**

This module will implement a context wrapper that is specific for RFID (or similar) based positioning of Users and Devices.

Development status

Design stage

Intended audience

Project partners

License

BSD (only the wrapper will be provided)

Language

C#

Environment (set-up) info needed if you want to run this sw (service)

PC running Windows

Specific Hardware and Software drivers are needed.

Platform

Amigo .NET based deployment framework

Tools

-

Files

Source code

Documents

None

Tasks

- Complete the design
- Implement the functionalities

- Test, debug, validate

Bugs

No bugs reported for now

Patches

No patches created for now

1.2 Context Interpreter

Provider

TELIN

Introduction

A Context Interpreter combines multiple sources of information (possibly over time) in order to obtain higher-level information about particular entities. Although a context interpreter is a functional element, and not a specific component, this particular Context Interpreter is intended as an example of how information from multiple context sources can be combined.

Development status

Not developed yet

Intended audience

Context aware application developers and context source developers

License

LGPL

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

PC, JVM, OSGi-Based Programming & Deployment Framework from WP3

Platform

JVM/OSGi-Based Programming & Deployment Framework

Tools

-

Files

-

Documents

None specific yet. For an overview see D4,2 chapter 2.

Tasks

-

Bugs

-

Patches

-

1.3 Context Source

Provider

TELIN

Introduction

The context source (as a baseclass of ContextWrapper and ContextStorageService) provides both synchronous and subscription-based interfaces.

Development status

Development in progress, existing Context Source code has to be modified to the Amigo specific environment.

Intended audience

Context aware application developers and context source developers

License

LGPL

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

PC, JVM, OSGi-Based Programming & Deployment Framework from WP3

Platform

JVM, OSGi-Based Programming & Deployment Framework from WP3

Tools

-

Files

-

Documents

None specific yet. For an overview see D4,2 chapter 2.

Tasks

-

Bugs

-

Patches

-

1.3.1 Context History Component**Provider****IPSI****Introduction**

The context history component contributes to the context management intelligent user service. Based on context histories, the context management service will provide data that helps AMIGO applications to get to know users over time and if possible do predictions based on patterns found. Reasoning based on context histories, as opposed to ontology-based reasoning, will especially take into account an enhanced understanding of the users'

interactions over time and be embedded into and related to their other interactions happening in parallel, before, or afterwards (time-based reasoning).

Development status

The implementation of the first version has just started.

Indented audience

Project partners at first and then later also other developers of custom AMIGO services and applications.

License

The software itself is under LGPL license, but it might make use of proprietary binaries/libraries for which no source code is provided.

Language

C#

Environment (set-up) info needed if you want to run this sw (service)

Hardware: Windows PC

Software: Windows XP, .NET 2.0

Indicate all special needs

Platform

-

Tools

Visual C# .NET 2005 for compiling the source code

Files

Software code will be available.

Documents

Design documentation is provided in D4.2.

Tasks

The implementation of the first version has just started.

Bugs

The implementation of the first version has just started.

Patches

The implementation of the first version has just started.

1.4 Context Broker

Provider

FT

Introduction

The context broker is a directory service which is aware of context sources that are actually available. Based on this information it is able to connect context providers to context consumers. Conceptually, this information could be viewed as a table of context sources identifiers indexed by context source capabilities.

Development status

Not developed yet

Intended audience

Context aware application developers and context source developers

License

LGPL

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

PC, Jena, OWLApi, OpenSLP

Platform

JVM

Tools

-

Files

-

Documents

Not available yet

Tasks

Development including adaptation of OpenSLP

Bugs

None yet

Patches

None yet

2 User Modeling and Profiling Service

User modeling and profiling provides the methodology to enhance the effectiveness and usability of services and interfaces in order to (a) tailor information presentation to user and context, (b) reason about user's future behavior, (c) help the user to find relevant information, (d) adapt interface features to the user and the context in which it is used, (e) indicate interface features and information presentation features for their adaptation to a multi-user environment. These goals are achieved by constructing, maintaining and exploiting user models and profiles, which are explicit representations of individual users preferences. User profile update will be performed using static and dynamic modeling methodologies.

