Roma Tre University
Ph.D. in Computer Science and Engineering

Dealing with Multimodal Languages Ambiguities: a Classification and Solution method

Maria Chiara Caschera

Dealing with Multimodal Languages Ambiguities: a Classification and Solution method

A thesis presented by
Maria Chiara Caschera
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Engineering

Roma Tre University
Dept. of Informatics and Automation

February 2009

COMMITTEE:
Ing. Patrizia Grifoni

REVIEWERS:
Prof. Irina Kondratova
Prof. Esteban Zimanyi

To my Family

Abstract

Starting from discussing the problem of ambiguity and its pervasiveness on communication processes, this thesis dissertation faces problems of classifying and solving ambiguities for Multimodal Languages.
This thesis gives an overview of the works proposed in literature about ambiguities in Natural Language and Visual Languages and discusses some existing proposals on multimodal ambiguities. An original classification of multimodal ambiguities has been defined using a linguistic perspective, introducing the notions of Multimodal Grammar, Multimodal Sentence and Multimodal Language.
An overview of methods that the literature proposes for avoiding and detecting ambiguities has been done. These methods are grouped into: prevention of ambiguities, a-posterior resolution and approximation resolution methods. The analysis of these methods has underlined the suitability of Hidden Markov Models (HMMs) for disambiguation processes. However, due to the complexity of ambiguities for multimodal interaction, this thesis uses the Hierarchical Hidden Markov Models to manage the Semantic and Syntactic classes of ambiguities for Multimodal Sentences; this choice permits to operate at different levels going from the terminal elements to the Multimodal Sentence. The proposed methods for classifying and solving multimodal ambiguities have been used to design and implement two software modules. The experimental results of these modules have underlined a good level of accuracy during the classification and solution processes of multimodal ambiguities.

Acknowledgments

I would like to give thanks to my supervisor Patrizia because she has contributed to my growth as a researcher and to the quality of this thesis.
I am grateful to Fernando because he has taught me the many skills required to take my PhD, support, and knowledge.
I would to thank my parents, my brother and all my family because they have helped me to know and to give priority to the things that are important in my life.
I would to thank Arianna and Gabriele because they have been always willing to hear my ideas, no matter how ill-formed they have been, and to help me evolve them.
Finally I would offer thank to my friends and to the people of the Multi Media & Modal Laboratory at CNR for their encouragements.

Contents

List of Tables

List of Figures

Chapter 1

Ambiguity and Multimodal Interaction

1.1 Introduction

Ambiguity plays a very relevant role in communication between people, in computer mediated communication as well as during early-stage of complex problem solving; indeed, ambiguity allows the different involved actors to adapt themselves to each other, producing a convergence to the common goal.
Ambiguity pervasively characterizes the communication among humans, which involves the five senses, and, it can represent an advantage or an obstacle to be overcome in individual and social communication processes. For this reason it is usually preferred to manage ambiguity instead of prevent it.

This thesis dissertation faces the problem of ambiguity in Multimodal Human Computer Interaction according to a linguistic point of view, generalizing and extending methods used for Natural Language and Visual Languages to the Multimodal Languages.

1.2 Problems addressed, solutions and thesis organization

This thesis aims to provide answers to emerging questions such as:

1. *Why it is important to face the problem of ambiguity in multimodal interaction?*
2. *What are solutions and methods provided by the literature and how it is possible to extend and generalize them for Multimodal Languages?*

The first question is arising from the evolution of human-human communication, as it is more and more becoming computer mediated.
Humans communicate using their five senses in a synergistic manner expressing key-concepts involving two or more modalities simultaneously. Indeed, human-human communication involves several communication channels, which use gesture, sketch drawing, handwriting, facial expressions, gaze and speech or their combination. Naturalness of the human-human communication process is related with the high value of freedom degrees that people can use during the communication processes. But a high value of freedom degrees when one person expresses and sends a message implies high quantity of information, and a very complex process characterizes interpretation by the receiver. In fact, people actions sometimes do not correspond to their intentions; this can produce ambiguous and/or incorrect interpretations. That is, ambiguities can arise from the semantic gap between the communicative user's intentions and how the user conveys them. For this reason naturalness is closely related with arising of ambiguities. Indeed, more natural the interaction is more ambiguous can be its interpretation.
Naturalness from human–human communication is transferring itself to Human Computer Interaction (HCI); for this reason all problems of human-human communication connected with the interpretation, *ambiguities detection* and *solution* need to be discussed.
Facing the described problems, identifying critic aspects and proposing solutions corresponds to find an answer to the second question addressed at the beginning of this section (i.e What are solutions and methods provided by the literature and how it is possible to extend and generalize them?).

The thesis in the remaining sections of this chapter provides a description of the evolution of communication processes criticisms and features, focusing on Multimodal communication aspects. Some basic notions on multimodal interactions and an overview of the main relevant multimodal systems are also given.

Chapter 2 provides an overview of the literature on classes of ambiguities in Natural Language and Visual Languages. This sets the background for targeting the problem of ambiguities in multimodal communication, which is discussed in subsequent chapters of the thesis.

Chapter 3 presents an original extension of the classification of ambiguities presented in the previous chapter to the multimodal ambiguities and, it provides the set of rules to identify them. This extension, using a linguistic point of view, is based on the notions of Multimodal Grammar, terminal element of the Multimodal Grammar, Multimodal Sentence and Multimodal Language.

Considering the classifications of modal and multimodal ambiguities provided in Chapter 2 and Chapter 3, the Chapter 4 describes how the literature faces the problem of solving modal and multimodal ambiguities. The Chapter 4 presents several strategies for avoiding and detecting ambiguities grouping them into: prevention of ambiguities, a-posteriori resolution and approximation resolution methods. For each of these types, a general description of the proposed solutions is presented using illustrative examples and the suitability of the different solutions to face the ambiguity problem is assessed.

The analysis of methods for representing and managing ambiguities, has underlined the suitability of Hidden Markov Models (HMMs) for the disambiguation process of multimodal ambiguities. Chapter 5 proposes a model based on the Hierarchical Hidden Markov Models to manage the Semantic and Syntactic classes of ambiguities for multimodal sentences. The proposed approach operates at different modelling levels going from the terminal elements level to the multimodal sentence level.

Chapter 6 presents the design and the implementation of the Multimodal Ambiguities Classifier and of the Multimodal Ambiguities Solver modules, developed on the basis of the methods of ambiguities classification and solution defined in the previous chapters. These modules belong to a general software platform, the MultiModal Language Processing framework, whose architecture has been described. UML diagrams explaining the main components of the modules are given. The chapter ends providing an example of use of the software modules.

Chapter 7 presents some experimental results concerning the two modules developed in the thesis. These results underline a good level of accuracy during the classification process of multimodal ambiguities, whose significance can be considerably improved extending the experimental test.

Chapter 8 concludes the thesis by summarizing the research contributions as well as directions for further research activities.

1.3 From Human -Human Communication to Multimodal Human Computer Interaction

Studies on communication between humans have involved and currently involve experts and researchers of different disciplines for their complexity. Here some elements that describe the human-human communication process are introduced with the scope to borrow them respect to the communication processes and Human Computer Interaction.
A human-human communication process is a symmetric and interactive process that has the aim to exchange information by messages between two people, respectively a sender and a receiver.
The communication process, therefore, is characterized by: the *sender,* the *receiver,* the *exchanged message, the code* used to convey the message, the *channel* (i.e. the communication medium used to opportunely convey the codified message. Some examples are the visual channel, the auditory channel, and so on).

When delivering a message the sender has to turn content into an objective fact. This means the sender has to codify the message, so that it will be understandable to the receiver when transmitted.
When the message arrives to the receiver it has to be decoded: This implies the signs perception and recognition ability of the receiver, as well as the receiver's ability in combining signs and interpreting the message as a whole to obtain its meaning.
The description of the communication process between humans can be helpfully adopted to opportunely explain the *multimodal interaction paradigm*, which in the last years is arising as the new paradigm of Human Computer Interaction.
Some milestones of the HCI evolution are now sketched in order to provide the basic notions on Human Computer Interaction respect to the human-human communication involved in this thesis dissertation.
In the '80s Personal Computers stimulate heterogeneous people (for their skills in computer science, for cultural background, for their goals in using these emerging technologies) to use them. In that period the concept of end user (or simply user) was introduces; it lies for a person that uses Software Applications and PCs for her/his activities (such as for example office activities), and she/he is not required to be a Computer Science professionals. In that period Human Computer Interaction had a significant growth that configured it as the discipline aiming to facilitate the user's interaction with PCs, which involves experts from different scientific areas. Indeed, users had to take charge to learn how to exchange data and information with the first PCs. Each one needed a training period to learn how to use the computerized system, which on its hand was able to recognize a limited and pre-defined commands set as input.
In that period begun the shifting from text-based interaction (which presented many critical aspects such as for example the high value for the user's cognitive load when using it) to graphical interaction. Even if text based interaction was a critical and error-prone activity for no-skilled people in using PCs, it was usually preferred by a little number of specialists in Computer Sciences. However, the need to speed up training and to reduce errors stimulated growing of the WYSIWYG (What You See Is What You Get) interaction paradigm and of the Graphical User Interfaces (GUI). This new

paradigm produced a significant revolution, as it enlarged not only the potential users, but also the potential uses of computerized systems. Indeed, there was a shifting in using personal computing from computational activities only to daily working activities (i.e. office automation and other similar ones).

In the last years the technological evolution of mobile devices has produced a diversification of manners to communicate and interact. Indeed, not only keyboard and mouse are available to interact with the computerised system and not only working activities are supported. Each kind of activity can be potentially supported/executed by different devices in different situations by people having different knowledge, attitudes and features. This new scenario is shifting again the perspective and the paradigm of Human Computer Interaction, which is becoming more and more similar to the human-human communication. This kind of communication is usually multimodal. This new point of view significantly reduces the need of user's training, and the workload for managing user's input is shifted on the interpretation by the device system.

Multimediality and multimodality are concepts with multiple meanings. In [VaA97] multimediality is defined as a way to present and convey information using several different media. In fact, multimodal systems are hardware/software systems that can receive, interpret and process the users' inputs, and they can integrate and enable the coordinated production of two or more interaction modalities as output.

Multimodal interaction permits people to interact with devices using speech, handwriting, sketch, gesture and others inputs as in Figure 1.1; then the multimodal system has to recognize and to interpret inputs from the different users. As the process of multimodal input production by the user, similarly to the communication between humans, is non-deterministic, then the input recognition and interpretation can produce ambiguities. Ambiguities introduced at recognition level are generally due to the fact that imprecision, noises or other similar factors can influence the recognition process.

Ambiguities introduced at the interpretation level are due to the fact that more than one meaning can be associated to the multimodal input. The multimodal system in order to execute its activities by

the multimodal inputs needs to solve the arising ambiguities producing one only interpretation that in Figure 1.1 is denoted as the *correct interpretation.*

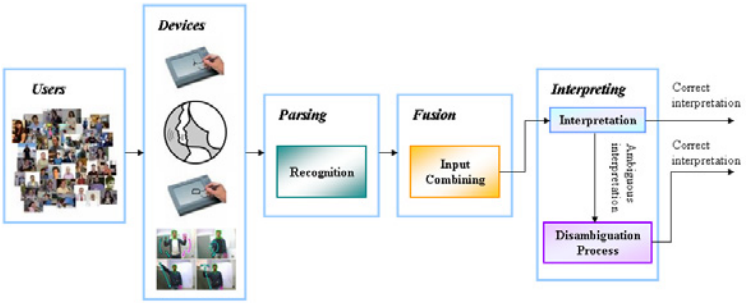


Figure 1.1: The Multimodal Interaction Paradigm

The multimodal systems combine different input modalities according to temporal constraints between modalities using approaches (fusion approaches) described in section 1.4.2 of the present chapter.

Indeed, naturalness and flexibility connected with the use of different modalities of interaction can produce more than one interpretation for each different modality (channel) and for their simultaneous use, and consequently can produce ambiguities. Therefore a focal point is to achieve a coherent and unique meaning of the multimodal input following different approaches according to the relations among used modalities that have been presented in the first chapter.

In this scenario, the role of multimodal interpretation is very important because it is the process that identifies the meaning of the user's input and finds the most proper association to the user intention.

When the meaning of the user's input is not univocally identified more than one interpretation is defined by the system and ambiguities appear.

Next sections of this chapter discuss features of multimodality, the synchronization problems combining different modal components and the different types of cooperation among them. Then, the

chapter shows how to combine and interpret information related to different modalities providing an example of fusion and interpretation approaches. Finally, the chapter ends providing a preview of this thesis work.

1.4 Discussing main multimodal features

Both multimediality and multimodality refer to more than one communication channel that can be viewed as temporal, virtual or physical link that exchanges information between user and system. In particular, communication channels include sensors and tools that receive and process information.
Nigay and Coutaz [NiC93] distinguish between multimediality and multimodality, observing that a multimodal system is able to automatically model, with an integrated approach, information content through a high level of abstraction. This difference leads to identify two main characteristics of multimodal systems, respect to the multimedia ones, according to this abstraction capability:

- *fusion:* a multimodal system integrates (fuses) the different data types by different input/output channels; and
- *temporal constraints:* a multimodal system manages temporal constraints imposed by information processing to and from input/output devices.

Definition 1: *A multimodal system is an HW/SW system that is able to receive, to recognize, to interpret, and to process input, and that generates outputs involving two or more interactive modalities in an integrated and coordinated way*. □

Communication among people is often multimodal, as it consists of combining different modalities. Multimodal systems allow several modalities of communication to be harmoniously integrated, making the system communication characteristics more similar to the human-human communication features. Therefore, discussing multimodality involves some fundamental aspects such as modalities synchronization and modalities integration. For this

reason, sections 1.4.1 and 1.4.2 describe the different perspectives proposed by the literature to synchronize and to fuse inputs.

1.4.1 Modalities Synchronization

When designing a Multimodal System it is possible to consider different points of view for modalities synchronising if a client server or a stand-alone application has to be defined.
Indeed, for a client server application it is necessary to consider aspects related to "the synchronization behaviour of an application", which "describes the way in which any input in one modality is reflected in the output in another modality" [W3C03].
For stand-alone applications it is necessary to consider the aspects that reflect the way in which the different input modalities are combined according to temporal constraints.
The synchronization behaviour is defined referring to the synchronization granularity that specifies the level at which the application coordinates interactions.
The synchronization granularity is specified in [W3C03], and it considers several levels of synchronization:

- **Event-Level:** if the inputs of one mode are received as events and immediately propagated to another mode.
- **Field-Level:** if the inputs of one mode are propagated to another mode after a user has changed the input field or the interaction with a field is terminated.
- **Form-Level:** if the inputs of one mode are propagated to another mode after a particular point of the interaction has been achieved.
- **Page-level:** if inputs in one mode are reflected in the other only after submission of the page.
- **Event-level synchronization:** if inputs in one mode are captured at the level the individual DOM (Document Object Model- which is a standard interface to the contents of a web page) events and immediately reflected in the other modality;
- **Event-level input-output:** if input is synchronized with output media;
- **Media synchronization:** if output media are synchronized between output media as specified by SMIL

(Synchronized Multimedia Integration Language), which is typically used for "rich media"/multimedia presentations;

- **Session level:** if the application suspended in one mode can be resumed in the same or another modality.

When designing a multimodal stand-alone application it is necessary to consider the different modal inputs synchronization, as it deeply influences their interpretation.
In literature this kind of inputs synchronization is defined as:

- **Sequential:** (Seq in Table 1.1) if the interpretation of the interactive step depends on one mode and the modalities can be considered one by one.
- **Time-Independent Synchronized:** (TIS in Table 1.1) if the interpretation of the interactive step depends on two or more modalities and the modes are simultaneous.
- **Time-Dependent Synchronized:** (TDS in Table 1.1) if the interpretation of the interactive step depends on two or more modalities and the semantic dependence of the modalities has a close temporal relationship.

Studies presented in [OCW00] have underlined that when users interact by speech and digital ink pen input they employ sequential and synchronized model. The synchronization model can be identified when user starts to interact and it holds over the overall section, therefore when the user defines the synchronisation model it persists. This fact allows multimodal systems to detect the synchronization model in order to improve the correctness of the interpretation.
Considering the temporal relations among chunks of information belonging to different modalities, they can be expressed using the temporal relationships defined in [AlF94]. Allen [AlF94] defines the set of all relationships between the temporal intervals Δt_1 and Δt_2 as shown in Figure 1.2.

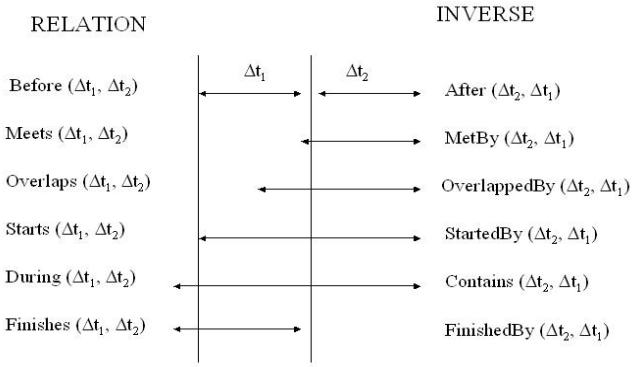


Figure 1.1.2: Temporal relationships among chunks of information belonging to different modalities

These relationships can be used associating the two temporal intervals Δt_1 and Δt_2 to two different modalities.
These temporal relationships are used in order to define the time slot that includes chunks of information belonging to different modalities that are "CloseBy". The satisfaction of this specific relationship defines which chunks of information combining during the interpretation process.
Considering the Figure 1.2 two temporal intervals $\Delta t1$ and $\Delta t2$ are NotCloseBy if they are in a Before and After relationship; otherwise they are CloseBy [CFG07].
The selection of chunks of information that are temporally close-by allows defining sets of chunks of information to combine using classes of cooperation between modalities. In particular we will consider the classification proposed by Martin [Mar97] that defines the following six cooperation classes:

- **Complementarity:** different chunks of information comprising the same command are transmitted over more than one mode.
- **Equivalence:** a chunk of information may be transmitted using more than one mode.
- **Redundancy:** the same chunk of information is transmitted using more than one mode.

