A Fusion Framework for Multimodal Interactive Applications

Hildeberto Mendonça
Université catholique de Louvain
Place du Levant 2
Louvain-La-Neuve, Belgium
hildeberto.mendonca@uclouvain.be

Jean-Yves Lionel Lawson
Université catholique de Louvain
Place du Levant 2
Louvain-La-Neuve, Belgium
jean-yves.lawson@uclouvain.be

Olga Vybornova
Université catholique de Louvain
Place du Levant 2
Louvain-La-Neuve, Belgium
olga.vybornova@uclouvain.be

Benoit Macq
Université catholique de Louvain
Place du Levant 2
Louvain-La-Neuve, Belgium
benoit.macq@uclouvain.be

Jean Vanderdonckt
Université catholique de Louvain
Place du Levant 2
Louvain-La-Neuve, Belgium
jean.vanderdonckt@uclouvain.be

ABSTRACT

This research aims to propose a multi-modal fusion framework for high-level data fusion between two or more modalities. It takes as input low level features extracted from different system devices, analyses and identifies intrinsic meanings in these data. Extracted meanings are mutually compared to identify complementarities, ambiguities and inconsistencies to better understand the user intention when interacting with the system. The whole fusion life cycle will be described and evaluated in an office environment scenario, where two co-workers interact by voice and movements, which might show their intentions. The fusion in this case is focusing on combining modalities for capturing a context to enhance the user experience.

Categories and Subject Descriptors

B.4.2 [**Input/Output and Data Communications**]: Interconnections—*Interfaces, Parallel I/O*; D.2.13 [**Software Engineering**]: Reusable Software—*Reuse Models*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human Factors, Human Information Processing*

General Terms

Management, Experimentation, Human Factors

Keywords

Multi-modal fusion, speech recognition, context-sensitive interaction.

ICMI-MLMI'09, November 2–4, 2009, Cambridge, MA, USA.

1. INTRODUCTION

Modality fusion fulfils an important role on the development of applications with support for multiple modalities. It integrates data and meanings coming from different sources. These applications are progressively evolving towards more robust semantic interpretation of the user's intentions and a fusion mechanism is important to combine data streams by reducing uncertainty. To formalize the coordination between modalities involved in the same multi-modal interface, it is required to extract relevant features from signal representations and thereafter to proceed to high level fusion.

The effort on the implementation of fusion mechanisms has shown that solutions are dependent on the context [19] and other variables such as available devices, granularity of collected data, available analytical algorithms and a large heterogeneity of technologies are difficult to integrate. These issues lead people to implement fusion mechanisms focused on specific case studies. This focus imposes researchers to rewrite or duplicate a solution for each new case study, which causes waste of time and resources, rewriting requirements that are common between them but not reusable.

To contribute with the consolidation and unification of multi-modal fusion approaches, this work proposes a framework to provide what is common in most fusion implementations and support extensions to allow developers and researchers to attach their algorithms. The framework will manage important requirements, such as: a) parallel processing of data coming from different devices; b) synchronization of data on time; c) formalization of a well-defined process, composed of activities executed sequentially and continually; and d) an agent-based layer responding to events of the process to perform different kinds of fusion.

The framework is not a fusion mechanism in itself, but it is the basis to instantiate fusion mechanisms. Developers can use it to plug their existing tools, devices and fusion techniques. The framework manages the execution and communication between all attached pieces, avoiding this basic structure to be repeatedly implemented. We will present common requirements of existing fusion mechanisms in sec-

tion 2, describe the architecture of the framework to support these requirements in section 3, describe the case study used to validate the framework in section 4, present the results obtained so far in section 5 and finally conclude the work discussing results and future works in section 6.

2. FUSION ESSENTIAL REQUIREMENTS

Many fusion engines have been developed so far with different approaches to allow data fusion, but they have specific purposes, solving particular scenarios of fusion [2] [10] [15]. When a new scenario is identified, a common practice is to implement the fusion from scratch due to the lack of tools or frameworks providing the basis for multiple implementations. We believe that the field can be substantially improved if a general support for fusion is provided, taking into consideration common features of existing fusion mechanisms. Exploring existing contributions, we have identified the following basic requirements:

- *Synchronization*: Since we have at least two modalities to justify fusion and they are both active during the user interaction, the synchronization in time is an important variable to analyse the correlation between events coming from each modality. This is based on the premise that if two events have approximate time stamps then they have a high probability to be correlated.
- *Cognitive Algorithms*: input modalities usually produce a high volume of data to be analysed. Efficient data analyses are provided by expert algorithms, using statistical models, first order logic, fuzzy logic, production rules and other approaches.
- *Context Representation*: It is important to know the context of the data to be analysed in order to improve efficiency. The context [3] is a complete description of the environment, where the interaction occurs, the user, who interacts with the system, and platform, which supports the multi-modal application and its fusion properties.
- *Visualization*: large volume of data under analysis must be appropriately visualized by specialists, who will follow the recognition, identify problems or non-recognized data. Visualization is a sort of high level test and demonstration.
- *Simulation*: Sometimes it is difficult repeat an experiment many times to validate the fusion mechanism. Simulation is important to perform tests during the development or demonstration of the fusion mechanism, without increasing time and resources during the process.