2.1 Reasoning Module

Provider

KNOW

Introduction

This component is responsible for exploring the user profile and responding to other services requests either for parts of the user profile (collection of user preferences for a particular situation) or discrete preferences.

Development status

Working on first version 1.0 of software.

Intended audience

Project partners

License

LGPL

Language

C#/C++

Environment (set-up) info needed if you want to run this sw (service)

Software:

- Windows 2K/XP
- .NET 2.0

Platform

.NET

Tools

Visual C# 2005 Express

Files

Source code will be made available.

Documents

UMPS architecture is described on D4.2. Separate software installation guide will be provided with the Service.

Tasks

Developing interfaces
Implementation of basic functionality
Design of more complex functionality implementation

Bugs

-

Patches

-

2.2 Static Modeler

Provider

KNOW

Introduction

This component is responsible for the creation, removal and modification of user profiles at user's or application's request. For the user, a GUI will be provided, allowing him to modify preference values, or to enable or disable the modeling of a series of preferences corresponding to a branch in the user profile.

Development status

Working on first version 1.0 of software.

Intended audience

Project partners

License

The software itself will be under LGPL license, but might make use of proprietary binaries/libraries for which no source code is provided.

Language

C#/C++

Environment (set-up) info needed if you want to run this sw (service)

Software:

- Windows 2K/XP
- .NET 2.0

Platform

.NET

Tools

Visual C# 2005 Express

Files

Source code for managing explicitly acquired user profile will be made available

Documents

UMPS architecture is described on D4.2. Separate software installation guide will be provided with the Service.

Tasks

Developing interfaces
Implementation of basic functionality
Design of more complex functionality implementation

Bugs

-

Patches

-

2.3 Feedback Analyzer

Provider

VTT

Introduction

This component will enable the update of the profile at system's initiative, based on explicit or implicit user feedback. In case of explicit feedback the user answers to well-defined system questions, while for implicit feedback meaningful semantics are automatically extracted by the system from user's comments.

Development status

Implementation of Dynamic Modeler, Feedback Analyzer and Multi-Profile Aggregator has not started yet because they are needed for Dynamic Modeler part, which is scheduled to be implemented after the Static Modeler.

Intended audience

Project partners

License

The software itself will be under LGPL license, but might make use of proprietary binaries/libraries for which no source code is provided.

Language

C#/C++

Environment (set-up) info needed if you want to run this sw (service)

Software:

Platform

.NET

Tools

Visual C# 2005 Express

Files

Source code will not be necessarily made available. Parts which will not be available as source code will be provided as dll.

Documents

UMPS architecture is described on D4.2. Separate software installation guide will be provided with the Service.

Tasks

Developing interfaces
Implementation of basic functionality
Design of more complex functionality implementation

Bugs

-

Patches

-

2.4 Context Module**Provider**

IPSI

Introduction

This component receives «context changed» events, and request the new context information whenever necessary.

Development status

-

Intended audience

Project partners

License

The software itself is under LGPL license, but it might make use of proprietary binaries/libraries for which no source code is provided.

Language

C#

Environment (set-up) info needed if you want to run this sw (service)

Software:

Platform

.NET

Tools

Visual C# 2005 Express

Files

-

Documents

UMPS architecture is described on D4.2. Separate software installation guide will be provided with the Service.

Tasks

Developing interfaces
Implementation of basic functionality
Design of more complex functionality implementation

Bugs

-

Patches

-

2.5 Dynamic Modeler**Provider**

VTT, KNOW

Introduction

This component is responsible for the modification (update) of the user profile using the logging data, resulted from implicit or explicit user feedback. This update can be made either by activating additional stereotypes, as a result of triggering events, or by learning of new parts of user profile in different way, e.g., by training a classifier to make decisions.