- **Transfer:** a chunk of information produced by one mode is analysed by another mode.
- **Concurrency:** independent chunks of information are transmitted using different modalities and overlap in time.
- **Specialization:** a specific chunk of information is always transmitted using the same mode.

A common belief establishes that chunks of information coming from different modalities are often redundant. However, the use of different modalities should count the complementarity of the chunks of information. For example, speech and sketch inputs often convey complimentary information at the semantic level defining the subject, the verb and main objects by speech modality and the location by sketch modality. Users only occasionally repeats the same information using different modalities, therefore analysing the interaction process speech and sketch are rarely redundant.
Anyway each kind of cooperation between modalities can improve the flexibility of the user-system interaction.
In [Bel01] synchronising different modalities is faced from the perspective of relations among modal components, which are classified as:

- **Active:** (Act in Table 1.1) when two events, produced by two different devices, cannot be completely and correctly interpreted without ambiguities if one of the two events is unknown.
- **Passive:** (Pas in Table 1.1) when an event produced by a given device cannot be completely and correctly interpreted without ambiguities if the state of the other devices is unknown.

Table 1.1 presents correspondences among types of cooperation, input synchronization, and relation among modal components [CFG07b].

Table 1.1.1: Correspondences among types of cooperation, relation among modal components and input synchronization

TYPES OF COOPERATION	RELATION AMONG MODAL COMPONENTS	INPUT SYNCHRONIZATION
C	Act, Pas	TDS, TIS
E	No relation	Seq
R	Act	TDS, TIS
T	Pas	Seq
CC	No relation	No synchronization
S	No relation	Seq

The type of cooperation among modalities is closely connected with the interpretation of the multimodal message. This needs to look at the fusion process, which can be obtained by signal or information fusion at the semantic level. In semantic fusion, time is very important as the information chunks provided by different modalities are integrated if they are temporally close. The information chunks produced by different modalities are fused considering them as events.
The following section will present the fusion concept and some of the existing multimodal fusion approaches.

1.4.2 Modalities Fusion Approaches

Multimodal interfaces provide the user with multiple interaction paradigms through different types of communication input and data fusion represents one of their main relevant processes. That is, an important issue in multimodal interaction is the integration and synchronization of several modalities in a single system.
A common belief establishes that different modal inputs involved in multimodal input have to be temporally co-occurrent. This overlap defines which inputs to combine during the interpretation

process. However, several studies have underlined that speech and gesture modalities are often independent and synchronised during the multimodal interaction and therefore the synchronization does not always involve the simultaneity. Empirical facts underline that multimodal signals do not sometimes temporally co-occur neither during human-human communication nor during multimodal communication between users and system; therefore to design a multimodal system it is necessary to take into account not only temporal constrains but also other several features.

Therefore, there is an emerging need for integration among the various input modalities, through signal integration and semantic fusion, and an additional need to disambiguate the various input modalities and coordinate output modalities, to enable the user to have a range of integrated, coordinated interaction modalities.

In order to combine information provided by different modalities, the literature proposes two main approaches: *signal fusion* that matches the modalities to obtain a low-level of interpretation by grouping the input events in multimodal events; and information fusion at the *semantic level* that transfers the multimodal inputs to the high-level interpretation module, in order to obtain the meaning of their events.

The first approach is known as early fusion and during the recognition process based on this approach the interpretation of one modality affects the interpretation of another modality. This approach is preferred for example for matching and synchronizing modalities as speech and labial movement that are temporally close. However it is not suitable when modalities differ at the temporal or semantic level [WOC99].

The semantic fusion is used for modalities that differ in a temporal and semantic scale, such as speech and digital pen ink inputs. In this approach, time is very important because chunks of information with different modalities are considered, and integrated if they are temporally close. This approach is known as late fusion and it integrates at the semantic level chunks of information that are semantically complementary; they are integrated al the level of utterance. Systems presented in the literature based on the late fusion approach have different recognisers for different modalities. The late fusion defines the type of actions that will be triggered by

the user and the used parameters. These parameterised actions are passed to the application dialog manager to start their execution.
Examples of systems that use these fusion approaches are: QuickSet [CJM97] and Portable Voice Assistant [BMM98]. These systems will be described in the specific section according to the used fusion approaches to represent events of multimodal inputs (sections 1.4.2.1- 1.4.2.3).
In this section an example of multimodal information processing based on the late fusion is shown in Figure 1.3 where chunks of information are processed and recognized in parallel.

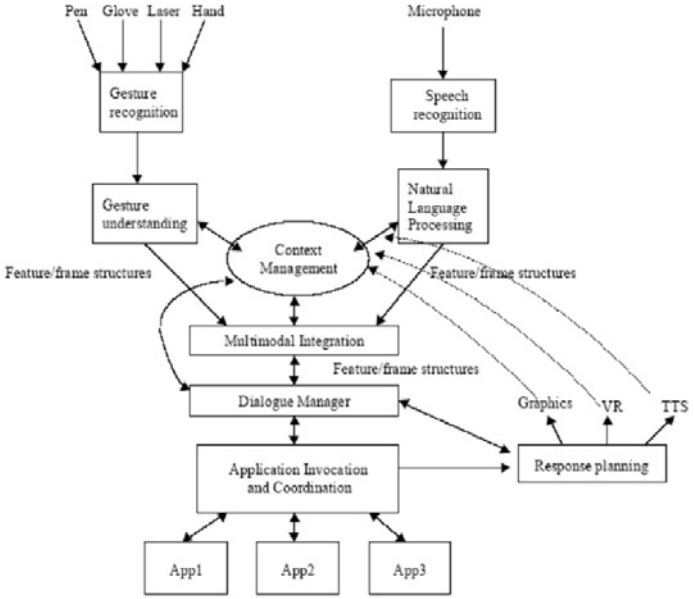


Figure 1.3: Example of information processing flow for speech and gesture modalities [Ovi03]

In this case speech and gesture inputs are separately recognized and they are fused trough the module Multimodal Integration considering the context and the current dialogue. During this integration process each different possible interpretation is classified according to its probability of correctness. The

interpretation with the highest probability is sent to the control component that translates the interpretation into a sequence of commands/actions.
The integration can be also carried out using an intermediate approach between the signal integration and the semantic fusion.
In order to semantically integrate information referring to different modalities it is necessary to use a common representation.
In literature there are several approaches to represent events of multimodal inputs:

- Typed Feature Structures [CJM97];
- Melting Pots [NiC95];
- Partial Action Frame [VoW96][VoW97][VoM98].

Semantic fusion is not further defined by the characteristics of the events because the above approaches can be used indifferently. For example, both Typed Feature Structures and Partial Action Frame can achieve speech and pen fusion. In literature, events for fusing gesture and speech have been represented by Typed Feature Structures. The Melting Pot approach can be used to fuse speech, keyboard and mouse inputs.
These structures are combined using different fusion approaches that will be dealt in the following sections.

1.4.2.1 Typed Feature Structures

Using the first approach multimodal inputs are transformed into typed feature structures that represent the semantics attributed to the various modalities.
A Typed Feature Structure includes two different types of information [VaS98]:

- Type: that defines the class of the objects described by the structure;
- A set of couples attribute-value: where each value can be defined by feature structures.

$$\text{LING-SIGN}\begin{bmatrix}\text{PHON} & /\text{DOG}/ & \\ \text{SYNTAX} & \begin{bmatrix}\text{CAT} & \text{N} \\ \text{CASE} & \text{NOM} \\ \text{N\_TYPE} & \text{PROPER}\end{bmatrix} \\ \text{SEM} & \end{bmatrix}$$

Figure 1.4: Example of Typed Feature Structure [VaS98]

In particular, information about the type allows representing the typed feature structure as a hierarchical structure as defined in the following figure.

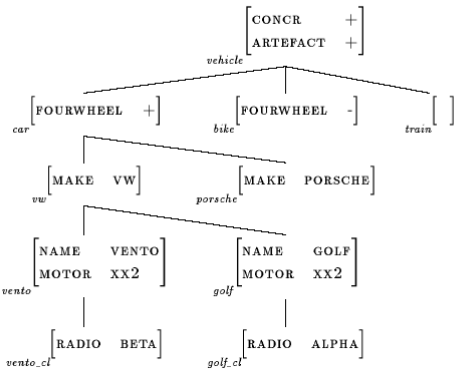


Figure 1.5: Representation of the vehicles' class using the graph of the hierarchical types [VaS98]

This figure underlines that each child inherits information from its parent through the link from the parent to the child, and each child improves the structure by a new couple attribute-value.
An important aspect connected with the feature structures is the unification that allows combining them. This operation firstly verifies the consistence of the chunks of information to combine and, after, it unifies them in a unique structure.

$$A = \begin{bmatrix} \text{MANAGER} & \begin{bmatrix} \text{NAME JONES} \\ \text{TEL 2345} \end{bmatrix} \end{bmatrix}$$

$$B = \begin{bmatrix} \text{MANAGER} & \begin{bmatrix} \text{SEX MALE} \end{bmatrix} \end{bmatrix}$$

$$A \cup B = C = \begin{bmatrix} \text{MANAGER} & \begin{bmatrix} \text{NAME JONES} \\ \text{TEL 2345} \\ \text{SEX MALE} \end{bmatrix} \end{bmatrix}$$

Figure 1.6: Unification of feature structures [VaS98]

The unification of two feature structures A and B defines the feature structure C (Figure 1.6) that contains the minimum information content that includes information defined by both A and B. This is a recursive process as Figure 1.7 shows.

$$\begin{bmatrix} \text{AGR} \begin{bmatrix} \text{PER} & 1 \end{bmatrix} \\ \text{CASE} \quad \text{NOM} \end{bmatrix} \cup \begin{bmatrix} \text{AGR} \begin{bmatrix} \text{NUM} & \text{SG} \\ \text{GEN} & \text{MAS} \end{bmatrix} \\ \text{CAT} \quad \text{N} \end{bmatrix} = \begin{bmatrix} \text{AGR} \begin{bmatrix} \text{PER} & 1 \\ \text{NUM} & \text{SG} \\ \text{GEN} & \text{MAS} \end{bmatrix} \\ \text{CAT} \quad \text{N} \\ \text{CASE} \quad \text{NOM} \end{bmatrix}$$

Figure 1.7:Recursive unification of feature structures [VaS98]

An example of system that uses Typed Feature Structure is QuickSet [OCW00] [CJM97] [MCO98]; it is a collaborative multimodal and multi-agent system that allows a map-based interaction using speech and gesture modalities. Inputs coming from these two modalities are considered separately by the speech and gesture recognition agents that produce typed feature structures [JCM97].
Speech and gesture inputs can define partial commands that are defined as under specified feature structures where some values can be not specified. In [JCM97], the authors provided an example where the user says “m1a1 platoon” defining the following structure:

$$create\_unit \begin{bmatrix} \text{object} : & \begin{bmatrix} \text{type} : \text{m1a1} \\ \text{echelon} : \text{platoon} \end{bmatrix}_{unit} \\ \text{location} : & \begin{bmatrix} \quad \end{bmatrix}_{point} \end{bmatrix}$$

Figure 1.8:Under specified structure [JCM97]

In detail, this system fuses information by different modalities considering:

1. Temporal level;
2. Probabilistic level;
3. Selection of the best candidate.

At the temporal level the Quickset combines speech and gesture inputs that are temporally close defining their possible combinations.
Considering the probabilistic level, the system assigns a weight to each possible combination according to temporal and semantic features. The value of the weight is obtained combining the probabilities of the modal inputs obtained by the Members-Teams-Committee-technique [WOC99] [OvC00] [WOC02]. This statistical hierarchical technique has been designed in order to decrease the uncertainty connected with the candidate different interpretations. It is based on a “divide et impera” architecture that assigns weights in a bottom-up sequence and it is defined by three levels: multiple members, multiple teams, and committee. The members are single recognisers that process a set of data providing arrays about recognition results and probabilistic estimations connected with inputs. This information is conveyed to team leaders and every teams weight information using a different weighting measure. Finally, the committee weights results obtained by the different teams and it classifies the possible multimodal interpretations. The interpretation at the top of the list is sent to the application bridge agent and it is executed.

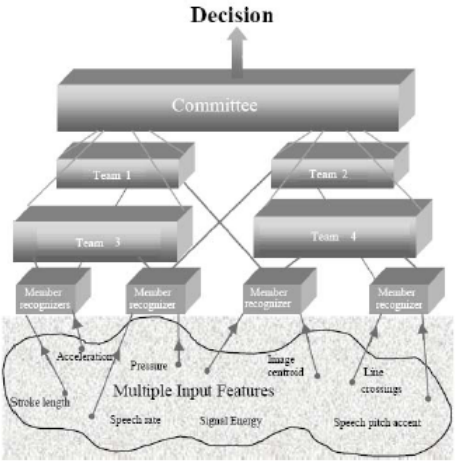


Figure 1.9: Members-Teams-Committee architecture [WOC02]

In the third step the system selects the best candidate among the possible combinations according to the current state of the system and the dialogue context.
Considering the previous example, authors suppose that the command is completed by the gesture modality by a drawing that can be interpreted as both a point and a line. Therefore two possible structure are defined as the Figure 1.10 and the Figure 1.11 show:

$$command\left[\text{location}: \left[\begin{array}{l}\text{xcoord}: 95305\\ \text{xcoord}: 94365\end{array}\right]_{point}\right]$$

Figure 1.1.10: Structure defined by the point interpretation [WOC02]

$$command\left[\text{location}: \left[\begin{array}{l}\text{coordlist}:\\ [(95301, 94360),\\ (95305, 94365),\\ (95310, 94380)]\end{array}\right]_{line}\right]$$

Figure 1.11: Structure defined by the line interpretation [WOC02]

The speech interpretation defined in Figure 1.8 requires the field *point* and therefore the correct combination is the point interpretation. The multimodal interpretation is the following:

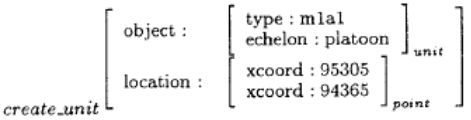


Figure 1.12: Multimodal interpretation [WOC02]

1.4.2.2 Melting Pot

A melting pot encapsulates types of structural parts of a multimodal event. The content of a structural part is a time-stamped piece of information. The melting pot is a bi-dimensional object as the following figure defines.

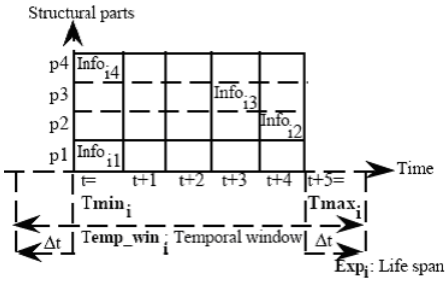


Figure 1.13:Structure of a melting pot [NiC95]

The structural part models the structure of the command that the Dialogue controller is able to interpret. A command is complete when all its structural parts are defined. Structural parts that compose the same command can have different timestamps; the different information connected with the same structural part can be redundant or inconsistent [NiC93]. A melting pot is removed when its lifecycle is complete or spent.

An example of melting pots fusion is shown in Figure 1.14. In this example the user says the sentence " Fly from Boston to this city" at the time t_i and she/he selects the city Denver on a map at the time t_{i+1}.

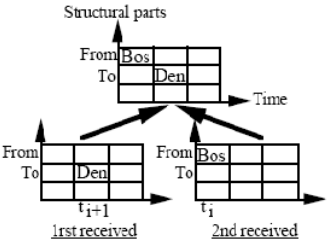


Figure 1.14: Melting pots fusion [NiC95]

The fusion process combines information belonging from the two melting pots in a third one that is defined in the top of the Figure 1.14.
The melting pots are constructed by events in elementary inputs with different mechanisms of fusion: micro-temporal, macro-temporal, and contextual fusion.
The micro-temporal fusion combines two information units produced concurrently or very close to one another. This fusion is applied when two melting pots, m_i and m_i' are complementary and temporally close.

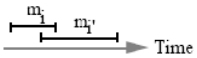


Figure 1.15:Micro-temporal fusion [NiC95]

The macro-temporal fusion combines sequential or temporally close information units, when these units are complementary. The system processes these units in a sequential way. This fusion is applied when the structural parts of the two melting pots are complementary and their timestamps are not overlapped but they are in the same temporal slot.

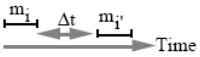


Figure 1.16:Macro-temporal fusion [NiC95]

Finally, the contextual fusion combines information units according to semantic constraints without considering temporal constraints. This fusion combines two melting pots that are complementary and belong to the same context.

1.4.2.3 Partial Action Frame

A further approach is the Partial Action Frame [VoW96][VoW97] [VoM98] that is a frame-based approach.
It considers the input modality channels as parallel streams that are aligned in action frames and jointly segmented in parameter slots.
Inputs from each modality are represented as an information stream that consists of a sequence of tokens that are combined in order to determine the output action and its parameters. In particular, a multimodal input event is defined as a set of parallel streams that can be aligned and jointly segmented such that each part of the segmented input influences part of the interpretation. Each part of the segmented input is a parameter slot that defines one action parameter. In each parameter the input segments slot contain enough information to determine the value of the corresponding parameter.

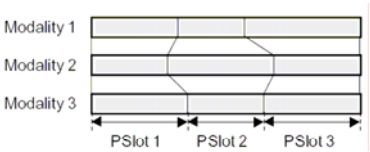


Figure 1.17:Alignment and segmentation of multimodal input [VoM98]

In detail, each unimodal input stream is partitioned into input events that define the sequence of tokens. An input event in the unimodal stream includes different information that is dependent on application and modality. For example considering the speech

modality data coming over the speech channel may be divided into utterances based on periods of silence or prosody information such as a drop in pitch at the end of a sentence. Using this approach input events are complete speech acts.
Considering speech and pen input events, they start when the user begins speaking or drawing, and end when no input signal is detected within a predefined time-out interval. Considering this approach input events from different modalities are grouped if they occur close together in time, for example if they overlap or if one event starts within a time-out interval after another event ends, as describe the following figure.