To provide a general support for fusion it is necessary to consider more than the features of existing fusion mechanisms, but also a model to support the variety of implementations and platforms in use. For instance, the literature presents Neural networks [28] and Bayesian networks [24] as distinct approaches to fuse data, but considering a general framework, both approaches should be supported and even complementary when necessary. This particularity shows the need for requirements not considered in fusion mechanisms so far, which are:

- *Component-based*: because of the complexity of data fusion, mechanisms are usually composed of several implementations. A component-based architecture allows high cohesion and low coupling to provide extensibility and reuse.
- *Multi-platform*: there are useful implementations available for fusion in different programming languages and platforms. The possibility to reuse what is already available instead of re-implementing in the chosen platform, might allow fast development and better results.
- *Scalability*: a fusion mechanism could be heavy and time consuming due to the volume of data to be analysed. A multi-thread architecture is needed to allow fully use of the computational resources.
- *Distribution*: sometimes a unique machine is not powerful enough to produce efficient results and using multiple machines can fulfil the demanded computational power.

We have developed a framework called Meanings4Fusion (M4F) [17] to fulfil these requirements. It is currently in its alpha version and it was validated in the case study described in section 4. M4F was developed in Java on top of the OpenInterface (OI) Platform [14], further described in section 3. The figure 2 shows a print screen of the framework's user interface.

3. FRAMEWORK DESIGN

The design of the framework is based on the essential requirements discussed in the previous section. It is organized in uncoupled and complementary layers, which are:

1. *modality processing pipeline* (MPP): a set of components with well delimited roles that transforms data by reducing its granularity from a signal level to a semantic level; and
2. *autonomous fusion agents*: agents activated during the execution of the modality processing pipeline to perform modality fusion techniques on the data under transformation.
3. *user interaction*: user interface to allow configuration, execution and visualization of the fusion process.

The modality processing pipeline is a bus of data coming from each modality. The autonomous fusion agents constantly analyse these data, comparing with existing data and performing fusion. These layers are described in more details hereafter.

3.1 Modality Processing Pipeline

The modality processing pipeline is composed of 4 stages, which aim to reduce the granularity of data until a level of natural understanding (data readable by human beings). The process starts when data representing signals are received by the framework through TCP/IP connections. This protocol was chosen because of its reliability, which is more important when analyses are necessary, and to allow distribution. The TCP/IP connections are managed by the Grizzly framework [12] to allow scalability. The result of the recognition is then segmented to determine the useful

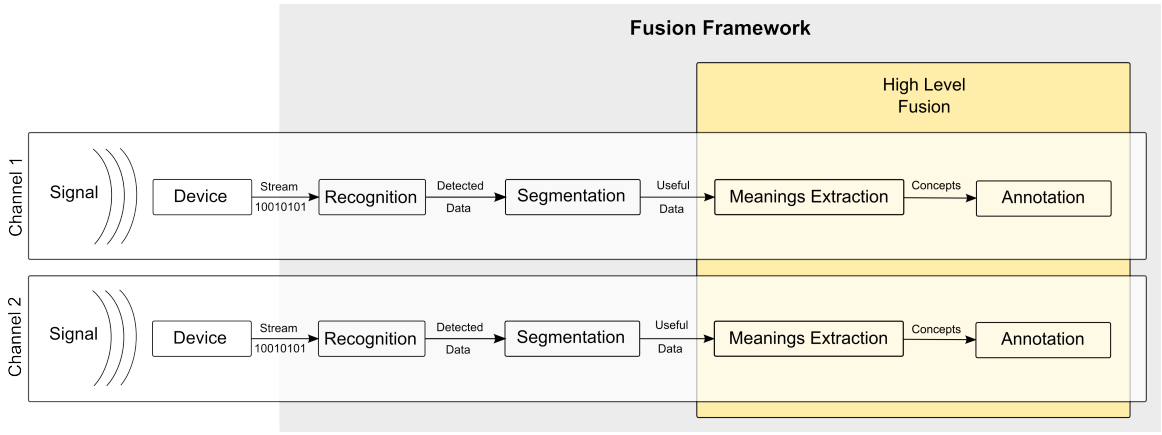

Figure 1: The fusion framework life cycle.

part of the content. This part is once again analysed to identify possible meanings and these detected meanings are finally annotated according to the domain of the context under experimentation.