Development status

Implementation of Dynamic Modeler, Feedback Analyzer and Multi-Profile Aggregator has not started yet because they are needed for Dynamic Modeler part, which is scheduled to be implemented after the Static Modeler.

Intended audience

Project partners

License

The software itself will be under LGPL license, but might make use of proprietary binaries/libraries for which no source code is provided.

Language

C#/C++

Environment (set-up) info needed if you want to run this sw (service)

Software:

- Windows 2K/XP
- .NET 2.0

Platform

.NET

Tools

Visual C# 2005 Express

Files

Source code will not be necessarily made available. Parts which will not be available as source code will be provided as dll.

Documents

UMPS architecture is described on D4.2. Separate software installation guide will be provided with the Service.

Tasks

Developing interfaces
Implementation of basic functionality
Design of more complex functionality implementation

Bugs

-

Patches

-

2.6 Multi-Profile Aggregator

Provider

VTT

Introduction

This component provides an aggregated profile in case of multiple users found in the same context (i.e. the same room).

Development status

Implementation of Dynamic Modeler, Feedback Analyzer and Multi-Profile Aggregator has not started yet because they are needed for Dynamic Modeler part, which is scheduled to be implemented after the Static Modeler.

Intended audience

Project partners

License

The software itself is under LGPL license, but might make use of proprietary binaries/ libraries for which no source code is provided.

Language

C#/C++

Environment (set-up) info needed if you want to run this sw (service)

Software:

Platform

.NET

Tools

Visual C# 2005 Express

Files

Source code will not be necessarily made available. Parts which will not be available as source code will be provided as dll.

Documents

UMPS architecture is described on D4.2. Separate software installation guide will be provided with the Service.

Tasks

Developing interfaces
Implementation of basic functionality
Design of more complex functionality implementation

Bugs

-

Patches

-

3 Awareness and Notification Service

3.1 Rule Manager

Provider

TELIN

Introduction

From an external perspective, the RuleManager enables users/application developers to introduce and manage their ANS rules in the system. Internally, the RuleManager parses these rules and checks them for syntax and semantic correctness. The rules are conditioned and then forwarded to be processed by the Controller. The condition part of the rule is de-integrated into atomic event requirements passed to the EventMonitor.

Development status

Detailed design of external and internal interfaces and distinction of internal components. Implementation of the interfaces and frameworks for the internal components.

Indented audience

Application developers and end-users of the system. Although, a user interface has to be used for efficient rule creation by end-users.

License

Apache v2 license

Language

The used programming language is Java. The rules are expressed in our ECA language which is summarized in D4.2.

Environment (set-up) info needed if you want to run this sw (service)

Hardware: PC

Software: J2SE

Platform

Computer system with a JVM needed to execute the java binaries.

Tools

Eclipse for compilation of the source.

Files

-

Documents

See D4.2 for a detailed overview of the design of the RuleManager.

Tasks

-

Bugs

-

Patches

-

3.2 Controller

Provider**IPSI****Introduction**

From an external perspective, the RuleManager enables users/application developers to introduce and manage their ANS rules in the system. Internally, the RuleManager parses these rules and checks them for syntax and semantic correctness. The rules are conditioned and then forwarded to be processed by the Controller. The condition part of the rule is de-integrated into atomic event requirements passed to the EventMonitor.

Development status

Detailed design of external and internal interfaces and distinction of internal components. Implementation of the interfaces and frameworks for the internal components.

Indented audience

Application developers and end-users of the system. Although, a user interface has to be used for efficient rule creation by end-users.

License

Apache v2 license

Language

The used programming language is Java. The rules are expressed in our ECA language which is summarized in D4.2

Environment (set-up) info needed if you want to run this sw (service)

Hardware: PC

Software: J2SE

Platform

Computer system with a JVM needed to execute the java binaries.

Tools

Eclipse for compilation of the source.