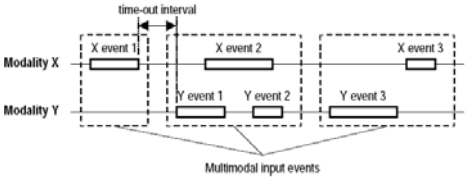


Figure 1.18:Input grouping using temporal proximity [VoM98]

This approach separately parses the input of each mode and then analyses and transforms them into a semantic frame containing slots that specify the control parameters.
Using this approach modal fusion is implemented both at syntactic (temporal alignment) and semantic levels. In detail, temporal alignment is based on time stamps associated with input tokens and the semantic fusion uses some restriction based on time-stamps.
The integration process uses mutual information network architecture and a dynamic programming algorithm to generate an input segmentation. After obtaining the frames with the corresponding slot information, grammar-based algorithms are applied to capture syntactic elements and to extract their meaning.
Information in the partial action frames can be incomplete or ambiguous if not all elements of the command were expressed in a single modality. These partial frames are combined into a complete frame selecting slot values from the partial frame to maximize a score based on their mutual information.
This approach is similar to the melting-pot approach.

This frame-based approach is used in Portable Voice Assistant (BBN) [BMM98] that integrates speech and pen input events defining the user's request by a frame-based description. This system provides an application for on-line vehicle repair manual and parts ordering.

1.4.3 Interpretation Approaches

When chunks of information conveyed by different modalities are fused, the system has to define the correct interpretation of them. Therefore an important unit for building multimodal systems is the interpretation process.
The interpretation of user input is strictly connected with different features, such as available interaction modalities, conversation focus, and interaction context. A correct interpretation can be reached by simultaneously considering semantic, temporal and contextual constraints.
For example in multimodal system based on video and audio inputs [HaS04] the interpretation defines a multimodal corpus of digital and temporally synchronized video and audio recordings of human monologues and dialogues.
Literature provides different methods to interpret multimodal inputs. Methods for fusing and interpreting multimodal inputs can be divided taking into account the level where the fusion has been executed: at the acquisition level; at the interpretation level; and at the decision level.
This section presents a selection of mathematical methods for interpreting multimodal and modal inputs using recognition-based fusion strategies, decision-based fusion strategies and hybrid multilevel fusion strategies.
In particular, considering recognition-based strategies the following section will analyse how Hidden Markov Models (HMMs) can be used to interpret user input. This section provides attributed relational graphs method as example of decision-based strategies. Taking into account hybrid multilevel strategies the following section will provide examples of this class of methods.

1.4.3.1 Recognition-based approaches

One of the meaningful recognition-based approaches is the Hidden Markov Models, which are stochastic models that allow supporting the interpretation of the user input. These methods have been used for modelling speech input [Fos98] different sketching styles [SeD05], handwriting [HBT94] and gesture [EKR98].
In detail, for defining the input interpretation this method uses a sequence of vectors of features extracted from modalities used during the interaction process. This sequence is compared with a set of hidden states that can represent the speech word, the letter of the handwritten word or the drawing by the sketch modality according to the definition of the parameters of the model and the modalities to be interpreted. The purpose of this method is to identify the model that has the highest probability to generate the observed output, given the parameters of the model.
In detail, the set of the parameters that characterises an Hidden Markov Model are:

- the set of states,
- the prior probabilities that define the probability that a specific state is the first state,
- the emission probabilities that define the probability of a specific observation in a specific state,
- the observation sequence.

In this model an important parameter to define is the number of HMM states. In fact, in the case that this number is low thence the discrimination power of the model is reduced because more than one pattern should be modelled on one state. On the other hand whether this number is excessively high thence the number of training samples is not sufficient respect to the number of the model parameters. Therefore, in literature two different approaches have been developed to determine the number of the HMM states. The first one uses a fixed number of states training each category of samples with the same number of states. The second approach uses a variable number of states dividing each component in sub-

components according to a specific criterion and associating each sub-component to an HMM state.
A detailed description of these models will be provided in the chapter 3 analysing how they can be used in order to manage issues connected to the interpretation process.

1.4.3.2 Decision-based approaches

The interpretation process can be applied also at the decision level. Therefore, this section provides an analysis considering how an example of decision based approach, the referent resolution approach, can be used to correctly interpret information come from multiple modalities such as sketch, speech, gesture and handwriting.
In particular, when there are different types of interaction modalities, it is not easy to identify all objects, which a user refers during her/his input, and to correctly interpret the combination of the modal inputs. Users can make precise, complex or ambiguous references. Consequently literature has proposed a process that finds the most proper referents to objects, which a user refers to during the input. This process is known in literature as referent resolution [CPQ06]. In detail, this method aims to find the most proper meaning for the expression defining the user inputs as a referring expression, and the specific entity or entities to which she/he refers as referents.
The referent resolution problem implies the dealing of different types of references. This problem has been coped with probabilistic approaches, for example using a graph-matching algorithm [CHZ04].
This approach represents information belonging to different modalities and context by attributed relational graphs [TsF79] and the referent resolution problem is faced as a probabilistic graph-matching problem.
In detail, information concerning each modality is represented by an attributed relational graph, and the referent resolution problems are dealt using information about properties of referring expressions and referents and inter-relations among nodes of the same graph. Each node of the graph represents an entity and it contains semantic and temporal information about the entity. While

an edge defines relation between two entities and it includes: temporal relation that represents the temporal order between the two entities during the interaction process; and semantic type relation between the two entities that expresses if the two entities express the same semantic type.
In [CHZ04] three attributed relational graphs (AGRs) are defined: the first contains information about the speech input; the second is connected with the gesture input; and the last refers to the conversation context defining an history AGR that includes information about objects referred during the last interaction.
A further decision-based approach is proposed in [RSH05] that interprets independently modal inputs therefore the user can express different modal inputs using different languages. The main components of this approach are the semantic network and the interpretation reaction module.

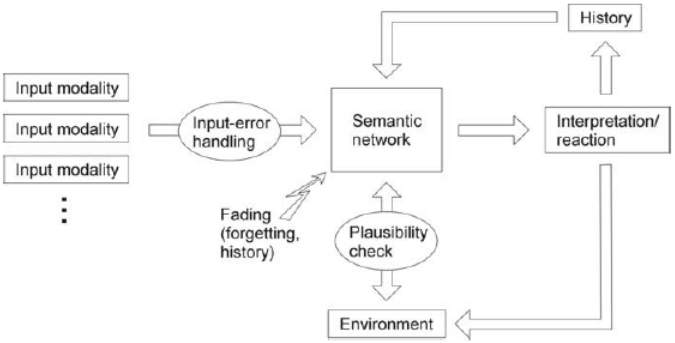


Figure 1.19:Structure of decision-based fusion [RSH05]

The semantic network component is composed of three elements [AAA94]: a semantic network that represents the dictionary where each concept is represented by a frame; a dictionary that allows inferring concept from each term; and a structure that contains knowledge about concepts and their relations. The semantic network is composed by nodes, where each one contains the vocabulary term that it refers to and its activation value (i.e. the value that makes a state reachable). Links among nodes are weighted and the activation of each node ensues from the activation

of connected nodes in the net. When a node is activated by user input the activation value of the node is computed and it affects the activation values of connected nodes. The activation value decreases after a period when it is not activated and this value is decreased according to the forgetting rate function.
The second component is the interpretation/reaction module where possible interpretations of the user input are defined as a frame. An interpretation is composed of a list of slots connected with nodes of the semantic network. Connections and dependencies are integrated in the structure of the frame as the following figure shows.

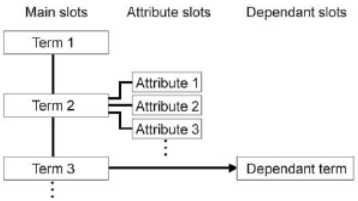


Figure 1.20:Frame's structure [RSH05]

The main slots correspond to the links in the net and each slot defines a dependency. For example if the user's input is "put that blue box there (x/y)", the frame structure is <(move)(object)(there (location))> where the main slots are <(move)(object)(there)> and the slot (there) needs an additional slot (location). In the figure the slot of the Term 3 needs a further input ("dependant term") that have a specific type. The sequence in which main slots are filled is not relevant, while dependent slots have to be filled in the same order of the connected main slots. The third kind of slots is the attribute slot and it is also connected with the semantic network. For example if the main slot is in "object" than the attribute slots can be "size, colour, position".

1.4.3.3 Hybrid multilevel approaches

The last section of this paragraph describes examples of hybrid multilevel approaches, those interpret multimodal inputs at different levels: acquisition, recognition and decision levels.

An example of this class of methods is finite-state mechanisms [JBV02] based on weighted finite-state automaton with multimodal grammar. This method parses, understand and integrates speech and gesture inputs by a single finite-state devices defining three-tape finite state automaton which represent speech input, gesture input and their combined interpretation. The speech and gesture streams are combined considering their content into a single semantic representation. The interpretation of the multimodal command is defined using a multimodal context-free grammar where each terminal symbol contains three components: the spoken language stream, the gesture stream and the combined meaning. However, this approach is limited due to the fact that the multimodal grammar is not able to recognize any string that is not accepted by the grammar. To overcome this issue a corpus-driven stochastic language model has been defined [JoB05a]. In this approach the finite-state-based interpreter is combined with an edit-based transducer for lattice inputs defined by speech and gesture modalities. This edit machine allows integrating stochastic speech recognition with handcrafted multimodal grammars.
This approach is used in MATCH (Multimodal Access To City Help) [JBV02] that enables users to interact using pen, speech or dynamic combinations of the two. It is a multimodal city guide available on mobile devices and usable in different environments and contexts. It provides restaurants and subway information for New York City, combining speech and pen inputs. The output is given by speech, icons and call-outs. All MATCH commands must be expressed by speech, pen, or multimodally. This is achieved by considering: parsing, integrating, and understanding of speech and gesture inputs in a single multimodal grammar. This grammar is formalized into a multimodal finite-state device.
Furthermore Literature describes some approaches to interpret user's inputs combining parse tree [Col97] with semantic grammars [GaW98]. This approach considers semantic information associated with the interpretation of the sentence and converts the parse tree into a set of typed feature structure [Car92]. These feature structures represent objects in the domain and relationships between objects.

1.5 Conclusions

In the last decades critical aspects in HCI have been gradually shifted from users to computerized systems, making systems more flexible and easy to use and interaction more natural. This transformation on one hand simplifies the user's activity, because she/he does not need to have a training before using the system, and on the other hand it requires the systems has to be able to recognize and to interpret complex inputs such as the multimodal ones.

A positive effect of this shifting consists of the wide diffusion of the use of computerized systems in the everyday life activities. The negative aspect consists of the increasing complexity of the multimodal system. This complexity is more and more relevant when a system has to recognize and interpret multimodal inputs. Indeed, multimodal applications can combine different information, such as for example visual information (involving images, text, sketches and so on) with voice, gestures and other modalities to provide flexible and powerful dialog approaches enabling users to choose one or more of the multiple interaction modalities. Multimodality aims to imitate the communication among humans that is based on sensorial systems, because it can produce a quality improvement of the communication processes in terms of naturalness. However naturalness, as in human-human communication, can imply a very high level of freedom-degrees at recognition and interpretation level, producing more than one result for each modal or multimodal input. This is the problem of *ambiguity*, which usually arises when humans communicate among them visually, and/or using natural language, etc.. The same problem arises when a user interacts with a multimodal system. Literature on ambiguities in Natural Language and in Visual Languages contains relevant research results that are presented in chapter 2 and in chapter 4 of this thesis. Starting from these researches chapter 3 proposes an original classification of multimodal ambiguities whose solution methods are proposed in chapter 5.

Chapter 2

From Natural Language to Visual Language ambiguities classification

2.1 Introduction

The ambiguity of communication is not a new problem and it has been studied from different points of view.
Some among the most relevant studies on ambiguity are related with the *Natural Language* (NL), visual communication and *Visual Languages* (VL). Indeed, communicating by text, speech or using visual information (tables, graphics, images) is very frequent.
The communication process requires sharing *a common meaning* for a communication act (such as a sentence for natural languages, or more generally a message) exchanging between actors involved in the process. When a gap appears between the sender communication intention and the meaning that the receiver gives at the message, than an error or an ambiguity can arise.
A NL ambiguity can be detected if two or more alternative linguistic structures can be identified for the same sentence producing a non-deterministic situation. Natural Language Understanding involves knowledge of words, their meaning (lexical semantic), the syntactic structure and how different components are combined in order to assume larger meanings (compositional semantic). Therefore, unambiguous understanding of a sentence

implies: 1) a words dictionary sharing and knowledge of the meaning of each word; 2) a well known and shared syntactic structure that enables to identify both, the role for each word in a sentence and how different words are grouped and connected, and finally 3) knowledge about the meaning of words opportunely combined.
The concept of ambiguity for visual information was described by the Gestalt theory that, with its idea that a figure can be exchanged with its background according to the adopted point of view, has deeply influenced the cultural scenario during the 20th century. But, according to the goal of this chapter, the discussion is here mainly focused on the ambiguity in visual communication and in particular when using Visual Languages in Human Computer Interaction. Starting from Lexical and Syntactic ambiguities and their sub-classes in Natural Language, the analysis of ambiguities has been discussed and extended for Visual Languages. Indeed, before studying and proposing a classification of multimodal ambiguities in Chapter 3, this chapter presents how the literature on Natural Languages and the literature on Visual Languages face the ambiguity classification problem.
The last section of this chapter discusses notions introduced and results obtained in NL and VL and points to the opportunity of extending them to the multimodal interaction.

2.2 Classification of natural language ambiguities

Very detailed works on natural language ambiguities are provided by the literature. In [BKK01] are introduced four classes of linguistic ambiguities: *lexical*, *syntactic, semantic* and *pragmatic* ambiguities, which are presented in the next sub-sections.

2.2.1 Lexical ambiguity

A Lexical ambiguity appears when one word has more than one generally accepted meaning and it can identify two different situations: homonymy and polysemy.

Homonimy situation occurs when two words have the same representation but different meanings.
Let us consider for example the word *bank.* It can have different meanings and can be interpreted as a *shore of a river* or as a *financial institution* according to the contexts.
Polysemy occurs when a word has different meanings, but connected each other. For example the word *green* can identify the physical object or its colour. Another example is the word *paper*. It can indicate for example an article to be published or a sheet of material made of cellulose pulp.
Differently from the homonimy, for polysemy the different meanings are related to one another, and the process of extension of similar meanings clearly arises [ClC79].

2.2.2 Syntactic ambiguity

Syntactic ambiguity lies for an ambiguity that arises from the way the sentence is composed. The grammatical construction of the sentence or phrase can have more than one way to be read.
The syntactic structure of the sentence in natural language can be represented by a parse tree according to a given formal grammar. A parse tree, or *concrete syntax tree,* is an ordered and rooted tree obtained analysing an input sequence in natural language in order to determine its grammatical structure. The internal nodes are labelled by non-terminal of the grammar, leaf by terminal symbols. Terminal and non-terminal symbols are symbols used by a formal grammar in its production rules.
A sentence in natural language presents a syntactic ambiguity if there are multiple possible parse trees for a given sentence and this section provides some examples of how parse trees reflect syntactic ambiguities.
For describing some examples of syntactic ambiguities it is used the set of non-terminal symbols defined by Penn Treebank [MSM94] corresponding to the syntactic categories defined for natural language (Table 2.1).

Table 2.1:Penn Treebank syntactic categories

Part-of-sentence	**TAG**
Sentence	S
Noun phrase	Np
Verb phrase	Vp
Prepositional phrase	Pp
Coordinating conjunction	Cc
Cardinal Number	Cd
Adjective	Jj
Adjective, comparative	Jjr
Adjective, superlative	Jjs
Adverb	Rb
Adverb, comparative	Rbr
Adverb, superlative	Rbs
Determiner	Dt
Existential there	Ex
Foreign Word	Fw
Interjection	Uh
List Item Marker	Ls
Modal	Md
Noun, singular or mass	Nn
Noun, plural	Nns
Particle	Rp
Predeterminer	Pdt
Preposision or subordinating conjunction	In
Proper Noun, plural	Nnps
Proper Noun, singular	Nnp
Symbol Should be used for mathematical, scientific or technical symbols	Sym
To	To
Verb, 3rd person singular present	Vbz
Verb, base form subsumes imperatives, infinitives and subjunctives	Vb
Verb, gerund or persent participle	Vbg
Verb, non-3rd person singular present	Vbp
Verb, past participle	Vbn
Verb, past tense includes the conditional form of the verb to be	Vbd
Wh-determiner	Wdt
Wh-pronoun	Wp
Possessive wh-pronoun	wp$
Wh-adverb	Wrb

Syntactic ambiguities are classified in: *analytic, attachment, coordination* and *gap* ambiguities, and some examples are presented according to these different classes.

2.2.2.1 Analytic ambiguity

Analytic ambiguities can be identified when the syntactic role in a sentence (i.e. it is a noun phrase, a verb, an adjective, an adverb, a pronoun, a preposition, etc.) is itself not clear.
Let us consider for example the sentence:

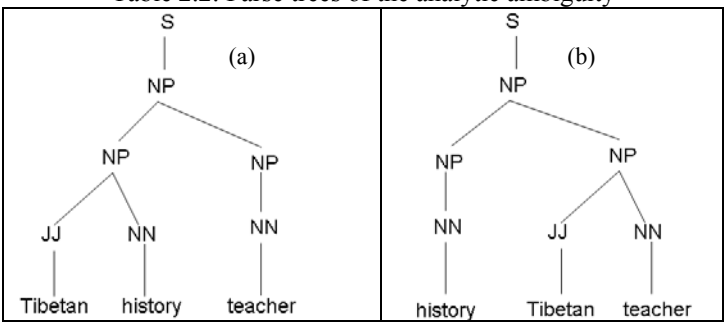
Tibetan history teacher

This sentence is ambiguous because it can be interpreted as:

(Tibetan history) teacher
history (Tibetan teacher)

The parse trees for the two different interpretations are respectively showed in Table 2.2(a) and in Table 2.2(b). The Natural Language sentence "*Tibetan history teacher"* presents an analytic ambiguity and there are two different parse trees for the given sentence, as Table 2.2 shows.
In particular, the two trees show that *Tibetan* and *history* are part of the same *noun phrase* in the tree of Table 2.2. (a), and in the second tree *Tibetan* and *teacher* are part of the same *noun phrase*.