We are using the OI platform to create the MPP. This platform is composed of the OI Kernel, which is a generic runtime for integrating heterogeneous code by means of non-intrusive techniques, and the SKEMMI, which is a tool used to manage and compose components on top of the OI Kernel [13]. This tool is used in this project to create the MPP as represented in figure 3, where each component has its own representation and they are linked through their inputs and outputs.

3.1.1 Recognition

The developer implements a recognition component to manage detection devices, such as cameras, microphones, and accelerometers. The recognition phase is focused on the analysis of signal features to identify useful content. For instance, considering that a camera is used to track people in a room, the recognition component should return the region where a person was located in the scene. The same image can provide the relative position of a person considering other objects in the scene.

3.1.2 Segmentation

The segmentation component receives recognized data and creates segments to delimit the useful data from the original content. From the previous example, a segment is all the sequence of regions where the person was moving. Other segments delimit the objects identified in the scene. The type of segment depends on the type of content. A temporal segment can, for instance, delimit all instants where the microphone detected people's conversation, but it rejects parts of the audio where noise was detected. A spatio-temporal segment can specify a region in a frame and associate a timestamp for each useful frame. A string delimits which part of a giving text is actually part of the domain. Table 1 lists the supported segments and the kind of content to which they can be applied.

3.1.3 Meanings Extraction

What is recognized and segmented is not actually endowed with meaning. A second round of analysis is needed to identify the semantics of the useful content. This is the role of the meanings extraction component. Here, the segmentation component offered an important support by reducing the scope of the analysis to the delimited content, improving the performance of the meanings extraction. An example of meanings extraction is the semantic analysis of a sentence, recognized using a speech recognition tool. It could also be the meaning of the relative position of a person to an object in the scene. The validation of extracted meanings is made by searching the terms found in the knowledge base, which describes the context in the format of an ontology.

Table 1: Types of Supported Segments

Type	Content
Spatial	video frame, picture
Temporal	audio, video, accelerometer, stream
Spatio-Temporal	video frames, 3D sequence
String	Text, stream, accelerometer

3.1.4 Annotation

Lastly, extracted meanings are instantiated in the knowledge base according to the concepts of the domain described in an ontology. Because the adopted ontology model is based on RDF (Resource Description Framework), meanings are stored in a triple format, composed of a subject, an object, and a predicate linking them [26]. Some examples of triple: 1) "<person> -is close to- <the computer>", 2) "<gesture> -is- <circular>", 3) "<John> -wants to call- <Nick>".

Figure 1 depicts a modality processing pipeline for each modality. The framework is responsible for creating a new channel each time a new device is connected. The data received through the channel is processed by the pipeline. It is important to notice that no fusion is done in this layer, but all data is prepared to be fused by autonomous fusion agents described as follows.

The annotation stage was inspired from the work of Jean-Claude Martin, who adopts Anvil for video annotation in a multi-modal framework [16]. We use an annotation tool developed in the context of the IRMA Project [4] that can annotate, not only videos, but also images, sounds and immutable texts.

3.2 Autonomous Fusion Agents

The MPP should be processed without interruptions of the fusion implementation. It could be time consuming, interrupting the modality processing and, consequently, losing important data coming from devices. A standard implementation of fusion in separate threads could be done, but it represents more effort to detect events from the MPP. Therefore, a multi-agent architecture is supported to fulfil this need.

This layer assumes the architecture of a multi-agent system, which is a system that includes multiple autonomous entities with diverging interests [22], called intelligent agents [21]. The intelligent agent is capable to perceive the environment and act according to its conclusions. Its capability to perceive will monitor the meanings extracted from segments. Since the environment is described in the ontology, the agent uses it as its knowledge base. Its capability to act will perform reasoning with extracted meanings from all connected modalities to produce a conclusion about the user intention. Previous conclusions about the user intention are reused by the agent in new reasoning cycles to improve the latest conclusion.

We have integrated the SOAR [18] cognitive tool in the fusion framework to fulfil the reasoning needs of this layer. The tool already provides an implementation for multiple agents execution, restricting our responsibility to write production rules.