Files

-

Documents

See D4.2 for a detailed overview of the design of the RuleManager.

Tasks

-

Bugs

-

Patches

-

3.3 Event Monitor**Provider**

Philips

Introduction

The main functionality of the *EventManager* component is to allow easy access to Context Management Service components and to manage the combinations of events and context sources.

Development status

Detailed design of external and internal interfaces and distinction of internal components. Implementation of the interfaces and frameworks for the internal components.

Indented audience

Project partners at first and then later also other developers of custom AMIGO services and applications.

License

BSD

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

Hardware: PC

Software: J2SE

Platform

Amigo OSGi based deployment framework. Computer system with a JVM needed to execute the java binaries.

Tools

Eclipse to compile the source code.

Files

-

Documents

No documentation yet

Tasks

-

Bugs

-

Patches

-

3.4 Notification Profile Manager**Provider****IPSI****Introduction**

The NotificationProfileManager enables ANS to access the notification preferences of users that are externally stored in the UMPS. This component also makes sure that the settings of a retrieved user notification profile are not conflicting. For more information about the data model of a user notification profile see document D4.2.

Development status

Detailed design of external and internal interfaces and distinction of internal components. Implementation of the interfaces and frameworks for the internal components.

Indented audience

Project partners at first and then later also other developers of custom AMIGO services and applications.

License

LGPL

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

Hardware: PC

Software: J2SE and Jess rule engine

Platform

Computer system with a JVM needed to execute the java binaries.

Tools

Eclipse to compile the source code.

Files

-

Documents

No documentation yet

Tasks

-

Bugs

-

Patches

-

3.5 Notifier**Provider****IPSI****Introduction**

The main task of the Notifier is to determine the right intensity for notifications and to send them to the right applications.

Development status

Detailed design of external and internal interfaces and distinction of internal components.
Implementation of the interfaces and frameworks for the internal components.

Indented audience

Project partners at first and then later also other developers of custom AMIGO services and applications.

License

LGPL

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

Hardware: PC

Software: J2SE and Jess rule engine

Platform

Computer system with a JVM needed to execute the java binaries.

Tools

Eclipse to compile the source code.

Files

Implementation is about to start

Documents

No documentation yet

Tasks

-

Bugs

-

Patches

-

4 User Interface Service

4.1 Voice Service

4.1.1 *Signal acquisition and preprocessing*

4.1.1.1 *Speaker Recognition Component*

Provider

KNOW

Introduction

Speaker recognition component will provide two basic functionalities: enrollment of new users and recognition of enrolled users.

- *Enrollment* happens when a new user joins the system – a person speaks and the application builds a *voiceprint model* for that person.
- *Recognition* happens in the future whenever the application wants to ensure the identity of a speaker – the user speaks and the speech is compared to existing voiceprint models. If the comparison scores above a predefined threshold, the identity is validated, otherwise the identity claim is rejected.

Speaker recognition component includes: speaker recognition engine and speaker recognition server.

Development status

First version 1.0 of engine software is deployed as WIN32 dlls.

Intended audience

Project partners

License

Speaker recognition engine:

KNOWLEDGE owns the IPR for speaker recognition engine software. Licensing negotiable with separate agreement.

Speaker recognition server:

LGPL

Language

C#/C++

Java

Environment (set-up) info needed if you want to run this sw (service)

Hardware:

- PC computer with network connection

Software:

- Windows 2K
- .NET 2.0

Platform

.NET

Tools

-

Files

No source code available yet.

Documents

Deliverable D4.2

Tasks

For D4.3 KNOWLEDGE will develop the speaker recognition server (basic functionality).

Bugs

-

Patches

-

4.1.2 Explicit speech interaction**Provider****KNOW****Introduction**

Explicit speech interaction provides means for user-system communication using natural language dialogues. Two layers of abstraction can be distinguished:

- *Explicit speech interaction framework*, that includes general-purpose modules responsible for start-up, shutdown, initialization and communication tasks of speech interface as well as speech application execution environment, and,
- *Speech applications resources*, that are task-oriented collections of resources and scripts able to provide a complete interaction for fulfilling the specific tasks.