Table 2.2: Parse trees of the analytic ambiguity



2.2.2.2 Attachment ambiguity

An attachment ambiguity occurs when a set of words in a sentence can be legally attached to two different parts of the sentence. It, similarly to the analytic ambiguity is due to different syntactic structures of the sentence. However in this ambiguity a conjunction attaches the preposition to different parts of the sentence. For example let be given the following sentence:

the police shot the rioters with guns

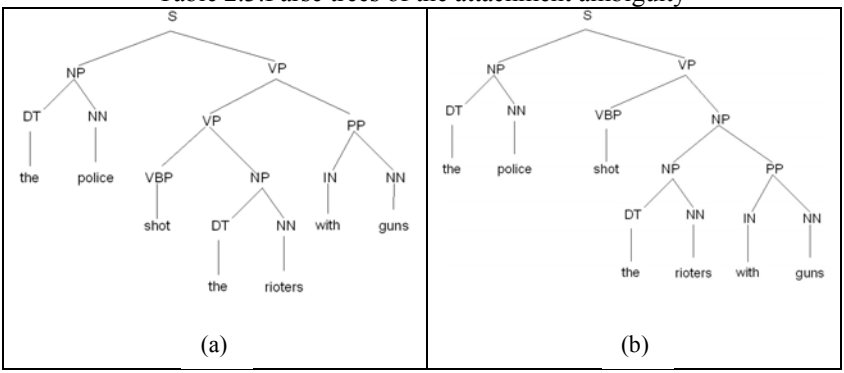
Two possible interpretations are:

the police ***shot*** *(the rioters)* ***with guns***

the police shot (the ***rioters with guns****)*

The first interpretation considers *with guns* attached to the verb *shot* (Table 2.3(a)) while a second attaches *with guns* to *the rioters* (Table 2.3(b)).

Table 2.3:Parse trees of the attachment ambiguity



2.2.2.3 Coordination ambiguity

Coordination ambiguity arises when in a sentence there is more than one conjunction or when there is coordination with an adjective.

An example of this ambiguity is given by the following sentence where different set of the sentence can be conjoined by a conjunction *and*.

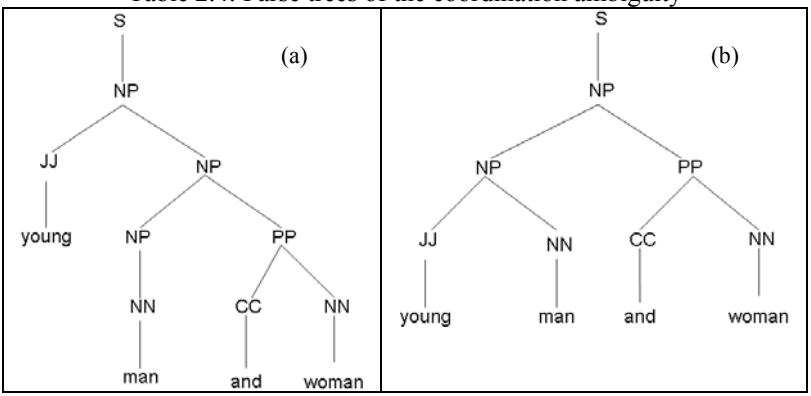
young man and woman

In this case the two possible interpretations are:

young (*man and woman*)
(young man) and (woman)

Table 2.4 (a) and Table 2.4 (b) respectively show the syntax tree for the given interpretations.

Table 2.4: Parse trees of the coordination ambiguity



2.2.2.4 Gap ambiguity

Gap ambiguity can arise if an element of the sentence is omitted. A gap ambiguity is defined by an ellipsis in a sentence that implies omission of a lexically or syntactically necessary constituent.
For example let be given the following sentence:

Perot knows a richer man than Trump

This sentence defines an ellipsis, i.e. the omission of a lexically necessary structural constituent after *Trump*.
This omission implies two different meanings of this sentence. For example we could have:

Perot knows a man who is richer than Trump is
Perot knows a man who is richer than any man Trump knows

In this example the first meaning has no ellipsis, and the second has an ellipsis, which is the implied *knows* coming just after *Trump* [BKK01] and showed in figure 2.1.

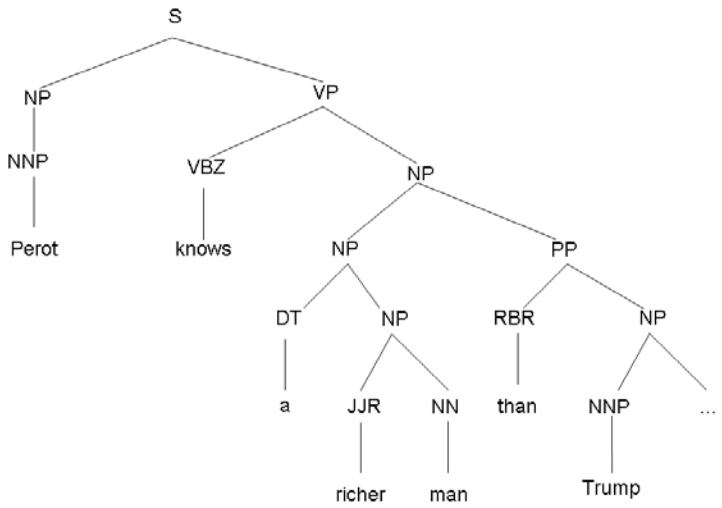


Figure 2.1: Gap Ambiguity

2.2.3 Semantic Ambiguities

Semantic ambiguities appear when a sentence has more than one interpretation even if neither lexical nor syntactic ambiguities appear in the sentence. It could be produced by coordination ambiguity, scope ambiguity and referential ambiguity. Coordination ambiguities have been presented in section 2.2.2.3.

Scope ambiguity appears when there are quantifiers that can define different relation in the sentence. Let be given the sentence:

all linguistics prefer a theory

even if it is neither lexically nor syntactically ambiguous, the sentence can have more than one meaning. Indeed, the scope of *a* is prevalent respect to the scope of *all*, then the sentence means that:

all linguists love the same one theory

When the scope of *all* is prevalent respect to the scope of *a*, the sentence meaning is:

Each linguist loves a perhaps different, theory

Referential ambiguity is also classified as pragmatic ambiguity; it refers to the meaning of the word and the context, and in particular when there is an anaphora and it can be referred to two different words. Let be given for example the anaphora *they* in the sentence:

the trucks shall treat the roads before they freeze.

2.2.4 Pragmatic ambiguities

Pragmatic ambiguities appears when a sentence can have more than one meaning in the context and this class can be divided in: *referential ambiguity* and *deictic ambiguity* that appears when pronouns and other grammatical elements have more than one reference in the discourse context.

2.3 Classification of Visual Language Ambiguities

The classification presented for Natural Language ambiguities was extended to the ambiguities for Visual Languages.
People use visual notations to model objects and handwritten words, to express relationships between them and to formulate sentences. Communication using visual elements produced the development of Visual Languages. A Visual Language (VL) is based on simple visual elements (glyphs/graphemes) that form the Characteristic Structure (CS). A Characteristic Structure is the set "of image pixels, which form functional or perceptual units" for the user. They can be grouped to form structured Visual Sentences.

According to [BCL95], a Visual Language is a set of Visual Sentences, and a Visual Sentence is given by an image, its description, its interpretation function and its materialization function. The interpretation function is the function that interprets the image (visual input). The materialization function provides a visual feedback on an image (taking into account of the changes due to the previous input).
The image materializes the meaning intended by the sender and must be interpreted by the receiver.
The image connected with a Visual Sentence can have more than one interpretation for the user and for the system and for this reason it can produce ambiguities. One reason of ambiguities is that a unique space is used to convey different kinds of information. Moreover the user gives his/her own semantics to information and the user's actions do not correspond to his/her intentions. So ambiguities are found at different levels: ambiguities can arise from both, by the user and by the system side. This section only deals with ambiguities by the computer side perspective.
Literature on Visual Languages provides some different definitions of ambiguity. [Fut99] distinguishes between lexical and syntactic ambiguities, with the meaning previously introduced for NL. In [FaA00] taxonomy of ambiguities in Visual GIS query languages is defined. This taxonomy has been developed using the formalism to describe Visual Languages introduced in [BCM99]. It considers ambiguities describing how the System materializes and interprets a Visual Sentence according to the user's actions performed to formulate it.
According to [MHA00], an "ambiguity arises when there is more than one possible way to interpret the user's input". They focused their attention to manage ambiguities by means of the dialog between the user and the System.
When the image (*i*) connected with a Visual Sentence is not able to exactly express the user's intentions (is not able to be completely faithful to his/her intentions) the system can produce an erroneous interpretation, i.e. 1-n associations matching between the image *i* and its description. Ambiguities are generally produced by: 1) the cited one-to-many relationships, 2) imprecision introduced by the user on the Visual Sentence. In the first case an image *i* can assume

more than one meaning. The second kind of ambiguity is connected with incorrect/imprecise information, which does not permit to the System to interpret, or univocally interpret the Visual Sentence.
This chapter gives a classification of ambiguities for Visual Languages that groups *Lexical, Syntactic ambiguities* of the language and *Imprecision ambiguities*, i.e. ambiguities due to more than one interpretation of the Visual Sentence by the System according to the imprecision introduced by the user's input. These latter ambiguities can generate lexical and syntactic ambiguities.

2.3.1 Lexical ambiguities

Languages ambiguities are grouped in Lexical and Syntactic ambiguities of the language. This classification [Fut99] is generally valid also for Visual Languages.
Lexical ambiguity is also known as semantic ambiguity and it involves alternate senses for simple items, so it appears when a Characteristic Structure or a relationship between two of them has more than one generally accepted meaning. Lexical ambiguities can be due to:

1) a Characteristic Structure having more than one generally accepted meaning [Fut99],
2) a relationship between two Characteristic Structures having more than one generally accepted meaning,
3) the unclear user's focus for the Visual Sentence; this is the case of the Target ambiguity introduced in [MHA00].

Below three examples of the introduced sub-classes of lexical ambiguities are provided.
As an example of a Characteristic Structure, which has more than one generally accepted meaning, it is considered a rectangular shape. It can (for example) have the following accepted meanings: a) entity, b) external agent, c) action.
So it needs to match the shape in a grammar in order to give it only one meaning to it within a restricted context. In particular, the rectangle given by a bold line in Figure 2.2A is an entity in the

context of an Entity-Relationship model; it is an “external agent” in the VL of data flow diagram (Figure 2.2B) and it is an “action” in the VL for flow chart (Figure 2.2C).

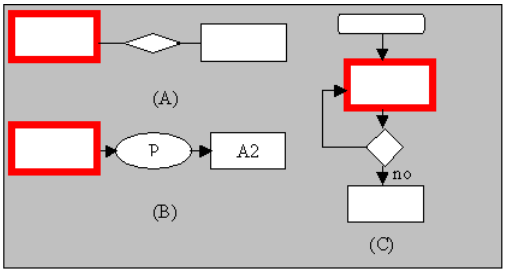


Figure 2.2: A characteristic structure with more than one accepted meaning

Let be now considered a lexical ambiguity due to different meanings given to a relationship between two Characteristic Structures. As an example of a relationship between Characteristic Structures with more than one generally accepted meaning let be given a Visual Sentence involving three different Characteristic Structures: one line, one rectangle and one oval. In particular, if the Visual Sentence has to specify that the line crosses the rectangle, the rectangle overlaps the oval and no relationships between the oval and the line are given, then the use of the bi-dimensional space (Figure 2.3) materializes a not required spatial relationship that can have different meanings.

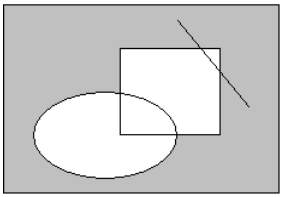


Figure 2.3: Ambiguity due to not required spatial relationships between two of the three Characteristic Structures, forced in a bi-dimensional space

The third class of Lexical ambiguity is the *Target ambiguity*. It is introduced when the target of the user's input is unclear. For instance, in Figure 2.4A it is unclear what is the Characteristic Structure on which the user focuses his/her attention. Then, the system could give an ambiguous interpretation of the user's information goal. For this reason could be necessary to make explicit the information goal. It could be specified (for example) using an arrow that points the target (Figure 2.4B).

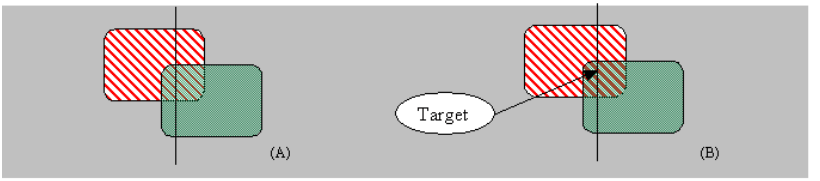


Figure 2.4: Target ambiguity

Therefore, a target ambiguity is defined as the ambiguity introduced both: 1) when the target of the user's input is unclear and/or, 2) when the target of the user's query is unclear.

2.3.2 Syntactic ambiguities

Syntactic ambiguity is also known as structural ambiguity and it appears when the role, which an element of the language plays in a sentence, is unclear. There are three subclasses of syntactic ambiguity: *analytic*, *attachment, gaps, segmentation and occlusion* ambiguities.

2.3.2.1 Analytic ambiguity

The first class of the syntactic ambiguity is the Analytic ambiguity. It appears when the categorization of a structure is itself in doubt according to its role in the Visual Sentence. For instance in Figure 2.5 the analytic ambiguity is due to the fact that the oval shape

labelled “relation node” in the lower left corner of the Figure 2.5 looks like an element of the diagram but it does not belong to it because it is an element of the legend.

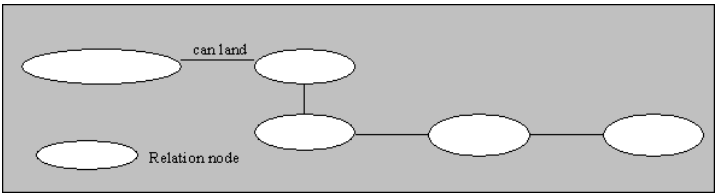


Figure 2.5:Analytic ambiguity

2.3.2.2 Attachment ambiguity

The second subclass of syntactic ambiguity is the Attachment. It consists of the ambiguity that arises in the matching between a Characteristic Structure and the text item, which label it. For instance, the parser could have the same difficulties to associate text labels and Characteristic Structures in Figure 2.6. In fact, the “A” label might refer to the line or the region identified by the texture ; the “B” label might to refer to the region identified by the texture or the region identified by the texture ; the “C” label might refer to the region identified by the texture or to the line.

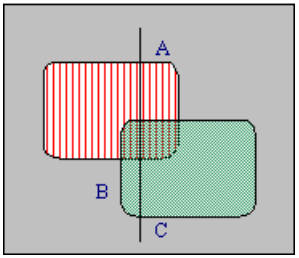


Figure 2.6:Attachment ambiguity

2.3.2.3 Gap Ambiguity

The third subclass (gaps ambiguity) appears when an element of the sentence is omitted, producing a gap in the information. Figure 2.7 shows an example of this event. In fact, the tick marks are not all labelled and the parser can have some difficulties for the interpretation because it has to associate a value and a meaning to each one of the tick marks, independently from the fact that this value is implicitly or explicitly given.

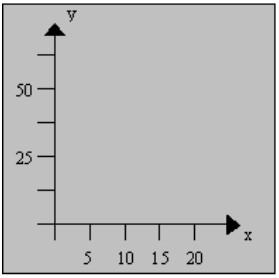


Figure 2.7: Gap ambiguity

Gap ambiguity is due to lacking information on at least one Characteristic Structure. Let us suppose this information is generally contained in one attribute of the description of the Characteristic Structure and this attribute has an empty value for lacking information.

2.3.2.4 Segmentation ambiguity and Occlusion ambiguity

This section introduces the Segmentation and Occlusion ambiguities as discussed in [Fut99] and in [MHA00] too. Segmentation ambiguity is considered as a kind of syntactic ambiguity, or more specifically a kind of analytic ambiguity and it has been more evidently observed when diagrammatic notations are used. Diagrams are replete with a variety of ambiguities, as detailed by Futrelle in [Fut99]. Many of these ambiguities are subtle and difficult to solve.

Since any Image could be formed by a number of subparts, segmentation ambiguity appears when the user considers a portion of a single line as an entity itself.
The diagram of Figure 2.8 gives an example of Segmentation ambiguity. The short lines in the lower left corner of the diagram could represent tick marks or the ends of the x and y axis lines. Then, the parser could have difficulties to understand if these short lines are entities themselves or if they are part of the long axis lines.

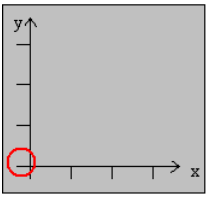


Figure 2.8: Segmentation ambiguity

Occlusion is given by the overlap of different elements and one of them does not allow the user to see part of the other one. In this case it could be difficult to understand if these elements are separately used or they are the components of a complex Characteristic Structure.
For instance, to create a table with a box that contains the name of the table, as shown in Figure 2.9, a simple way is to create two different rectangular shapes and one overlaps the other so that only the small visible rectangle that contains the table name is relevant.

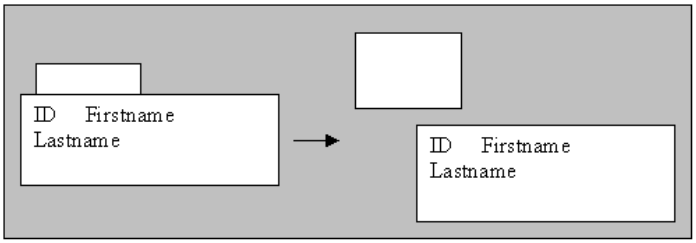


Figure 2.9: Occlusion ambiguity

That is, two rectangles could be recognized as a unique complex Characteristic Structure.

The sketch-based languages can produce some of the previously introduced syntactic ambiguities.
For example, the sketch in Figure 2.10A can be ambiguous. In fact, due to the segmentation and to the occlusion ambiguities, the sketch in Figure 2.10A has various interpretations according with the parsed symbols; two of these interpretations are shown in Figure 2.10B and Figure 2.10C. Figure 2.10B considers the sketch as formed by ten graphical components: four closed shapes (A, B, C, D) and six polylines (1-6). Figure 2.10C considers the sketch as formed by seven graphical components: five closed shapes (A, B, C, D, E) and just two polylines (1-2). Obviously, because the sketch is a diagram, the correct interpretation is Figure 2.10B. However, changing the application domain can lead to changes in the correct interpretation. So, if the sketch in Figure 2.10A is not a diagram but a map, Figure 2.10C is probably the correct interpretation.