3.3 User Interaction

To simplify the use of the framework and provide practical information about the modality processing and fusion results, a user interface was developed for this purpose. The figure 2 depicts a representative screen-shot of the main window. It shows horizontal bars corresponding to each channel of the framework in a time line, which are divided in grey bars representing identified segments. The content of the segment and its associated meanings are shown in a pop-up window when the user clicks on the grey bar. These bars show, in real time, what happens during the MPP processing.

This user interface also helps the user to configure and perform the modality fusion. In configuration time, the user composes a pipeline in the OI platform (figure 3) to deal with each modality and associate it to a kind of channel, such as video analyses. A channel is instantiated for each input modality, mapping and connecting computers and devices. Once configured, the framework can be started, activating all channels at the same time. Each channel opens a port and waits for data from different sources. The user interface provides all support to start, pause and stop this process.

4. CASE STUDY

This case study considers the use of the framework to instantiate a fusion mechanism to be applied in an office environment scenario. The purpose of this application is to assist workers in their daily tasks in the office. The application will analyse their behaviour, perceive their intentions and recommend them to perform the task. In order to develop and validate the whole experiment, we defined five different scenarios where two people interact mutually in a room, talking in a natural way and behaving without restrictions. Each scenario tries to explore specific combinations of speech and behaviour to increase the robustness of the

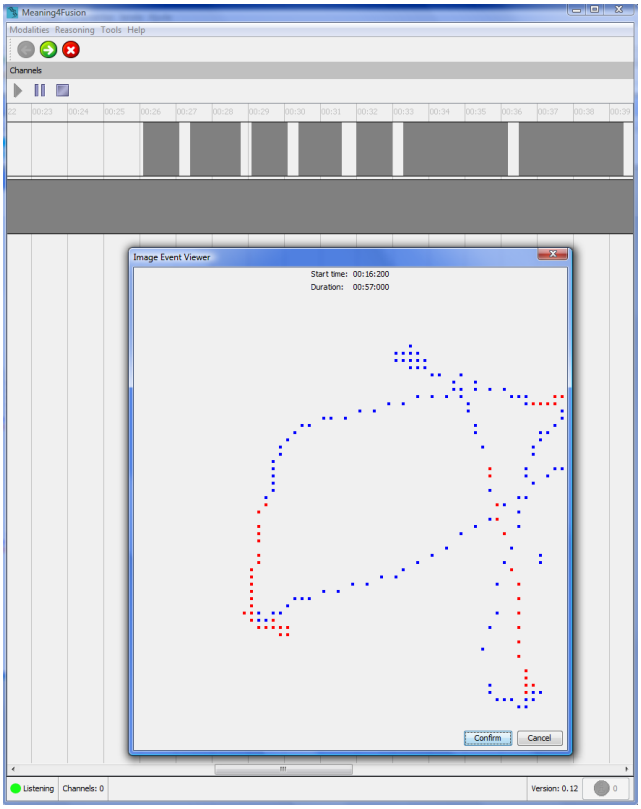

Figure 2: The Meanings4Fusion framework

system. This system is able to track people, analyse their behaviour, movements, speech, and makes decisions about how to prompt necessary information when required or provide any other assistance.

In order to efficiently analyse behaviour, the system has to correctly process, interpret and create joint meaning of the data coming from speech analysis and video scene analysis. We consider that human behaviour is goal-oriented, so our main aim is to recognize users' plans and produce an advice about how to better perform the task.

The system manages the data streams arriving from two sources: video scene and speech. In particular, we show a technique distinguishing between the data from different modalities that should be fused and the data that should not be fused, but analysed separately.

The fusion mechanism [25], instantiated on the fusion framework, employs various components in the MPP layer according to Table 2. In the autonomous agents layer we have considered a decision-making framework, SOAR [18], to perform the high level fusion and Protégé [23] to manage the knowledge base.

The process starts when an audio or a video signal is detected by the first time. There is no restriction about which signal should start first because all modalities can be processed in parallel and independently.

4.1 Speech Analysis

When an audio stream is received, the speech recognition component processes it, generating a string of what was said. The same signal is sent to the speaker identification component, which will associate what was said with who said that.

Figure 3: The OI pipeline of the framework.

Table 2: Fusion Mechanism Components

Component	Implementation	MPP Part
Speech recognition	Sphinx-4 [27]	Recognition
Speaker identif.	Matlab	Recognition
Syntactic parsing	C&C Parser [7]	Segmentation
Semantic analysis	C&C Boxer [7]	Meanings extr.
Video analysis	Open CV [6]	Recognition
Behaviour Analysis	Own	Meanings extr.