Development status

-

Intended audience

Project partners

License

It will be developed as open source software under LGPL license, except the speech application execution environment (recognition and TTS engines), which is commercial software covered by vendors' licenses.

Basic functionality, general-purpose parts will be provided as open source software libraries.

Language

C#/C++, GRXML, SSML

Environment (set-up) info needed if you want to run this sw (service)

Hardware:

- PC computer with network connection

Software:

- Windows 2000 SP4
- Speech recognition engine Scansoft OSR3
- TTS Scansoft RealSpeak

Platform

.NET

Tools

-

Files

No source code available.

Documents

Deliverable D4.2

Tasks

For D4.3 KNOWLEDGE will provide the *Explicit speech interaction framework*:
(basic functionality).

Bugs

-

Patches

-

4.1.3 Implicit speech input**Provider**

INRIA

Introduction

This functionality provides a generic framework for the application developer to guide and help him in the development of services that can compute and/or handle implicit speech inputs. This framework is based on a subscription mechanism, where implicit speech consumers subscribe into the implicit speech broker to be warned when implicit speech information of a given type occurs.

Development status

Version 0.1 is available. It consists of a preliminary version of the interfaces and access points definitions. The functionalities are not implemented yet.

Intended audience

Project partners.

License

LGPL

Language

JAVA

Environment (set-up) info needed if you want to run this sw (service)

Hardware: standard desktop or laptop PC

Software: real-time large vocabulary continuous speech recognizer

Platform

Native linux or cygwin.

J2EE version 1.4, Sun Java System Application Server

Tools

-

Files

Source code

Documents**Tasks**

- Complete the interfaces
- Implement the functionalities
- Write the documents
- Test, debug, validate

Bugs

No bugs reported for now

Patches

No patches created for now

4.1.4 Single Channel Speech Enhancement**Provider**

PAE

Introduction

The single channel speech enhancement software provides a stream of audio signals for other Amigo services, like speech recognition. State-of-the-art noise reduction algorithms combined with a real-time protocol server (RTP) and appropriate hardware are the key elements of the single-channel speech enhancement service.

Development status

A first software version is under development and will be available in the next month.

Intended audience

All interested Amigo consortium partners

License

This software is proprietary software by PAE. It will not be made open source since it is not middleware software. Special arrangements can be made with Amigo partners.

Language

C++ / C on Linux operating systems

Environment (set-up) info needed if you want to run this sw (service)

Hardware: single- or multi-channel soundcard; microphones; PC

Software: SuSe Linux 9.2; Jack Audio Connection Kit;

Platform

SuSe Linux operating system (9.2 or higher)

Tools

-

Files

One executable (asreng – main engine), modular based shared libraries and special configuration files.

Documents

Installation guide and usage guidelines will be made available.

Tasks

Reduction of computational demands, additional experiments and usage tests, low latency interface towards other Amigo software modules

Bugs

No known bugs.

Patches

Intermediate critical patches

4.1.5 Multi Channel Speech Enhancement**Provider**

PAE

Introduction

The multi channel speech enhancement software provides a single channel stream of audio signals for other Amigo services. State-of-the-art noise reduction and beamforming algorithms combined with a real-time protocol server (RTP) and appropriate hardware are the key elements of the multi-channel speech enhancement service.

Development status

A first software version is under development and will be available in the next month.

Intended audience

All interested Amigo consortium partners

License

This software is proprietary software by PAE. It will not be made open source since it is not middleware software. Special arrangements can be made with Amigo partners.

Language

C++ / C on Linux operating systems

Environment (set-up) info needed if you want to run this sw (service)

Hardware: multi-channel soundcard; far field microphones; PC

Software: SuSe Linux 9.2; Jack Audio Connection Kit;

Platform

SuSe Linux operating system (9.2 or higher)

Tools

-

Files

One executable (asreng – main engine), modular based shared libraries and special configuration files.