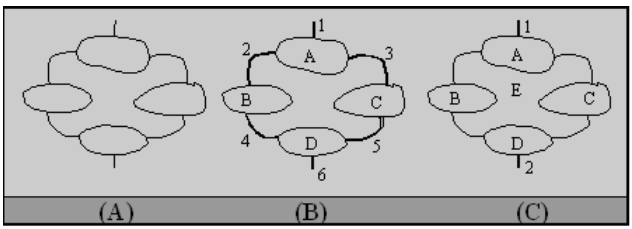


Figure 2.10:A sketch and some possible interpretations

2.3.3 Ambiguities due to imprecision produced by the Human Computer Interaction behaviour

Imprecision and noises are introduced by users in their interaction behaviour (drawing) and/or by tools and sensors. For this reason signs on the sketch may not be univocally parsed. That is, for example, in the interaction by sketch. The information provided by

the sketch may thus be not sufficient to identify a unique interpretation.
Imprecision due to the user's interaction behaviour can produce some difficulties in the recognition of characteristics that identify Characteristic Structures belonging to the Visual Sentence. For example, when a user sketches a rectangle that is not precise enough, the system could interpret it as an ellipse. This case introduces the *shapes ambiguity* class.
An example of ambiguity due to the imprecision characterizing the hand drawing activity is shown in Figure 2.11A. In fact, it could represent both: 1) a rectangle (Figure 2.11.B), or 2) an oval Figure 2.11.C.

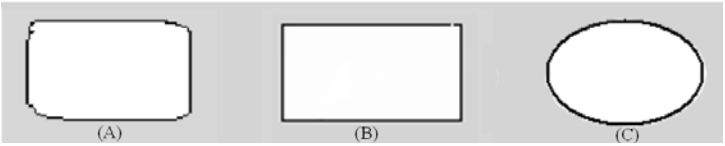


Figure 2.11: A sketch that defines a shape ambiguity

Another class of ambiguity due to the imprecision introduced during the interaction behaviour is the *constraints* (properties) *ambiguity*. It concerns the evaluation of unary predicates that inspect shape properties. An example is the classification of a line into vertical, oblique or horizontal line. Considering lines represented in Figure 2.12, which of those the system can classify as horizontal, vertical or oblique?

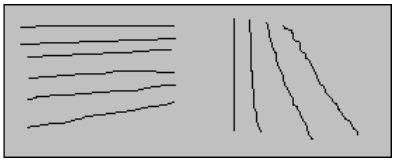


Figure 2.12: Constraints ambiguity

The third class of ambiguity, *relationships' ambiguity*, is usually observed in sketch-based languages. It is due to the drawing

imprecision when many shapes are closely located and it is difficult to know the relationships between each couple of shapes.
An example, shown in Figure 2.13, represents two regions that could be in a touch or overlap relationship.

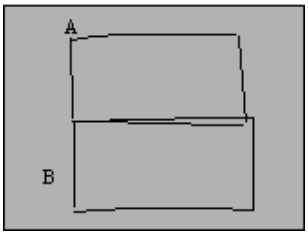


Figure 2.13: Relationships' ambiguity

2.4 Conclusions and discussion Modal Ambiguities to Multimodal Ambiguities

This chapter, describing how literature faces the problem of ambiguities for languages, focuses the discussion on the goal of the present thesis, i.e. dealing with the language ambiguities. In particular, this chapter has described some of the most relevant and commonly identified classes of ambiguities (such as lexical, syntactic, semantic, etc.) for NL, their features and how some authors [Fut99] propose to adopt them for Visual Languages. Moreover, a class of *ambiguity due to imprecision* has been added to classification proposed by the literature for VLs. This class of ambiguity has been introduced embedding, at an abstract level, imprecision connected with visual representations.
Starting from the described NLs and VLs classification of ambiguities, this thesis describes in Chapter 3 the analysis of ambiguities that can appear during multimodal interaction.
NLs and VLs ambiguity classes characterize many modal communication and interaction such as speech, textual, sketch, etc..
In the next chapter, modal classifications are extended taking into account of issues that appears during the combination of modal

inputs into a multimodal one introducing the Multimodal Language and Multimodal Sentence notions.

In fact, ambiguities can arise considering multimodal sentences both, if modal ambiguities are propagated at multimodal level, and/or if elements connected with each modality are correctly interpreted, but information referred to each modality can be not coherent at the syntactic or semantic level. The idea is to consider a multimodal sentence as a whole and not only the sum of its component according to a holistic point of view.

Chapter 3 will provide a detailed description of the analysis of multimodal ambiguities and their classification.

Chapter 3

Classifying Multimodal Ambiguities

3.1 Introduction

The interpretation of multimodal inputs is a very important process because it identifies the meanings of the user's inputs according to her/his communication goal. When a gap appears between the user's goal conveyed by an input message and how the system interprets it, errors or ambiguities can arise.
This chapter, according to the thesis goal of focusing on problems related to the interpretation process of multimodal inputs [LDR04], provides basic definitions (e.g. for *Multimodal Element*, *Multimodal Sentence* and *Multimodal Language*) used to analyse and describe ambiguities features.
Actors involved in a communication process exchange among them *messages*, i.e. the objects formed by one or more pieces of information (the *elements*). Considering the communication process by the linguistic point of view messages can be expressed by sentences. In particular, as this thesis deals with multimodal communication, the notion of *Multimodal Sentence* and *Multimodal Language* are given (see section 3.3.2).
The adoption of a linguistic point of view has implied the use of a grammar-based approach (see the definition of *Multimodal Grammar* in section 3.3.1) for fusing multimodal inputs and interpreting *Multimodal Sentences* that they compose. Indeed, as

Figure 3.1 shows, the modal parsers recognise the different modal inputs that are opportunely integrated (fused) and interpreted using a Multimodal Grammar.

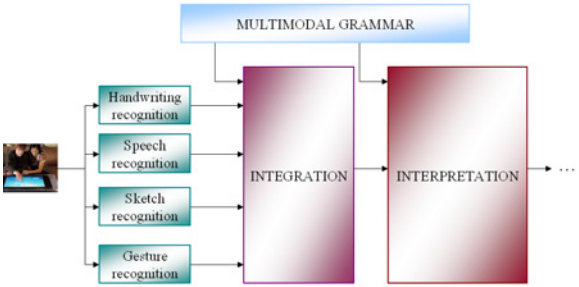


Figure 3.1: The process from multimodal input to Multimodal Sentence interpretation

Similarly to the interpretation of NL and VLs sentences, the interpretation of a *Multimodal Sentence* can produce more than one result. This could produce the arising of ambiguities described in Chapter 2 for Natural Language and Visual Languages, as well as the arising of ambiguities coming from the different modalities combination in a *Multimodal Sentence*. Ambiguities can be due to both: a) the incorrect interpretation of at least one or more separate interpretations of a modal input, and b) the incorrect interpretation deriving from a combination of information belonging to the involved modalities that are not coherent at the semantic level.
A correct interpretation, as it will be explained in section 3.3.1.2, can be obtained considering constraints about the syntax, the context and temporal relations.
First of all it is important to capture the main features to consider for correctly interpreting multimodal inputs. In particular, starting from the analysis of NL and VLs ambiguities given in Chapter 2, this chapter proposes a classification of ambiguities for *Multimodal Languages* according to the features identified to detect them.
Some meaningful examples of multimodal ambiguities and their formal description using the Multimodal Attribute Grammar (MAG), which is an extension of the Attribute grammars [Knu68] combined with the Linear Logic [Gir87], is here provided. MAG has been adopted as it highlights the multidimensionality of

multimodal interaction considering multimodal inputs as a whole. Some meaningful examples of multimodal ambiguities and how they can be described and detected are given.
The chapter ends with a discussion that summarises the Multimodal Ambiguity classes, their features and rules that can be used to intercept them.

3.2 From Modal Ambiguities to Multimodal Ambiguities

Starting from the analysis of modal ambiguities, provided in the Chapter 2, this chapter presents a classification of ambiguities arising from different modalities during multimodal interaction.
Ambiguities of a *Multimodal Sentence* can be due to input ambiguities propagated at multimodal level because unsolved, and/or to incoherence at the syntactic (having more than one syntax-tree) or semantic level (e.g. redundant inputs having different meanings).
The basic idea is that all the ambiguities presented by Chapter 2 can propagate themselves at the multimodal level. Therefore, this chapter extends the dissertation on ambiguities at multimodal level, according to the general classification of ambiguities in: *Semantic ambiguities* that take into account issues connected with the meaning of the *Multimodal Sentence* and its components (the elements), and *Syntactic ambiguities* that are connected with the structure of the *Multimodal Sentence*.

Semantic ambiguities are connected with the semantic of the elements of the language and their combination. This kind of ambiguities arises when:

- one element has more than one generally accepted meaning (*Lexical ambiguities*);
- two different elements of the *Multimodal Sentence* have the same syntactic role but they refer to two different concepts by different modalities (*Temporal-Semantic ambiguities*);
- the focus of the user is not clear (*Target ambiguities*).

Let be given, for example, a circle drawn by a user. It is an example of *lexical ambiguity* as the language implies more meanings such as "plane curve everywhere equidistant from a given fixed point" or the "*o* character".
Obviously, the example of lexical ambiguity introduced for NL (Chapter 2) involving the term "bank" is an ambiguity for *Multimodal Languages* too, if it propagates itself at multimodal level. It can be interpreted as a "shore of the river" or as a "financial institution" [Buv96].
An example of *Temporal-Semantic* ambiguity appears when a *Multimodal Sentence* combines two elements defined by two different modalities, such as when a user says "river" using the speech modality and she/he draws the shape for a lake by sketch modality. As the two inputs are connected with two different concepts, it is not possible to define the correct interpretation.
An example of *target ambiguity* appears when it is impossible to identify the user's focus in the *Multimodal Sentence*, such as when the user asks to the system information about a restaurant using the speech modality and meanwhile, she/he selects by freehand sketch contemporaneously two different restaurants. In this case, it is impossible to identify on which one of the restaurants was the user focus.

Syntactic ambiguities are connected with the structure of the *Multimodal Sentence* and appear when alternative structures for the *Multimodal Sentence* can be generated during the interpretation process. These ambiguities arise when the role that an element of the language plays during interaction is not univocally defined and the elements of a *Multimodal Sentence* can be syntactically combined in more than one way.
Syntactic ambiguities can arise:

- when an element of the *Multimodal Sentence* is omitted (*Gap ambiguities*);
- when the categorization of the element is itself not univocally defined (*Analytic ambiguities*);

- when an element of the sentence can be legally attached to two different parts of the sentence (*Attachment ambiguities*).

In particular, *Gap ambiguities* are common in diagrams, and they can be due to omitted labels.
An example of *Analytic ambiguity* [Hir87] is given by the sentence *Tibetan history teacher*. As defined in Chapter 2 this sentence can be interpreted as *(Tibetan history) teacher,* i.e. the teacher of Tibetan history, or *history (Tibetan teacher).*
Considering *Attachment ambiguity*, it appears if an element of the sentence can be attached to two different parts of the sentence defining two different structures of the sentence leading to two different interpretations. An example of Attachment ambiguity is given when the user says by speech *he wrote a letter to the woman with the pen,* while using gesture she/he indicates a man. This ambiguity is produced by the impossibility to decide between these cases: 1) in the first interpretation *with the pen* is attached to the verb *wrote*; 2) in the second interpretation *with the pen* is attached to the element *woman*. These two interpretations associate different meanings to the sentence; in the first case, the man is using a pen to write a letter, while in the second one the woman has the pen.
Similar considerations for Segmentation and Occlusion ambiguities (discussed in Chapter 2) can be done, as modal ambiguities that propagate themselves at multimodal level.
The next section introduces notions of *Multimodal Grammar*, *Multimodal Sentence* and *Multimodal Language*, which are used in the other sections of this chapter to define the classes of multimodal ambiguities

3.3 A Grammatical/Logical approach to detect ambiguities: basic concepts

This section defines the main concepts, which are used to classify and represent multimodal ambiguities features, according to the hypothesis that this work uses a linguistic approach, and then it

gives the notion of *Multimodal Language* for dealing multimodal dialogue.
In particular, the treatment of syntax and semantics of the *Multimodal Language* is dealt using a Multimodal Attribute Grammar combined with the Linear Logic [Gir87] in order to formalize multimodal inputs, to represent *Multimodal Sentences* and to detect multimodal ambiguities.
The Linear Logic extends the Classical Logic introducing the notion of *resource* and the concept of *formulas as resource*. Girard [Gir87] stated "a completely new approach to the whole area between constructive logics and computer science is initiated". Linear Logic is conceived as logic of resources and actions, and it overcomes the problem of the predicate-based encoding of Classical First Order Logic. In fact, the Linear Logic relaxes the monotonicity constraints of the Classical Logic and it is able to model changes in the time. Linear Logic supports general forms of reasoning and it is used to formalize different things such as Petri nets and functional language implementation. These features of Linear Logic satisfy some needs deeply connected with multimodality and its characteristic of changing over the time. So the proposed approach uses Linear Logic combining it with a Multimodal Attribute Grammar for dealing *Multimodal Sentences* and detecting multimodal ambiguities that appear during the interpretation process.

The following sections will provide the definitions of: *Multimodal Grammar* (see section 3.3.1), *Multimodal Sentence and Multimodal Language* (see section 3.3.2).

3.3.1 Definition of the Multimodal Grammar

This work uses a context-free grammar because it is suitable to describe the syntax of natural language, as each *Multimodal Sentence* has its interpretation by NL.
The Multimodal Attribute Grammar, which is a context-free grammar, is an advanced attribute-based grammar [MMW98] allowing to compute derived attributes of non-terminal symbols using computation embedded into the grammar productions.

So, the following definition presents the grammar used in this approach

Def. 3.1–*The multimodal grammar is a triple (G,A,R) defined by:*

- *G that is s context-free grammar (T,N,P,S) with T as set of terminal symbols, N as set of non-terminal symbols, P as set of production rules and S ∈ N as start symbol;*
- *A is the collection of attributes of terminal and non-terminal symbols;*
- *R is a collection of semantic rules.* □

3.3.1.1 Definition of Terminal Elements of the Multimodal Grammar

Terminal symbols in the Multimodal Attribute Grammar are the building units of the *Multimodal Language*. These symbols (*terminal elements* in the following) of the *Multimodal Language* include information about: the modality used to specify the elements; the representation of the elements in the specific modality; the temporal intervals connected with the elements; the syntactic roles that the elements play in the *Multimodal Sentence*; the semantic definition of the elements considering their representation according to the modality. In particular the semantic meanings of the elements are given considering a domain ontology that provides a conceptual structure of the context.
Therefore, a *terminal element* E^i is defined as:

Def. 3.2- *A terminal element* E^i *is a 5-pla* (E^i_{mod}, E^i_{repr}, E^i_{time}, E^i_{role}, $E^i_{concept}$), *with:*

- $E^i_{mod:}$ *that defines the modality used to create the element* E^i
- E^i_{repr}*: that defines the representation of the element* E_i *in the specific modality,*
- E^i_{time}*: that defines the temporal interval connected with the element* E_i,

- E^i_{role}: *the syntactic role that the element* E_i *plays in the Multimodal Sentence,*
- $E^i_{concept}$: *that specifies of the concept name referred to the conceptual structure of the context.* □

For example, let be given the concept "river" and the multimodal input that is defined by the sketch and speech modalities. The representation corresponding to the sketch modality is (), while the representation related to the speech modality is the signal connected with the word river ( "river").
The element "*river*" defined by the sketch modality is characterized by the following n-pla:

- E^i_{mod}= (sketch)
- E^i_{repr}= ( "")
- E^i_{time}= (5, 14)
- E^i_{role} = (nn)
- $E^i_{concept}$=(river)

The same element "*river*", defined by the speech modality, differs from the previous element defined by sketch in the attribute connected with the modality, attribute about the representation and attribute connected with the temporal interval.
*In particular, the element "*river*" defined using the speech modality is characterized by the following 5-pla:*

- E^i_{mod}= (speech)
- E^i_{repr}= (" "river" ")
- E^i_{time}= (3, 5)
- E^i_{role} = (nn)
- $E^i_{concept}$=(river)

3.3.1.2 Definition of production rules of the Multimodal Grammar

The production rules of the multimodal grammar can be divided in: rules that refer to the construction of the *syntax of the grammar* P^g*;* rules about the *context* P^c; and *temporal rules* P^t.
So the set of *production rules* P is:

$$P = \{P^g, P^c, P^t\}$$

For defining the production rules of our approach, that will treat both syntax and semantics of the *Multimodal Language*, the Linear Logic has been chosen.
An exhaustive description of Linear Logic is out of scope of this dissertation and the Multiplicative Intuitionist fragment of Linear Logic, which uses the multiplicative connective "$\otimes$" (conjunction of hypotheses) and the linear implication "$\cdots\circ$", is sufficient for our purpose.
In particular this work analyses the *Multimodal Sentence* and its interpretation in natural language.
So according to its scope this work uses the production rules of the natural language for defining the production *rules of the syntax of the grammar* P^g. These rules are expressed in Linear Logic and some examples of these rules are defined below:

$$s \cdots \circ np \otimes vp$$
$$np \cdots \circ dt \otimes nn \otimes pp$$
$$np \cdots \circ np \otimes pp$$
$$np \cdots \circ nn \otimes pp$$
$$np \cdots \circ det \otimes n$$
$$np \cdots \circ jj \otimes n$$
$$np \cdots \circ nn$$
$$vp \cdots \circ vbz \otimes np$$
$$vp \cdots \circ vbz \otimes np \otimes pp$$
$$pp \cdots \circ in \otimes np$$

When considering the context, it implies the setting of the elements and it denotes the relations among them. Therefore *context rules* allow identifying the correct interpretation of an element and relations among elements. For example "monitor is *part-of* a computer" and "Tiber is *instance-of* river", and relations among actors and actions, as for the actor *speaker* an appropriate action is *speak*. In detail, the knowledge about the context includes synonyms, generalizations, specializations, definitions, different lexical categories of the element and related terms.
In particular, context rules define what is true in the context of interaction and they allow deducing which is the context of interaction.

For example, let us suppose that the user draws the sketch in the Figure 3.2 using the sketch modality, and she/he says the word "Tiber"("Tiber").