The string is sent to the syntactic parsing component to identify the syntax of each word, which is important for the natural language semantic analysis component, responsible for the identification of the subject, the agent, the predicate, the object of interest and other elements. From the semantic analysis, it is possible to extract semantic structures very similar to the structure of the knowledge base, represented by ontology. If we find the identified semantic in the ontology then it means that the sentence is valid inside the context and can be useful to fuse with other meanings coming from other modalities.

4.1.1 *Speech Data*

The speech was recorded by two non-native English speaker subjects, one 23-year Chinese male and one 32-year Swedish male. The data was recorded using 16kHz, 16 bit audio. The 5 scenarios consisted of a total of 72 sentences and 148 seconds. For the development of speaker identification, we used ten phonetically rich sentences for training, and for parameter tuning we used another ten phonetically rich sentences and ten words of different length. To illustrate the recorded data, we put below the dialogue of the second scenario, where the conversation is short but also complex to detect the user intention.

1) Beto: Hi Ronald! How is life going?
2) Ronald: I am fine.
3) Ronald: I want to call Nick.
4) Beto: What for?
5) Ronald: He mentioned that he attended a wine tasting course.
6) Beto: It sounds interesting, I like wine.
7) Ronald: Actually I plan to join the next class. He also mentioned a book about French wines, but I cannot recall the name of the author.
8) Beto: Why don't you send an e-mail to Nick?
9) Ronald: Maybe I can find a book about it in the library.
10) Beto: Yes, you are right.
11) Beto: Did you find it?
12) Ronald: Yes, I did.

4.1.2 *Speech Recognition*

For speech recognition we used Sphinx 4, which is an open source, Java-based speech recognizer [27]. For acoustic modelling, we used the 8 Gaussian triphone models, trained on the Wall Street Journal Corpus, which are supplied along with Sphinx. Since we wanted to allow the system to monitor a discussion between two or more people, we want to have a large vocabulary language model. For this purpose, 3-grams with a maximum of around 5000 words were trained using the orthographic transcriptions from the Wall Street Journal Corpus. The 5000 words were selected as the most common ones plus the ones that are present in the scenarios.

4.1.3 *Speaker Identification*

Speaker Identification is the task to determine who is speaking. For the application described in this report, a standard speaker identification system was considered. It is based on Mel-Frequency Cepstral Coefficients (MFCCs) and Gaussian Mixture Models (GMMs), as in [20]. We used 28 log-Mel filters between 300 and 8000 Hz, cosine projected to 24 dimensions, where the first 12 with their delta were used. A simple data driven procedure for speech detection was tried: The MFCCs were clustered using the k-means algorithm with euclidean distance for two clusters. Then the cluster, which had the highest energy was marked as "speech" and the other one as "non-speech". Then these two clusters were used for frame-based segmentation. Visual inspection showed that the approach seemed reasonable. No channel compensation was used. This system was implemented in Matlab.

4.1.4 *Syntactic and Semantic Analysis*

The next stage after speech recognition is syntactic and semantic analysis of the discourse. For our purposes we use the CCG (Combinatory Categorial Grammar) parser, Release 0.96, developed by S. Clark and J. Curran [7]. The grammar used by the parser is taken from CCGbank developed by J. Hockenmaier and M. Steedman [11].

CCGbank is a treebank containing phrase-structure trees in the Penn Treebank (WSJ texts) converted into CCG derivations. It allows easy recovery of long-range dependencies, provides a transparent interface between surface syntax and underlying semantic representation, including predicate-argument structure. The grammar is based on "real" texts, and that is why it has wide-coverage, thus making parsing efficient and robust. The CCG parser has Boxer, as add-on to generate semantic representations - Discourse Representation Structures (DRSs), the box representations of Discourse Representation Theory (DRT) [1]. DRSs consist of a set of discourse referents (representatives of objects introduced in the discourse) and a set of conditions for these referents (properties of the objects).

4.2 Video Scene Analysis

When a video stream is detected, the image is processed by the video analysis component. This component analyses some image features to calculate the position of each person on the horizontal plan of the scene, their movements' direction and it identifies who is each person according to

a predefined profile. It is also important to identify people through this modality because we have to know the position of who is talking in order to associate the user intention with his/her actions.