Documents

Installation guide and usage guidelines will be made available.

Tasks

Reduction of computational demands, additional experiments and usage tests, low latency interface towards other Amigo software modules

Bugs

No known bugs.

Patches

Intermediate critical patches

4.1.6 Acoustic Scene Analysis Service**Provider**

PAE

Introduction

The acoustic scene analysis service intends to provide context information gained from audio signals. For the first phase of the Amigo project a voice activity detection and a speaker change detection a planed.

Development status

The planning phase for the software has been finished and the implementation will start soon.

Intended audience

All interested Amigo consortium partners

License

This software is proprietary software by PAE. It will not be made open source since it is not middleware software. Special arrangements can be made with Amigo partners.

Language

C++ / C on Linux operating systems

Environment (set-up) info needed if you want to run this sw (service)

Hardware: single- or multi-channel soundcard; (far field) microphone(s)

Software: SuSe Linux 9.2; Jack Audio Connection Kit; Single- or multi-channel speech enhancement software

Platform

SuSe Linux operating system (9.2 or higher)

Tools

-

Files

One executable (asreng – main engine), modular based shared libraries and special configuration files.

Documents

Installation guide and usage guidelines will be made available.

Tasks

Implementation of modules and experiments;

Bugs

No known bugs.

Patches

Intermediate critical patches after first software release.

4.2 Gesture Service**4.2.1 2D Gesture Service**

Provider**LORIA-INRIA****Introduction**

The 2D tactile interface allows to select objects in a scene sorting them according to their saliency. The scene is compound of a background image and foreground artefacts. These artefacts could be either basic geometric shapes (circles, triangles, squares...) or complex polygonal figures, each of them associated in the scene with a type and a unique identifier. The scene is encoded in MMIL format (MultiMedia Interaction Language). The selection of the objects is done with a 2D gesture using a special pen on a tactile screen (TabletPC). The list of the selected objects (identifier and type) is outputed in the MMIL format as a visual-tactile event. The module can also receive MMIL events (refresh, clean, highlight...) in a Soapmmil environment (jakarta-tomcat webservices). The Soapmmil environment can eventually be bypassed for standalone use.

- **Initialization.** The visual scene has to be delivered as a MMIL event (XML). The scene consists in a background image and a list of points or polygones, associated with a specific id.
- **Inputs.** Mouse dragging events + scene update events (e.g. refresh, clean, polygone highlighting).
- **Internal process :** mouse dragging events are captured, tracking the path of the tactile pen over the scene. The intersection of the convex envelop of this path and the underlying polygones gives the salience measure of the selection of the corresponding objects.
- **Outputs :** a MMIL event containing the list of id and salience value, corresponding to the polygones covered by the tactile pen.

Development status

1.0

Intended audience

Project Partners

License

LGPL

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

Hardware:

- TabletPC (Toshiba Portégé 3500)

Core software:

- jdk 1.4.1
- ant-1.5.3

Additional software for SOAPMMIL protocol (if needed):

- jakarta-tomcat-4.1.29
- soapmmil-0.15

Platform

Window XP (Tablet PC edition)

Tools

-

Files

Source code available in Java, build with ant

Documents

Technical report in french

Tasks

Development of this service as a JavaBean component for a more suitable integration.

Bugs

-

Patches

-

4.2.2 3D Gesture Service

Provider

VTT

Introduction

3D gesture service enables recognition of free form 3D gestures based on wireless handheld sensor device. The sensors on the device can be used to detect discrete pre-trained gestures based on acceleration patterns during gestures. The system has optional support for physical selection of objects by means of infrared-based pointing and it will be realised if there is use

for it in the final demonstrations. Pointing enables the user to first select a known target device from the environment by pointing, and then control it with a set of gestures. Gesture movement data is wirelessly transmitted to a computer where the 3D Gesture Service takes care of data processing and recognition. The 3D Gesture Service provides methods to train and recognize personal hand gestures. The 3D Gesture Service provides the gesture, tilting and pointing events to Multimodal Service, which takes care of the integration and filtering of different modality inputs. The 3D Gesture Service consists of sensor and radio hardware and PC software for recognising the gestures.