Figure 3.2: User input by sketch

The drawing in Figure 3.2 can be interpreted both as a river and a street if they are both meanings of the Language. Considering the speech input it can be interpreted as the river Tiber and therefore the context rule "*Tiber is instance-of river*". Analysing only the sketch modalities it is not possible to define if the correct interpretation is river or street, while combining both modalities, it is possible to detect that it is not true that "*Tiber is instance-of street*" but it is true that "*Tiber is instance-of river*". Therefore, considering the evaluation of the context rules and information and representations about the used modalities it is possible to interpret the input as "Tiber river", as formally defined below:

((" ") ⊗ (sketch)) ⊗ (("Tiber") ⊗ (speech))
⊗ ("Tiber is instance-of river") ⋯∘ (river) ⊗ (Tiber)

The general rule about the context is the following:

Def. 3.3- $\forall\ E^i, E^j$:: *elements* $\exists\ P^k \in P^c$ *s.t.* $\{((P^k$ *is true)* $\wedge\ ((E^i_{repr} \otimes E^i_{mod}) \otimes (E^j_{repr} \otimes E^j_{mod}) \otimes P^k)) \cdots \circ E^i_{concept} \otimes E^j_{concept}\}$ □

In detail this formal notation defines that it is possible to obtain the correct concepts, which representations of elements in the specific modalities refer to, using context rules connected with the defined elements.

Finally *temporal rules* impose constraints on temporal intervals of elements. These rules establish how to take into account elements whose temporal intervals are contained in a defined temporal slot.

Temporal relations among elements expressed by different modalities, introduced in Chapter 1 (see section 1.3.1), play an important role for the specification of temporal rules connected with the cooperation classes among elements for the different modalities.

In particular, starting from the relation "CloseBy" (see section 1.3.1), it is possible to analyse which is the cooperation class between modalities: redundancy if the same concept is defined using both the modalities; complementarity if the two modalities define two different concepts that define a further concept; and concurrency if independent concepts are defined using the different modalities and overlap in time. It is possible to formalize the cooperation classes between modalities are formalised as follows:

Def. 3.4.- E^i *and* E^j *are redundant if* $\exists\ E^i, E^j$:: *elements where:* $((E^i_{time}\ CloseBy\ E^j_{time}) \otimes (E^i_{concept} \equiv E^j_{concept}))$ □

Def. 3.5- E^i *and* E^j *are complementary if* $\exists\ E^i, E^j, E^k$:: *elements,* $P^i \in P^c$:: *rule where* $((E^i_{time}\ CloseBy\ E^j_{time}) \otimes (E^i_{concept} \neq E^j_{concept}) \otimes (E^i_{concept} \otimes E^j_{concept} \otimes P \cdots \circ E^k_{concept}))$ □

Def. 3.6- E^i *and* E^j *are concurrent if* $\exists\ E^i, E^j, E^k$:: *elements,* $\neg\exists\ P^i \in P^c$:: *rule where* $((E^i_{time}\ CloseBy\ E^j_{time}) \otimes (E^i_{concept} \neq E^j_{concept}) \otimes (E^i_{concept} \otimes E^j_{concept} \otimes$

$$P^i \cdots \circ E^k_{concept}))$$ □

Let be given an example of multimodal input using speech and sketch modalities. The user says by speech:

"the Tiber is near a lake"

while she/he draws the sketch ( ) as described in the following figure:

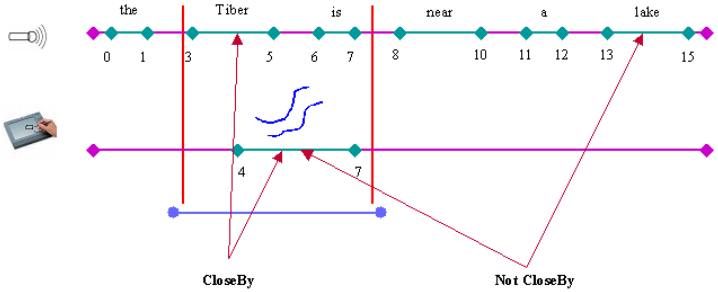


Figure 3.3: CloseBy relation

Considering the relation *CloseBy* the drawing ( ) is associated with the speech element "*Tiber*" and not with the speech element "*lake*" (Figure 3.3). The drawing ( ) and the speech element "*Tiber*" are analysed and the system defines that they are complementary because the element "*Tiber*" is an instance of the element "*river*" defined by the sketch modality. So the specialization relation defined by the context rules appears between these two elements. In this case the *Multimodal Sentence* is univocally interpreted.

The interpretation of the *Multimodal Sentence* is ambiguous when there is more than one combination of terminal elements that imply different meaning of the sentence and/or when one terminal element of the sentence is not univocally interpreted. In particular, the sentence is ambiguous if elements, which compose it, can be combined in more than one manner. Moreover, a terminal element

is univocally interpreted if the association with an element of the context is unique.

3.3.2 Definitions of Multimodal Sentence and Multimodal Language

The syntax rules of the grammar (Pg), the context rules (Pc) and the temporal rules Pt provide the syntax structure (syntax-tree) and the semantic for the language.
An ambiguous multimodal input can be associated to more than one syntax-tree. All the syntax-trees of the multimodal input will be combined in a direct acyclic graph that is associated to the *Multimodal Sentence* (below defined). In this work it will be called *syntax-graph*, and it is the syntax structure for the *Multimodal Sentence*.
Each terminal node of the *syntax-graph* is an element (terminal element) of the Multimodal Attribute Grammar, and each terminal node includes information about the specific element as previously defined (Def. 3.7).

Def. 3.8- *A syntax-graph is a direct acyclic graph that combines all the syntax-trees of the Multimodal Sentence and it has terminal elements of the grammar as terminal nodes.* □

Let be W the Penn Treebank Tag set [MSM94]; a path on the syntax-graph can be defined as:

Def. 3.9- *A syntactic (u,v)-path is an ordered sequence of syntactic roles* $\{w_0=u, w_1, w_2,...w_j=v\}$ *such that* $w_i \in W$ □

Starting from the introduced concepts it is possible to define a *Multimodal Sentence*, where the *Multimodal Sentence* is the grammatical unit that contains terminal elements that form functional or perceptual units for the user.

Def. 3.10- *A Multimodal Sentence is a 4-pla*

MMS : (E, syntax-graph, d, int_mms)

where :

- *E is a set of elements E^i for i=1..n with n where n is the number of elements that compose the Multimodal Sentence*
- *Syntax-graph is a direct acyclic syntactic graph that has elements E^i as terminal node;*
- *d is the description that defines the meaning of the Multimodal Sentence;*
- *int_mms is the interpretation function that maps the syntax_graph into the description d:*

  int_mms: (syntax_graph) → d □

The definition of the *Multimodal Sentence* is used to define the *Multimodal Language*, which extends the definition of Visual Language given in [BCL95].

Def. 3.11– *A Multimodal Language is a set of Multimodal Sentences* □

For clarifying this definitions an example of Multimodal Sentence is provided.
Let be given the example where the user says:

"show this near lake"

while she/he draws the sketches () and () as described in the following figure, where different inputs in the Multimodal Sentence are represented according to their temporal relations.

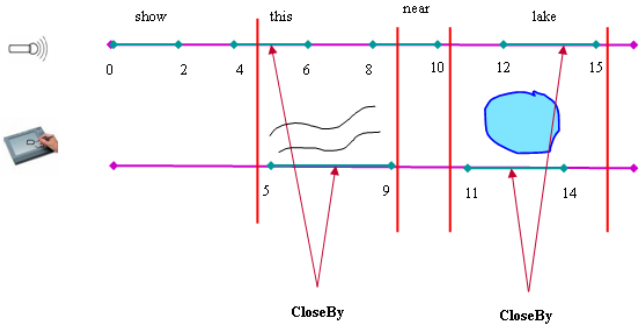


Figure 3.4: Example of Multimodal Sentence with complementary and redundant elements

This sentence defines the following syntax-graph:

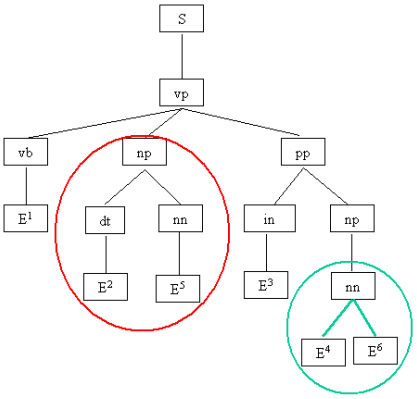


Figure 3.5: Syntax-graph of a Multimodal Sentence with complementary and redundant elements

The elements defined by the speech modality are:

- E^1 *is!* $(E^1_{mod}=speech) \otimes ! (E^1_{repr}=$  *"show")) ⊗ ! (*$E^1_{time}=(0,2)) \otimes ! (E^1_{concept}=(show)) \otimes !(E^1_{role}=(vb))$
- E^2 *is!* $(E^2_{mod}=speech) \otimes ! (E^2_{repr}=$  *"this")) ⊗ ! (*$E^2_{time}=(4,6)) \otimes ! (E^2_{concept}=(deictic)) \otimes !(E^2_{role}=(dt))$

- E^3 *is!* $(E^3_{mod}$*=speech)* $\otimes$ *!* $(E^3_{repr}$ =  *"near"))* $\otimes$ *!* $(E^3_{time}$*=(8,10))* $\otimes$ *!* $(E^3_{concept}$*=(near))* $\otimes$ *!*$(E^3_{role}$*=(in))*
- E^4 *is !* $(E^4_{mod}$*=speech)* $\otimes$ *!* $(E^4_{repr}$ =  *"lake"))* $\otimes$ *!* $(E^4_{time}$*=(12,15))* $\otimes$ *!* $(E^4_{concept}$*=(lake))* $\otimes$ *!*$(E^4_{role}$*=(nn))*

The elements defined by the sketch modality are the following:

- E^5 *is !* $(E^5_{mod}$*=sketch)* $\otimes$ *!* $(E^5_{repr}$ =  *"))* $\otimes$ *!* $(E^5_{time}$*=(5,9))* $\otimes$ *!* $(E^5_{concept}$*=(river))* $\otimes$ *!*$(E^5_{role}$*=(nn))*
- E^6 *is !* $(E^6_{mod}$*=sketch)* $\otimes$ *!* $(E^6_{repr}$ =  *"))* $\otimes$ *!* $(E^6_{time}$*=(11,14))* $\otimes$ *!* $(E^6_{concept}$*=(lake))* $\otimes$ *!*$(E^6_{role}$*=(nn))*

As E^4 and E^6 express the same common concept ("lake"), and are in the Close-by relation, they are terminal nodes having the same antecedent node (Figure 3.5), so they have to define redundant element. While terminal nodes, which have different antecedent nodes, are complementary (E^2 and E^5 in Figure 3.5).
The following section uses the given definitions, to analyse different multimodal ambiguities and to describe how they are grouped in classes. In particular, this work will specify rules that allow detecting when each class of multimodal ambiguities appears.

3.4 A Grammatical/Logical approach to detect ambiguities: Detection of Different Classes of ambiguities

This section presents features of multimodal ambiguities considering the communication process by the linguistic point of view.
Each *Multimodal Sentence* is processed and interpreted. Its interpretation is expressed by a sentence in natural language.
As described in the previous sections this approach is based on the Multimodal Attribute Grammar and rules about context and temporal relation among modalities expressed by Linear Logic.

This section shows how this approach allows identifying classes of multimodal ambiguities. A set of rules that allows detecting different classes of ambiguities will be provided.
Starting from the set P = {P^g, P^c, P^t} of production rules of the *Multimodal Language*, the following sections present an extension of these rules by a set of rules P_a that will allow detecting if the Multimodal Sentence is ambiguous and which class of ambiguities appears.
In particular it will be provided a classification of ambiguities connected with the properties of the language dividing them into:

- Semantic ambiguities:
  - lexical ambiguity
  - temporal-semantic ambiguity
  - target ambiguity
- Syntactic ambiguities:
  - gap ambiguity
  - analytic ambiguity
  - attachment ambiguity

In the following sections a detailed treatment of these classes of ambiguities is given.

3.4.1 Semantic ambiguities

The focus of this section is to define which rules allow to detect ambiguities connected with the semantics of the language: lexical, temporal-semantic and target ambiguities. They are presented in the sub-sections that follow.

3.4.1.1 Lexical Ambiguity

A lexical ambiguity is connected with the semantics of the elements of the language, and it appears when the meaning of an element is

not clearly identified. In order to clarify this kind of ambiguity let us suppose that using the speech modality the user says:

"show this in Rome"

while she/he simultaneously draws the sketch in Figure 3.6:

Figure 3.6: Sketch input of the user

Considering the drawing and the set of elements of the Multimodal Sentence and their meanings, the sketch in Figure 3.7 can be interpreted both as a river and a street. So the meaning of the user's input is not clearly identified.
In this case the Multimodal Sentence is:

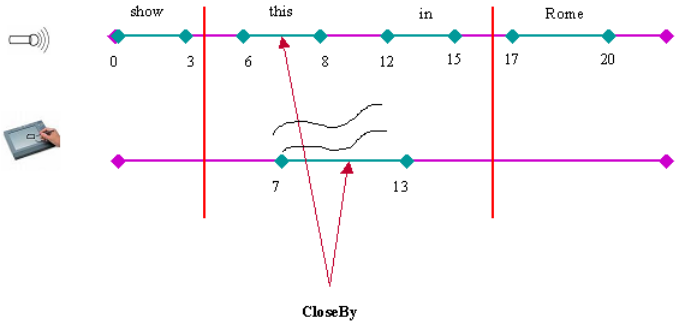


Figure 3.7: Example of input for Multimodal Sentence

The syntax-graph of the Multimodal Sentence of Figure 3.8 is:

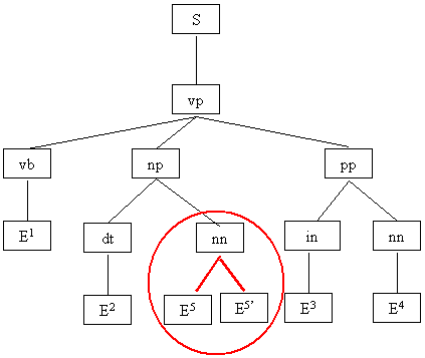


Figure 3.8: Syntax-graph of the user's Multimodal Sentence

In this example elements defined by the speech modality are:

- E^1 *is!* $(E^1_{mod}$*=speech)* $\otimes$ *!* $(E^1_{repr}$*="show"))* $\otimes$ *!* $(E^1_{time}$*=(0,2))* $\otimes$ *!* $(E^1_{concept}$*=(verb))* $\otimes$ *!*$(E^1_{role}$*=(vb))*
- E^2 *is!* $(E^3_{mod}$*=speech)* $\otimes$ *!* $(E^2_{repr}$ *="this"))* $\otimes$ *!* $(E^2_{time}$*=(5,7))* $\otimes$ *!* $(E^2_{concept}$*=(deictic))* $\otimes$ *!*$(E^2_{role}$*=(dt))*
- E^3 *is!* $(E^3_{mod}$*=speech)* $\otimes$ *!* $(E^3_{repr}$ *="in"))* $\otimes$ *!* $(E^3_{time}$*=(11,12))* $\otimes$ *!* $(E^3_{concept}$*=(adverb))* $\otimes$ *!*$(E^3_{role}$*=(in))*
- E^4 *is!* $(E^4_{mod}$*=speech)* $\otimes$ *!* $(E^4_{repr}$ *=" Rome"))* $\otimes$ *!* $(E^4_{time}$*=(15,18))* $\otimes$ *!* $(E^4_{concept}$*=(city))* $\otimes$ *!*$(E^4_{role}$*=(nn))*

while the drawing defined by the sketch input can be referred to two different concepts:

- E^5 *is !* $(E^5_{mod}$*=sketch)* $\otimes$ *!* $(E^5_{repr}$ *=*  *))* $\otimes$ *!* $(E^5_{time}$*=(7,13))* $\otimes$ *!* $(E^5_{concept}$*=(river))* $\otimes$ *!*$(E^5_{role}$*=(nn))*
- $E^{5'}$ *is !* $(E^{5'}_{mod}$*=sketch)* $\otimes$ *!* $(E^{5'}_{repr}$ *=*  *))* $\otimes$ *!* $(E^{5'}_{time}$*=(7,13))* $\otimes$ *!* $(E^{5'}_{concept}$*=(street))* $\otimes$ *!*$(E^{5'}_{role}$*=(nn))*□

In this case, the alignment of the element E^2 with the element E^5 detects a lexical ambiguity due to the fact that the element E^5 can have two different meanings, river (E^5) and street ($E^{5'}$), in the

context, and the deictic "*this*" of the element E^2 is not useful to disambiguate the meaning.
A lexical ambiguity is due to the fact that exists a node of the syntax-graph that has two successors that define two different concepts that do not refer to the same semantic concept.
Therefore, the rule that allows identifying this ambiguity is the following:

$$\exists\ E^i, E^j : \text{elements, } n : \text{node of the syntax-graph where}$$
$$((E^i_{concept} \neq E^j_{concept}) \otimes (E^i_{repr} \equiv E^j_{repr}) \otimes (E^i_{mod} \equiv E^j_{mod}) \otimes$$
$$(E^i_{role} \equiv E^j_{role}) \otimes ((E^i,n), (E^j,n) \text{ are arcs of the syntax-graph})$$

This rule intercepts lexical ambiguities that appear at modal level and propagated themselves at multimodal level.