The next step is to analyse the human behaviour, comparing the movements of the user with a set of rules. The behaviour is relative to fixed objects in the scene, which are defined in the context domain and are directly associated with the aid to be given by the system. The rules define the boundaries of what is near or far from a certain object. Then the result of the rule processing at this stage is: "<person> -is near- <the telephone>", "<person> -is far from- <the computer>" or "<person> -is moving to- <the library>". This result is produced for each person in each frame of the video. Individually, these results are not significant enough for fusion. We have to analyse the movements in many frames in order to have final conclusions. For instance: if in the last 80 frames, the rule engine produced "<person> -is moving to- <the library>" then we can conclude that there is a real intention to reach the library, considering some variables of the environment, such as area of the room.

The video sequences were recorded using a distributed 8-camera voxelised visual hull [8]. The description of the environment is obtained by processing each image of the videos using computer vision algorithms. In these video sequences, there are 3 types of fixed objects (a telephone, some books and a computer) located in different positions inside the scene, there are also two persons that are moving and interacting with these fixed objects. In order to have a good description of the environment for each one of these scenarios, it is necessary to extract the information of the position of each fixed object, the position of each person at any moment and also the motion direction of each person.

Extracting all this information from the video sequences are common issues of the Computer Vision field. These issues are mainly: a) *Object detection* - to find the position of each fixed object; b) *People detection and tracking* - to find the position of each person; and c) *Motion analysis* - to find the motion direction of each person.

The computer vision system to extract this information was implemented in C++ using OpenCV library [6] developed by Intel. The objects (the telephone, the books and the computer) in these video sequences are always fixed, also they do not change size or rotate because the camera's viewpoint is always the same. Hence, the detection of these objects can be easily done by using a template matching algorithm [5]. These algorithms compare a template with a region of an image in order to determinate a similarity measure, wherein the similarity measure is determined using a statistical measure.

To obtain these templates, a sample picture of each object is captured from any image of the video sequence. The OpenCV operator *cvMatchTemplate* was used, this operator returns the probable positions in the image where the template can be located. This way, the most probable position corresponds to the area where the object is detected. The similarity measure that provides better results for this problem was the correlation coefficient normalized. Because each object never changes position, this template matching step is only done once in the first captured image of the video sequence. Then, these positions are used for all the images in the video sequence.

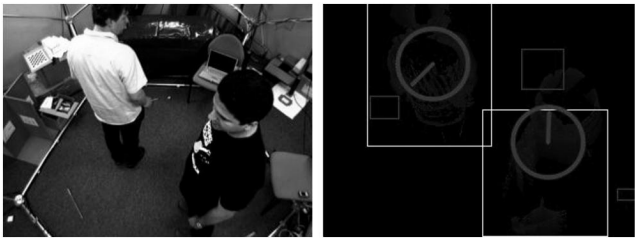

Figure 4: Results of the computer vision system.

In the video sequence of this project, two people are talking with each other and also moving randomly inside the scenario in order to interact with these objects. The issues here are mainly: a) the shape and size of each person can change over time because the people can move far or close to the viewpoint in different parts of the scenario; b) the people can approach too much each other in the video, making the identification of each one more difficult; c) one person can be partially occluded by the other person; d) some body parts of each person can be outside the scenario because of the viewpoint of the camera; and d) each person moves in a random way.

To solve these issues, a colour-based tracking was used. The colour of the clothes of each person was used, assuming that both people in the video will have different colour clothes. However, a background subtraction and a blob detection technique are needed in order to make the people detection robust for the cluttered environment and discriminate noise. Motion Templates algorithms were used based on papers by Davis and Bobick [9] to find the motion direction of each person. These algorithms are very fast and robust. The implementation was done using OpenCV Motion Template functions. These functions can determine where a motion occurred, how it occurred, and in which direction it occurred. To calculate the motion direction of each person, the silhouettes (obtained from background subtraction as described in section 3.2) are updated in time using *cvUpdateMotionHistory* operator, after the motion gradient is calculated using the information of the temporal silhouettes (applying *cvCalcMotionGradient*). Finally, connected regions of Motion History pixels are found using OpenCv operator *cvSegmentMotion*. With this result, we have region of motions with their gradient directions in the foreground image.

For the human behaviour analysis, we need to know whether every object is near or far from each person and the object toward which each person moves. In order to find this information we apply some rules using SOAR and define thresholds using the result of the computer vision system (position of each fixed object and each person and motion direction of the people).

4.3 Ontology Design and Reasoning

In order to describe the context and allow reasoning, an ontology with comprehensive modelling pattern of multimodal actions and prior knowledge about the objects and users is necessary. We used open source tool Protégé [23] to create the ontology in classes, properties and individuals.