Development status

First version 1.0 of hardware and software is ready and working. Both hardware and software will be developed before final demonstration. Hardware mechanics and design will be modified to suit the demonstration purposes.

Intended audience

Project partners

License

VTT owns the IPR for software and hardware. Licensing negotiable with separate agreement.

Language

LabView G/C#

Environment (set-up) info needed if you want to run this sw (service)

Hardware:

- VTT SoapBox wireless sensor device with receiver
- SoapBox based Ir tags (optional)
- PC computer with serial port and network connection

Software:

- Windows XP
- LabView runtime engine (supplied with 3D Gesture Service)
- 3D Gesture Service software

Platform

Windows, LabView runtime engine

Tools

LabView runtime engine (supplied with 3D Gesture Service)

Files

No source code available. 3D Gesture Service is available as executable simulator (3DGestSrv.exe). Gesture events are sent from graphical UI (3DGestSrvSim.exe).

Documents

3D Gesture Service system architecture is described on D4.2. Separate software and hardware installation guide will be provided with the Service.

Tasks

For 4.3 VTT will develop simulation software for 3D Gesture Service.

Both hardware and software will be developed before final demonstration. Hardware mechanics and design will be modified to suit the demonstration purposes.

Bugs

None so far

Patches

None so far

4.3 GUI Service

Provider

IMS

Introduction

In Amigo a so called "GUI-service" will be implemented to allow GUI-applications to connect to the AMIGO world. These GUIs could be developed by different manufactures and can be connected over the defined interfaces.

The GUI-Service allows the personalization of the navigation idea (the sorting of menu points into hierarchy) and the personalization of widgets to reach a personal look&feel which can be carried into different Amigo environments. This offers the possibility to operate an unknown environment efficiently without adapting to the logic of each environment, because the user interface is presented in a familiar way according to the user's way of thinking. This should help the user to operate an unknown environment like his own one he has installed at home. Of course this personalisation has to be supported by the used GUI-Application of the chosen manufacture as well.

Development status

Under development

Intended audience

Project Partners

License

LGPL

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

GUI-Service will employ/support:

- Hardware: PC/Laptop
- OS: Any Java-enabled OS (JRE >=1.5), OSGi
- Software: Jena2, JDom

Platform

JVM, OSGi

Tools

-

Files

-

Documents

Not yet available

Tasks

-

Bugs

None so far

Patches

None so far

4.4 Multimodal Dialog Manager

Provider

INRIA, KNOW, VTT

Introduction

A Multimodal Dialogue Manager manages the flow of dialogs that an application requests.

Development status

Preliminary version is under development.

Intended audience

Project partners.

License

The software itself is under LGPL license, but it might make use of proprietary binaries/libraries for which no source code is provided.

Language

C#/C++

Java

Environment (set-up) info needed if you want to run this sw (service)

Platform

-

Tools

-

Files

Source code

Documents

D4.2

Tasks

-

Bugs

No bugs reported for now

Patches

No patches created for now

4.5 Multimodal Fusion

Provider

INRIA-CNRS

Introduction

The Multimodal Fusion Module (MFM) deals with related information from the user, occurring in short periods of time and that can be mutually completed. The role of the MFM is to merge and cross-complete information from several input modules. The MFM then outputs the merged information with as primary target the Multimodal Dialog Manager.

- Inputs: several MMIL-encoded message. MMIL format is used for each of the input modalities, a conversion module may be required for each modality that is not handled by a MMIL-compliant module.
- Output: a unique MMIL message for any set of merged input modalities.
- Parameters:
 - An ontology of the task entities and events/commands
 - A modality-dependant tuning about time windows is possible, which can be turned if necessary, into a context-dependant tuning, while it is not the case currently.