3.4.1.2 Temporal-Semantic Ambiguity

A temporal-semantic ambiguity appears when different elements defined by different modalities are terminal nodes of the same node of the syntax graph.
Let be given the example where the user says by speech:


"this is a river"

while she/he draws by sketch (), as described by the following Multimodal Sentence:

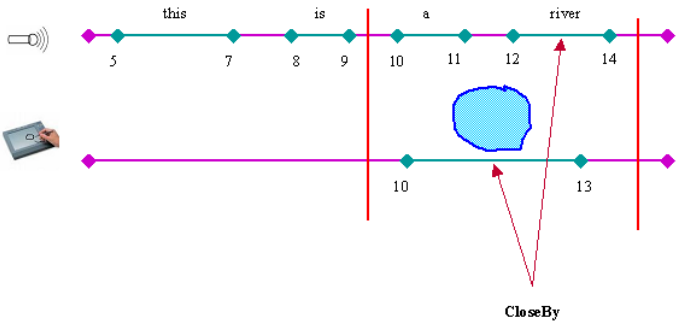


Figure 3.9: Example of input for Multimodal Sentence that defines a temporal-semantic ambiguity

This Multimodal Sentence has the following syntax-graph:

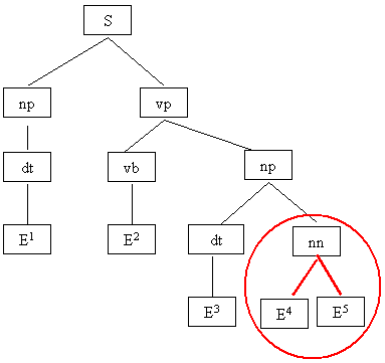


Figure 3.10: Syntax-graph of the Multimodal Sentence that defines a temporal-semantic ambiguity

The elements defined by the speech modality are:

- E^1 *is!* $(E^1_{mod}=speech) \otimes ! (E^1_{repr}=$ *"this"*$)) \otimes ! (E^1_{time}=(5,7)) \otimes ! (E^1_{concept}=(deictic)) \otimes !(E^1_{role}=(dt))$
- E^2 *is!* $(E^2_{mod}=speech) \otimes ! (E^2_{repr}=$ *"is"*$)) \otimes ! (E^2_{time}=(8,9)) \otimes ! (E^2_{concept}=(is)) \otimes !(E^2_{role}=(vb))$
- E^3 *is!* $(E^3_{mod}=speech) \otimes ! (E^3_{repr}=$ *"a"*$)) \otimes ! (E^3_{time}=(10,11)) \otimes ! (E^3_{concept}=(a)) \otimes !(E^3_{role}=(dt))$

- E^4 *is* ! (E^4_{mod}=*speech*) ⊗ ! (E^4_{repr} =  *"river"*)) ⊗ ! (E^4_{time}=(12,14)) ⊗ ! ($E^4_{concept}$=(river)) ⊗ !(E^4_{role}=(nn))

The element defined by the sketch input is:

- E^5 *is* ! (E^5_{mod}=*sketch*) ⊗ ! (E^5_{repr} = )) ⊗ ! (E^5_{time}=(10,13)) ⊗ ! ($E^5_{concept}$=(lake)) ⊗ !(E^5_{role}=(nn)) □

Also in this case there are two elements that are a terminal node of the same antecedent node and that define two different concepts that are not coherent at the semantic level. Therefore the rule that allows to detect this kind of ambiguity is the following:

$$\exists\ E^i, E^j : \text{elements, } n : \text{node of the syntax-graph where}$$
$$((E^i_{concept} \neq E^j_{concept}) \otimes (E^i_{repr} \neq E^j_{repr}) \otimes (E^i_{mod} \neq E^j_{mod}) \otimes (E^i_{role} \equiv E^j_{role}) \otimes (E^i_{time}\ CloseBy\ E^j_{time}) \otimes ((E^i,n), (E^j,n) \text{ are arcs of the syntax-graph})$$

3.4.1.3 Target Ambiguity

Finally the target ambiguity appears when the focus of the user is not clear and in particular two possible elements can be the targets of the user and so they can share the same role in the structure of the sentence.
For example let us suppose that the user interacting with an interactive map uses speech and sketch input. By speech she/he says:

 "show this near school"

and, at the same time she/he selects both an hotel and a restaurant by sketch (Figure 3.11).

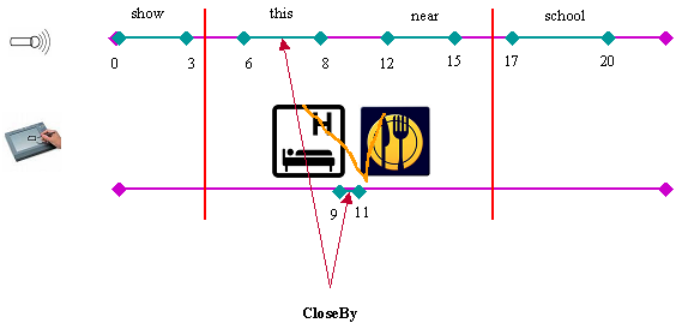


Figure 3.11: Example of input for Multimodal Sentence that defines a target ambiguity

The target ambiguity appears because the user checks two different elements ("*hotel*" and "*restaurant*") using the sketch modality (Figure 3.11).
This Multimodal Sentence defines the following syntax-graph:

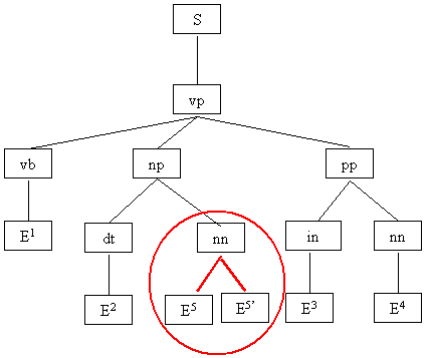


Figure 3.12: Syntax-graph of the Multimodal Sentence that defines a target ambiguity

Elements defined by the speech are:

- E^1 *is!* $(E^1_{mod}=speech) \otimes !\ (E^1_{repr}=$ *"show"* $)) \otimes !\ (E^1_{time}=(0,3)) \otimes !\ (E^1_{concept}=(verb)) \otimes !(E^1_{role}=(vb))$
- E^2 *is!* $(E^3_{mod}=speech) \otimes !\ (E^2_{repr}=$ *"this"*$)) \otimes !\ (E^2_{time}=(6,8)) \otimes !(E^2_{concept}=(deictic)) \otimes !(E^2_{role}=(dt))$

- E^3 *is!* $(E^3_{mod}$=*speech)* ⊗ *!* $(E^3_{repr}$ =  *"near"))* ⊗ *!* $(E^3_{time}$=*(12,15))* ⊗ *!*$(E^3_{concept}$=*(adverb))* ⊗ *!*$(E^3_{role}$=*(in))*
- E^4 *is!* $(E^4_{mod}$=*speech)* ⊗ *!* $(E^4_{repr}$ = *" school"))* ⊗ *!* $(E^4_{time}$=*(17,20))* ⊗ *!*$(E^4_{concept}$=*(city))* ⊗ *!*$(E^4_{role}$=*(nn))*

And using the sketch modality the user checks the following elements:

- E^5 *is ! (*E^5_{mod}=*sketch)* ⊗ *!* $(E^5_{repr}$ =  *))* ⊗ *!* $(E^5_{time}$=*(9,11))* ⊗ *!* $(E^5_{concept}$=*(hotel))* ⊗ *!*$(E^5_{role}$=*(nn))*
- $E^{5'}$ is ! $(E^{5'}_{mod}$=sketch) ⊗ ! $(E^{5'}_{repr}$ = )) ⊗ ! $(E^{5'}_{time}$=(9,11)) ⊗ ! $(E^{5'}_{concept}$=(restaurant)) ⊗ !$(E^{5'}_{role}$=(nn)) □

The alignment of the element E^2 with the elements E^5 and $E^{5'}$ detects a target ambiguity due to the fact that using sketch modality two different elements, "*hotel*" (E^5) and "*restaurant*" ($E^{5'}$), are identified.
Similarly to the temporal-semantic ambiguity, a target ambiguity is due to the fact that it exists a node of the syntax-graph having two successors that define two different concepts, which are not coherent at the semantic level (Figure 3.12). However, differently from the temporal-semantic ambiguity, in this case the identified elements have two different representations and, the rule that allows identifying this ambiguity is the following:

$$\exists\ E^i, E^j : \text{elements, } n : \text{node of the syntax-graph where}$$
$$((E^i_{concept} \neq E^j_{concept}) \otimes (E^i_{repr} \neq E^j_{repr}) \otimes (E^i_{mod} \equiv E^j_{mod}) \otimes (E^i_{role} \equiv E^j_{role}) \otimes ((E^i,n), (E^j,n) \text{ are arcs of the syntax-graph})$$

3.4.2 Syntactic ambiguities

Syntactic ambiguities arise when alternative structures of the Multimodal Sentence for a given set of elements can be generated during the interpretation process. In particular considering our

approach this class of ambiguities appears when a terminal node is not completely defined or more than one path on the syntax-graph allows to reach the same terminal node.
Syntactic ambiguities include: gap, analytic and attachment ambiguities. They are detailed in the following sub-sections.

3.4.2.1 Gap Ambiguity

This section starts analysing the gap ambiguity that appears when an element of the Multimodal Sentence is omitted. The detection of this kind of ambiguity can be connected with both rules about the grammar of the language and rules about the context.
Several examples of gap ambiguity have been presented in Chapter 2 (see section 2.2.2.4 and section 2.3.2.3). Some examples of gap ambiguity appear when the user specifies an action without specifying the object of the action.
Moreover, let us suppose that context rules impose to "*associate each deictic of the Multimodal Sentence to a Multimodal Element*".
An example of this kind of ambiguities appears when user interacts with a map saying by speech:

- "*Find this near this*"

And immediately after she/he draws a lake () using the sketch modality.
In detail the alignment of the elements that compose the user input in the multimodal sentence are the following:

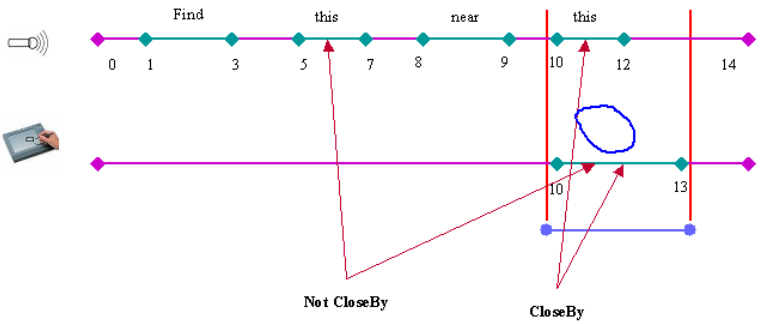


Figure 3.13: Alignment of elements that compose user input by speech and sketch modalities

The syntax-graph associated to the Multimodal Sentence is defined in Figure 3.14 and it underlines that there is a gap ambiguities because in this case the element that corresponds to the syntactic role (n) (Figure 3.14) is not defined in the Multimodal Sentence.

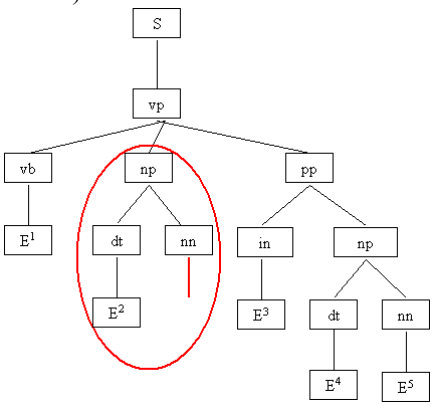


Figure 3.14: Syntax-graph associated to the Multimodal Sentence that defines a gap ambiguity

The Multimodal Sentence is composed by the following elements defined by the speech modalities:

- E^1 *is!* $(E^1_{mod}$*=speech)* $\otimes$ *!* $(E^1_{repr}=$ *"Find"* $)) \otimes$ *!* $(E^1_{time}=(1,3)) \otimes$ *!* $(E^1_{concept}=(verb)) \otimes$ *!*$(E^1_{role}=(vb))$

- E^2 *is!* $(E^3_{mod}$=*speech)* $\otimes$ *!* $(E^2_{repr}$ =  *"this"))* $\otimes$ *!* $(E^2_{time}$=*(5,7))* $\otimes$ *!*$(E^2_{concept}$=*(deictic))* $\otimes$ *!*$(E^2_{role}$=*(dt))*
- E^3 *is!* $(E^3_{mod}$=*speech)* $\otimes$ *!* $(E^3_{repr}$ =  *"near"))* $\otimes$ *!* $(E^3_{time}$=*(8,9))* $\otimes$ *!*$(E^3_{concept}$=*(adverb))* $\otimes$ *!*$(E^3_{role}$=*(in))*
- E^4 *is!* $(E^4_{mod}$=*speech)* $\otimes$ *!* $(E^4_{repr}$ =  *" this"))* $\otimes$ *!* $(E^4_{time}$=*(10,12))* $\otimes$ *!*$(E^4_{concept}$=*(deictic))* $\otimes$ *!*$(E^4_{role}$=*(dt))*

And using the sketch modality the element of the Multimodal Sentence is:

- E^5 *is !* $(E^5_{mod}$=*sketch)* $\otimes$ *!* $(E^5_{repr}$ =  *))* $\otimes$ *!* $(E^5_{time}$=*(7,13))* $\otimes$ *!* $(E^5_{concept}$=*(lake))* $\otimes$ *!*$(E^5_{role}$=*(nn))* □

The syntax graph shows a gap ambiguity because there is an instance of the action ("find") but there is not an instance of the object of the action (*n*).
Moreover, the context rule that imposes to "*associate each deictic of the Multimodal Sentence to a Multimodal Element*" is not satisfied.
Considering the previous formalism this ambiguity can be detected using the following rule:

$\exists$ E^i*:: terminal node of the syntax-graph, n: : vertex s.t.* $(E^i \equiv$ *null)*
$\otimes$ *((*E^i*,n) are arcs of the syntax-graph)*

3.4.2.2 Analytic Ambiguity

A further class of syntactic ambiguities is the analytic. It arises when the role of the element is not univocally defined in the Multimodal Sentence. In this case the element has more than one possible syntactic role in the Multimodal Sentence.
An example of this ambiguity, described in Chapter 2 (see section 2.2.2.1) and widely used in literature for natural language, is given by the sentence "The Tibetan history teacher" [Hir87].

Here a similar example in a map-based context is presented. Let us suppose that the Multimodal Sentence involves sketch and handwriting modalities and the user says by speech:

"*show Italian river*"

and immediately after she/he write the word "name" (name) using the handwriting modality.
The Multimodal Sentence is defined below.

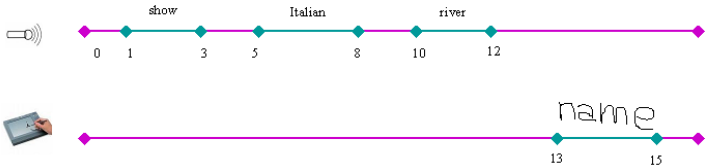
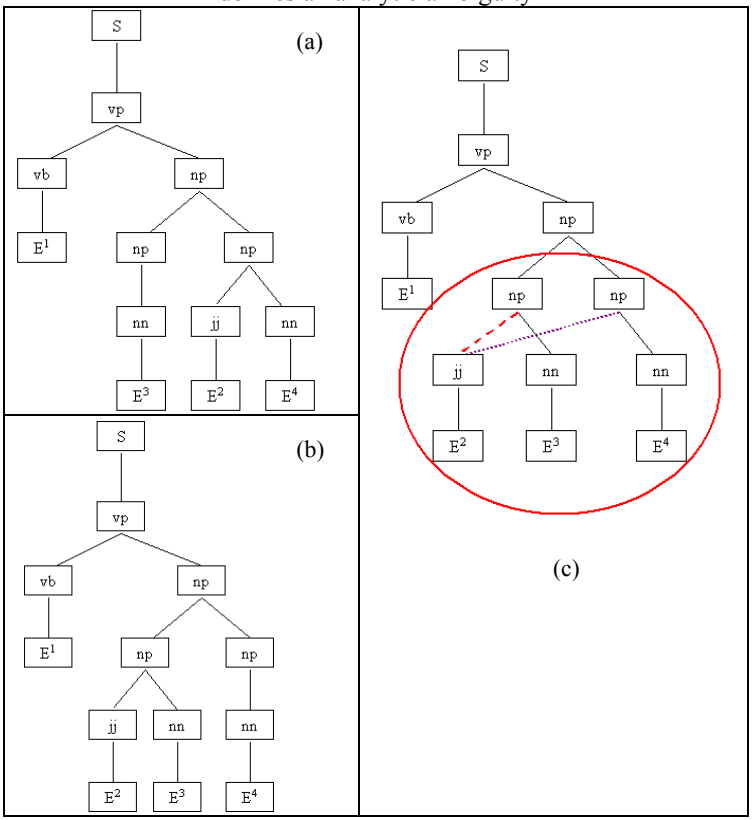


Figure 3.15: User input by sketch and handwriting modalities

This Multimodal Sentence defines the following syntax-graph:

Table 3.1: Syntax-graph associated to the Multimodal Sentence that defines an analytic ambiguity



Elements defined by the speech modality are:

- E^1 *is!* $(E^1_{mod}$*=speech)* $\otimes$ *!* $(E^1_{repr}$=🔊*"show"))* $\otimes$ *!* $(E^1_{time}$*=(1,3))* $\otimes$ *!* $(E^1_{concept}$*=(verb))* $\otimes$ *!*$(E^1_{role}$*=(vb))*
- E^2 *is!* $(E^2_{mod}$*=speech)* $\otimes$ *!* $(E^2_{repr}$ =🔊*"Italian"))* $\otimes$ *!* $(E^2_{time}$*=(5,8))* $\otimes$ *!*$(E^2_{concept}$*=(adjective))* $\otimes$ *!*$(E^2_{role}$*=(jj))*
- E^3 *is!* $(E^3_{mod}$*=speech)* $\otimes$ *!* $(E^3_{repr}$ =🔊*"river"))* $\otimes$ *!* $(E^3_{time}$*=(10,12))* $\otimes$ *!*$(E^3_{concept}$*=(river))* $\otimes$ *!*$(E^3_{role}$*=(nn))*

The element defined by handwriting modality is:

- E^4 *is* $!\ (E^4_{mod}$=*handwriting)* $\otimes\ !\ (E^4_{repr}$ = )) $\otimes\ !$ $(E^4_{time}$=*(13,15))* $\otimes\ !\ (E^4_{concept}$=*(name))* $\otimes\ !(E^4_{role}$=*(nn))* □

Considering the syntax-graph obtained by this Multimodal Sentence (Table 3.1c) there is more than one arc that reaches the element E^2.
The element E^2 can be part of the same *noun phrase* of both the element E^3 and E^4 because the Multimodal Sentence can be interpreted as: 1) show the Italian name of the river (Table 3.1a); and 2) show the name of the Italian river (Table 3.1b).
In this figure the element E^2 has two different roles in the syntax-graph because there are two different paths that allow reaching this element.
In general, this ambiguity is due to the fact that the same element E^j can plays two different roles in the same Multimodal Sentence and the rule for detect it is the following:

$\exists\ E^j$ *: element, n, m : vertexes of the syntax-graph where* $(((E^j,n),$ (E^j,m) *are paths on the syntax-graph)* $\otimes\ ((E^j,n) \neq (E^j,m)))$

3.4.2.3 Attachment Ambiguity

Considering the attachment ambiguity, it appears when the prepositional phrase (pp) can be legally attached to two different parts of the sentences. In order to specify this ambiguity let be suppose that the user interacts with a map using sketch and speech input.
Using the speech modality the user says:

"show this near school with garden"

and, at the same time she/he draws a house by sketch modality (Figure 3.16).
The Multimodal Sentence is defined below.