We use SOAR [18] cognitive architecture in order to create and apply rules to query semantically the modality triples in the ontology. The ontology is re-implemented for adapt-

ing the specific working memory structure in SOAR environment. The elements class, property and individual in Protégé corresponds to identifier, attribute and constant in SOAR, respectively. In order to have this knowledge base for reasoning, the ontology is created as states when agents are initialized. After receiving the input from external application, an operator is proposed and applied to map the input triple to the knowledge base then give out the result to the output link.

5. RESULTS

In our experimental work, we used scenarios only with natural human language. We did not work with isolated words, commands, restricted language or something similar. The goal was to experiment with normal complete utterances expressed in a natural way.

In this experiment, to make multi-modal data fusion means to interpret human behaviour, to identify the users' plan, infer their intentions. Having understood what the people in the scene want to do, the system makes the decision about how to assist them in the given case. Looking at our challenging example scenario, transcribed in section 4.1.1, we can see that there are 4 points in this dialogue where the speakers express their plan to do something. In (3) Beto wants to call Nick, in (7) Ronald plans to attend the next class, in (8) there is a possible plan to send an e-mail and then this possible path of decision is changed for another route, in (9) there is an intention to find a book in the library. How shall the system realize which plans to take into account and which ones not? When to react and when not? This is why we employ multi-modal information. When the plan is identified from the user's words, we look at the other modality data to see if the person is going to "confirm" his/her words with the corresponding actions or not. In (3), (7) and (8) the people expressing their spoken intentions were still standing, just continuing the talk. And only in the (9) Ronald moved to the bookshelves. That is why only in this last case of plan expression the system reacted and prompted where to find the desired book. By the way, in the phrase "Maybe I can find a book about it in the library" we have to resolve ambiguity between the library in the room and a library on the web. We do that using information from the other modality. We look if the person is moving to the books in the room or if he is moving to the computer.

When identifying the person's plan from speech, we basically rely on the linguistic semantic analysis as described in section 4.1.4, but we certainly take advantage of the obvious lexical signs of plan and intentions expression. For example, such verbs and phrases like "want", "wish", "plan", "going to", etc. (we defined 19 expressions like this in total) in a certain syntactic context and in the present or future tense clearly point at the person's intention to do something. And vice versa, negative forms of verbs like "I don't want", "I have no wish to...", "You don't want to..." as well as verbs in the past tense serve as stop-words, and signal that this plan should be discarded and not taken into account, because no system response is needed.

The fusion occurred when we identified a person moving to the library and we also detected the intention to find a book in a sentence like this: "I can find a book about it in the library". Therefore, we could conclude the person wants to find a certain book at the library and the computer could provide assistance for this person to find it, giving the exactly location of the book, such as the bookshelf and its relative position to other books.

The composition of many different tools and components was definitively a challenge, achievable due to the framework that has been developed on top of the OI platform. We used Java and C/C++ programs and OI allowed the communication between all of them without any particular change in the source code. The decision making part was strongly impacted by the poor results of the speech modality. However, we could provide two advices during the experiment: Nick's phone number and the location of the book about French wines, because in both cases, the user intention was well recognized with the speech recognition tool and user's movements were well tracked towards the telephone and the library.

We are using only open source software to compose each part of the fusion mechanism. Unfortunately, these tools follow the state of the art very slowly and we could not get results that are possible to expect from proprietary or inaccessible tool developed by companies and recognized research centres. One of the strongest impacts of this slow evolution was in the speech recognition modality, where Sphinx could not provide precise results and C&C Tools were inaccurate on semantic analysis when ignoring interrogative sentences and other punctuations. On the other hand, the computer vision modality provided precise results about people positioning and movement directions, due to the rich framework OpenCV, whose output is presented in Figure 4.

6. CONCLUSION

This instance of fusion mechanism contributed to validate the life cycle of the fusion framework. Each channel represents a modality. Two devices were connected to capture signals, which are processed in the recognition phase. For each detection, the framework creates a segment in the segmentation phase, delimiting useful data. Segments are processed and meanings are extracted from them. These meanings are annotated in the annotation phase. During the modality life cycle, an event-oriented architecture executes threads in parallel to perform high level fusion. Using the current proposal of the framework, we could instantiate a fusion mechanism to:

- manage spatial relationships based on the fixed objects in the room;
- make semantic fusion of events not coinciding in time;
- achieve good results in speaker identification;
- synchronization between image and speech identification;
- create an open framework to manage fusion between two or more modalities; and
- design the system so that each component can run in a separate machine due to the distribution mechanism interchanging data through a TCP/IP network.