Currently handled modalities: Natural Language, 2D/3D Gestures

- Semantic chunks (from natural language analysis) with a primary type (e.g. from the noun for participants or verbs for events: the TV, stop) and/or a set of atomic characteristics (e.g. from adjectives or verbs). Complex characteristics related to events or related to states with one relations such as “the pen I left on the chair yesterday” or “the pen near the box” are not handled in the current state of development.
- Weighted sets of conceptually-marked objects (from 2D or 3D gesture modules)
- Concepts related to the task (from 2D/3D gesture modules, menu-oriented UI)

Following examples are all eventually merged into a simple command Pause(HiFi):

- User gesture describing a pause THEN user gesture pointing toward the HiFi.
- User utterance “The first-floor stereo” WHILE user gesture describing a pause .
- User utterance “Stop the music”.

Development status

Most parts of the module are finished, some are to be done on MMIL interaction, time-window tuning, ontology proximity checking.

Intended audience

Project partners at first (mainly the Multimodal Dialogue Manager and the user input modules) and then later also other developers of custom AMIGO services and applications, for ontology-related interaction and context-dependant tuning.

License

LPGL

Language

Java

Environment (set-up) info needed if you want to run this sw (service)

Hardware: Any host providing a Java virtual machine

Software: JRE 1.5, JWSDP 1.6, Jena (depends on the Ontology service)

Tools:

Java Compiler, Ant

Files

Software code is partly available on repository.

Documents

Design documentation is provided in D4.2.

Tasks

The implementation of the first version has just started.

Bugs

The implementation of the first version has just started.

Patches

The implementation of the first version has just started.

4.6 Multidevice Service

Provider

IPSI, FT

Introduction

With the Multi Device Interface Service, we attempt to let the ambient intelligence environment assist in finding optimal compositions of interaction devices. By specifically addressing the uniqueness of different devices, a dynamic, reconfigurable, and context-aware composition of interaction devices can be achieved.

The service assists in finding the optimal set of devices and interfaces by matching the interaction characteristics of each available device with the properties of each so called Interaction Demand an application might formulate. Such an Interaction Demand might be to present the user either privately or publicly an incoming email, depending on who else is around. For instance, if only family members are around, the large TV screen might be used to display the greeting card of the grandmother, but when strangers are around, the mobile phone of the recipient might vibrate and the reception of the mail might take place on the PDA screen, which is harder to read, but obviously more private than the TV.

Development status

The implementation of the first version has just started.

Intended audience

Project partners at first and then later also other developers of custom AMIGO services and applications.

License

This software is proprietary software. It will not be made open source. Special arrangements can be made with Amigo partners.

Language

C++

Environment (set-up) info needed if you want to run this sw (service)

Hardware: Windows PC

Software: Windows XP, .NET 2.0

Platform

-

Tools

Visual C++ 2005

Files

Binary distribution as .NET assembly. Software code will not be available.

Documents

Design documentation is provided in D4.2.

Tasks

The implementation of the first version has just started.

Bugs

The implementation of the first version has just started.

Patches

The implementation of the first version has just started.

5 Privacy and Personal Security

5.1 Perceived Privacy rules

Provider

Philips

Introduction

Perceived privacy and security are difficult to encase in a model, but this task try to define common concepts, design rules and potentially some software to facilitate these guidelines. At the moment of this writing, it is not yet known what kind of software will result and therefore not all attributes can be presented here.

Development status

Specification phase

Indented audience

Project partners

License

Any resulting software will be under BSD license.

Language

Not yet known

Environment (set-up) info needed if you want to run this sw (service)

Not yet known

Platform

Not yet known

Tools

-

Files

-

Documents

No documentation yet

Tasks

-

Bugs

-

Patches

-