However, we have more issues to solve in our future works. To name just a few, we should:

- implement an effective learning mechanism with long term memory to improve the recognition after several

executions, by automatically increasing the training dataset;

- perform efficient decision making, even from information fragments, which could be achieved by giving more focus on complementarities and disambiguation of modalities;
- perform 3D video analysis in virtual worlds using the same strategy, but analysing the behaviour of virtual avatars in several situations; and
- add other modalities, e.g. eye gaze tracking, to evaluate the behaviour of the mechanism and the scalability of the system.

Since the basis of the framework is consolidated, these future works are oriented to increase the robustness of the solution, expand the possibilities of experimentation, and gradually improve the overall performance and reliability. However, that basis is not fixed and can be improved following the evolution of the field of multimodal applications, which is broadly discussed nowadays.

7. REFERENCES

[1] V. Akman. From discourse to logic: Introduction to model-theoretic semantics of natural language, formal logic and discourse representation theory. *Computational Linguistics*, 21(2):265–268, June 1995.

[2] J. Bouchet and L. Nigay. Icare: A component-based approach for the design and development of multimodal interfaces. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, Vienna, 2004.

[3] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt. A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, Vol. 15(No. 3):289–308, June 2003.

[4] U. catholique de Louvain. Interface de recherche multimodale dans le contenu audivisuel - irma. http://www.irmaproject.net/, December 2008.

[5] L. Cole and D. Austin. Visual object recognition using template matching. In *Proceedings of Australasian Conference on Robotics and Automation*, 2004.

[6] I. Corporation. Open source computer vision library - opencv. http://www.intel.com/technology/computing/opencv/index.htm, January 2009.

[7] J. Curran, S. Clark, and J. Bos. Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of the ACL 2007 Demonstrations Session*, pages 29–32, 2007.

[8] R. D. and M. B. A master-slaves volumetric framework for 3d reconstruction from images. In *Proceedings of the SPIE'07*, volume Vol. 6491, San Jose, USA, 2007.

[9] B. A. Davis J. The representation and recognition of action using temporal templates. Technical Report 402, MIT Media Lab Technical Report, 1997.

[10] R. Engel and N. Pfleger. *SmartKom: Foundations of Multimodal Dialogue Systems*. Springer, Berlin, Germany, 2006.

[11] J. Hockenmaier and M. Steedman. Ccgbank: A corpus of ccg derivations and dependency structures extracted from the penn treebank. *Comput. Linguist.*, 33(3):355–396, 2007.

[12] S. M. Inc. Project grizzly. https://grizzly.dev.java.net/, May 2009.

[13] J.-Y. L. Lawson, A.-A. Al-Akkad, J. Vanderdonckt, and B. Macq. An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components. In *Proceedings of the EICS'09*, Pittsburgh, USA, July 2009.

[14] J.-Y. L. Lawson and B. Macq. Openinterface platform for multimodal applications prototyping. In *ICASSP Show & Tell '08*, Las Vegas, USA, April 2008.

[15] J. C. Martin. Tycoon: Theoretical framework and software tools for multimodal interfaces, 1998.

[16] J. C. Martin and M. Kipp. Annotating and measuring multimodal behaviour - tycoon metrics in the anvil tool, 2002.

[17] H. Mendonça. Meanings4fusion. http://kenai.com/projects/meanings4fusion, May 2009.

[18] U. of Michigan. Soar. http://sitemaker.umich.edu/soar/home, August 2008.

[19] N. Pfleger. Context based multimodal fusion. In *Proceedings of the ICMI'04*, Pennsylvania, USA, October 2004.

[20] D. Reynolds and R. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, Vol. 3(No. 1):72–83, 1995.

[21] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, 2 ed. edition, 2002.

[22] Y. Shoham and K. Leyton-Brown. *Multiagent Systems*. Cambridge, New York, 2009.

[23] D. o. C. S. Stanford University. Protégé platform. http://protege.stanford.edu, January 2009.

[24] C. Town. Multi-sensory and multi-modal fusion for sentient computing. *International Journal of Computer Vision*, 71(2):235–253, February 2007.

[25] O. Vybornova, H. Mendonça, L. Lawson, and B. Macq. High level data fusion on a multimodal interactive application platform. In *Proceedings of IEEE ISM'08*, Berkeley, USA, December 2008.

[26] W3C. Resource description framework - rdf. http://www.w3.org/TR/PR-rdf-syntax/, January 1999.

[27] W. Walker. Sphinx-4: A flexible open source framework for speech recognition. Technical Report SMLI TR2004-0811, Sun Mycrosystems Inc., 2004.

[28] X. Zou and B. Bhanu. Tracking humans using multi-modal fusion. In *CVPR'2005, IEEE Computer Society Conference on*, San Diego, USA, June 2005.