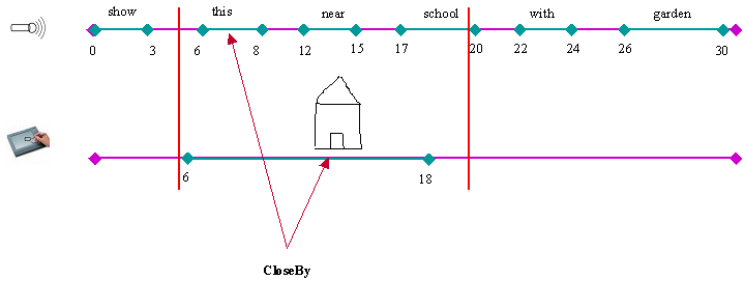
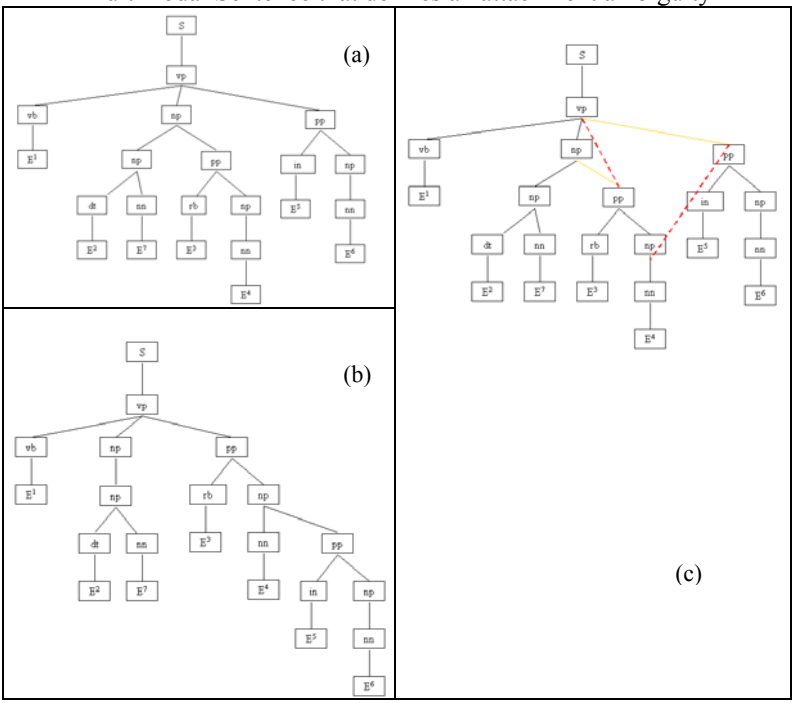


Figure 3.16: Elements defined by the user input

This Multimodal Sentence defines the following syntax-graph.

Table 3.2: Syntax-trees and syntax-graph associated to the Multimodal Sentence that defines an attachment ambiguity



The speech modality elements are:

- E^1 is! (E^1_{mod}=speech) ⊗ ! (E^1_{repr}= "show")) ⊗ ! (E^1_{time}=(0,3)) ⊗ ! ($E^1_{concept}$=(verb)) ⊗ !(E^1_{role}=(vb))
- E^2 is! (E^3_{mod}=speech) ⊗ ! (E^2_{repr} = "this")) ⊗ ! (E^2_{time}=(6,8)) ⊗ ! ($E^2_{concept}$=(deictic)) ⊗ !(E^2_{role}=(dt))
- E^3 is ! (E^3_{mod}=speech) ⊗ ! (E^3_{repr} = "near")) ⊗ ! (E^3_{time}=(12,15)) ⊗ ! ($E^3_{concept}$=(adverb)) ⊗ !(E^3_{role}=(rb))
- E^4 is! (E^4_{mod}=speech) ⊗ ! (E^4_{repr} = " school")) ⊗ ! (E^4_{time}=(17,20)) ⊗ ! ($E^4_{concept}$=(school)) ⊗ !(E^4_{role}=(nn))

- E^5 is! (E^5_{mod}=speech) ⊗ ! (E^5_{repr} =  " with")) ⊗ ! (E^5_{time}=(22,24)) ⊗ ! ($E^5_{concept}$=(adverb)) ⊗ !(E^4_{role}=(in))
- E^6 is ! (E^6_{mod}=speech) ⊗ ! (E^6_{repr} =  "garden")) ⊗ ! (E^6_{time}=(26,30)) ⊗ ! ($E^6_{concept}$=(garden)) ⊗ ! (E^6_{role}=(nn))

The sketch modality element is:

- E^7 is ! (E^7_{mod}=sketch) ⊗ ! (E^7_{repr} = )) ⊗ ! (E^7_{time}=(6,18)) ⊗ ! ($E^7_{concept}$=(house)) ⊗ !(E^7_{role}=(nn)) □

Considering this sentence "show this (house) near school with garden", an attachment ambiguity appears because two potential interpretations are defined: 1) in one syntax-tree "with the garden" is attached to the verb "show" (Table 3.2a); 2) in the second syntax-tree "with the garden" is attached to the element "school" (Table 3.2b).
The Table 3.2c shows that two different paths (----and ——) exist for reaching the prepositional part (pp) (Table 3.2c). Therefore an attachment ambiguity can be detected using the following rule:

$$\exists\, p, q : \textit{syntactic paths}, n: \textit{node of the syntax-graph where}$$
$$((n=pp) \otimes (n \in p) \otimes (n \in q) \otimes (p \neq q))$$

Therefore, each class of multimodal ambiguities can be detected using the rules provided in this chapter.

3.5 Conclusion and discussions

This chapter has discussed the problem of ambiguities in multimodal interaction, according to a linguistic point of view. It gives a classification of these ambiguities and the set of rules to identify them, using the notions of *Multimodal Grammar*, *Multimodal Sentence* and *Multimodal Language*, here defined.
These rules are summarized in the following table.

Table 3.3:Rules for detecting classes of multimodal ambiguities

Ambiguity classes		**Rule**
Semantic Ambiguities	Lexical	$\exists$ E^i, E^j : elements, n : node of the syntax-graph where $\exists$ $((E^i_{concept} \neq E^j_{concept}) \otimes (E^i_{repr} \equiv E^j_{repr}) \otimes (E^i_{mod} \equiv E^j_{mod}) \otimes (E^i_{role} \equiv E^j_{role}) \otimes ((E^i,n), (E^j,n)$ are arcs of the syntax-graph)
	Temporal-Semantic	$\exists$ E^i, E^j : elements, n : node of the syntax-graph where $\exists$ $((E^i_{concept} \neq E^j_{concept}) \otimes (E^i_{repr} \neq E^j_{repr}) \otimes (E^i_{mod} \neq E^j_{mod}) \otimes (E^i_{role} \equiv E^j_{role}) \otimes (E^i_{time}$ CloseBy $E^j_{time}) \otimes ((E^i,n), (E^j,n)$ are arcs of the syntax-graph)
	Target	$\exists$ E^i, E^j : elements, n : node of the syntax-graph where $\exists$ $((E^i_{concept} \neq E^j_{concept}) \otimes (E^i_{repr} \neq E^j_{repr}) \otimes (E^i_{mod} \equiv E^j_{mod}) \otimes (E^i_{role} \equiv E^j_{role}) \otimes ((E^i,n), (E^j,n)$ are arcs of the syntax-graph)
Syntactic Ambiguities	Gap	$\exists$ E^i:: terminal node of the syntax-graph, n: : vertex s.t. $(E^i \equiv null) \otimes ((E^i,n)$ are arcs of the syntax-graph)
	Analytic	$\exists$ E^j : element, n, m : vertexes of the syntax-graph where $(((E^j,n), (E^j,m)$ are paths on the syntax-graph) $\otimes ((E^j,n) \neq (E^j,m)))$ $\exists$
	Attachment	$\exists$ p, q : syntactic paths, n: node of the syntax-graph where $((n=pp) \otimes (n \in p) \otimes (n \in q) \otimes (p \neq q))$ $\exists$

Establishing the class a Multimodal ambiguity is referred to, it is very relevant for its solution.
In fact, different solution strategies and methods can be adopted to solve ambiguities, according to the different classes (as will showed in the next chapters).
After the analysis of modal and multimodal ambiguities this thesis dissertation in Chapter 4 describes methods provided in the literature for solving ambiguities. This analysis has lead to the definition of the methods used to solve multimodal ambiguities. These methods use information about ambiguities classes and define a specific resolution model for each class of multimodal ambiguities. These models will be will be presented and detailed in the Chapter 5.

Chapter 4

Methods for Solving Ambiguities

4.1 Introduction

Starting from the classifications of ambiguities given in the literature and provided in Chapter 2, this chapter presents several strategies and methods for avoiding and solving those ambiguities in particular considering the problem for Visual Languages and Natural Language.
The presented solution methods can be grouped adopting different points of views and classes. Here they are organised according to the following three classes: *prevention of ambiguities*, *a-posterior resolution* and *approximation resolution methods*.
Prevention methods consist of adopting a predefined interpretable interaction user's behaviour according to a set of transitions between different allowed states of the interaction process. A-posterior resolution methods are based on the mediation techniques [MHA00] enabling the user to disambiguate her/his intention by dialog. Finally, approximation resolution methods use probabilistic approaches. They are here presented and discussed, with a particular attention on Hidden Markov Models and Hierarchical Hidden Markov Models, which have been widely used for multimodal ambiguities resolution in Chapter 5.

4.2 Prevention methods for dealing ambiguities

This class of methods imposes users to follow predefined interaction behaviour according to a set of transitions between different allowed states of the interaction process. In particular,

prevention methods impose the user to respect some interaction constraints, and they are usually adopted in command user interfaces (that are interfaces using a set of textual or visual pre-defined commands) and in sketch-based interfaces (freehand drawing) whose behaviour is controlled by transitions between a limited set of allowed predefined states. The main methods to prevent ambiguous interpretation are:

- procedural method,
- reduction of the expressive power of the language grammar,
- improvement of the expressive power of the language grammar.

These methods are discussed in the following sub-sections, with descriptions of their main features and some examples.

4.2.1 The procedural method

This method permits the user to freely interact with the system; however, the user's input produces the system states transition according to a pre-defined sequence of states that controls the system evolution. The procedural method reduces the Human-Computer Interaction process within a closed procedure imposing the user to respect predefined interaction behaviour.
In the free-hand drawing process some constraints in user's drawing behaviour can be defined to avoid ambiguities; the system could require that the user draws one sketch according to a predefined sequence of the system's state diagram.
Let be given for example a system with two possible states for editing an entity-relationship diagram using a sketch-based interaction approach. At the beginning the system is in the state 1. The user draws an entity, which can be interpreted. Any other input doesn't produce any interpretation. When the user draws a second entity it is interpreted, and the system has a transition in the state 2. When all couples of entities are connected by a relationship, the system is in the state 1 and only input entities can be interpreted. Differently, the system is in the state 2. State 1 allows interpreting

only inputs of entities; state 2 interprets both drawing of an entity, or a relationship if two unconnected entities exist.
The user has to draw a sequence of strokes in which each entity or relationship must be completed before beginning a new one. At the beginning the system is in state 1. If the user draws the rectangle in Figure 4.1A containing the word *professor* and then the rectangle in Figure 4.1B containing the word *course*, the two rectangles are interpreted respectively as the entity *professor* and the entity *course*, due to the fact that a rectangle represents an entity. If the user's drawing produces an oval containing the word *course* (Figure 4.1C) then it will in any case be interpreted as a rectangle representing the entity *course*, in accordance with the state of the system (state 1) or, if the oval does not have the features to approximate a rectangle (it is not the case of Figure 4.1) there are not interpretations for the input.

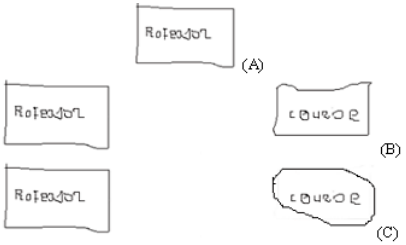


Figure 4.1: Example of procedural drawing of entity

The procedural method can also be used to avoid ambiguities such as the segmentation ambiguity (Chapter 2). Let be given the image associated with the visual sentence in Figure 4.2.

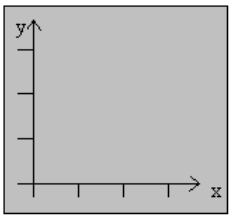


Figure 4.2: Segmentation ambiguity

A segmentation ambiguity can arise, because the short lines in the lower left corner of the diagram could represent tick marks or the ends of the x and y axis lines. The definition of constraints on the stroke drawing process can help to remove this ambiguity. Consider the constraint, which establishes that:

Each sequence of strokes must complete one visual element (axis and ticks) before beginning a new one

Suppose that Figure 4.3 shows the user's sequence of actions made to obtain Figure 4.2. The defined constraint does not allow including in one action only to draw two different visual elements. It is possible to consider the drawing process described by a two states diagram: state 1 is associated with the drawing of axis or sticks, while state 2 is associated with drawing of sticks only. The two drawing starting from the left of Figure 4.3 correspond to the state 1. The remaining corresponds to the state 2.

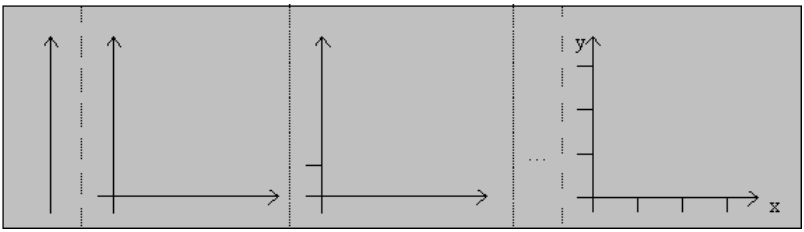


Figure 4.3: Procedural drawing to avoid segmentation ambiguity

The short lines in the lower left corner in Figure 4.2 are therefore the ends of the x and y axes, as a single stroke for drawing each one of the two axes has been made and they contain the short lines involved in the ambiguity.

Consider an example that uses the procedural approach, according to a given constraint, to reduce ambiguities due to not required spatial relationships between two of the three geometric objects in a two-dimensional space.
A sentence containing this kind of ambiguity is given below:

Let be given a line that crosses a rectangle and a rectangle that overlaps an oval. No relationships are expressed between the oval and the line.

Three different objects in a bi-dimensional space have to be represented as shows Figure 4.4. The Oval and the line are in an un-desired disjoint relationship, that differs from the previously given description.

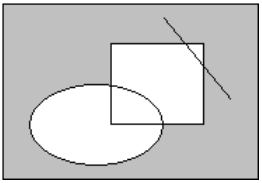


Figure 4.4: Ambiguity due to not required spatial relationships between two of the three Characteristic Structures, forced in a two-dimensional space

To solve this ambiguity the Visual Spatial Query Language proposed in [LeC95] uses the foreground/background metaphor notion.
Figure 4.5 shows how the Lee and Chin's Language expresses the query:

Find all the regions that are passed through by a river and partially overlap a forest

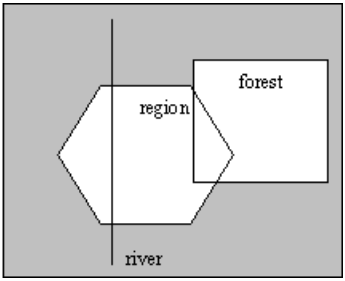


Figure 4.5: Lee and Chin's cs of the Visual Sentence

The user starts from drawing a line for the river and the area for the region in the foreground with a pass-through relationship (Figure 4.5). The river is then put in the background, as it has no relationship with the forest. Finally, the user draws the forest area overlapped with the region area in the foreground. As explained a state diagram permits to control and specify background and foreground, solving the ambiguity.
Some spatial visual languages, including that of [LeC95], adopt a strategy of ambiguity prevention through a procedural method. Lee & Chin's language enables removal of unwanted relationships between drawn symbolic graphical objects or the imposition of an a priori restrictive interpretation using the foreground/background metaphor. The relationships of a new symbolic graphical object depend on the state (foreground or background) of those drawn previously. It is necessary to consider both the visual representation and the drawing process to interpret a query. VISCO [WeH98] also uses a procedural method to prevent ambiguity.

4.2.2 Reduction of the expressive power of the Language grammar

Reduction of the expressive power of the Languages grammar is the second method used to prevent ambiguities.
Users sometime interact through meaningless actions, involving some elements of the language relationships among them in a completely free approach. Free interaction can produce ambiguities in interpretation. A common method to resolve these ambiguities is the reduction of the language grammar's expressive power. A set of constraints can be established on the grammar of the language to limit the user's actions, the number of characteristic structures used and their relationships. All meaningless commands and selections are disabled or ignored.
This method enables all commands and elements according to the syntactic and semantic constraints. It is adopted in the WYSIWYG (What You See Is What You Get) command user interfaces, where icons and characteristic structures are disabled when they cannot be

used according to their syntactic and semantic constraints, and/or the system can send the user a warning message on the interaction error.

Consider a system for editing an Entity-Relationship Diagram. The system provides the user with the characteristic structures used for an entity and a relationship. When the user edits his/her Entity-Relationship diagram, the system enables all actions that cannot produce syntactic and semantic errors and ambiguities. The relationship definition, therefore, requires the preliminary identification of the entities that it connects. This approach can avoid errors and ambiguities in sentences that cannot be easily interpreted.

This solution method prevents the segmentation ambiguity (Chapter 2) because each characteristic structure must be univocally intercepted by a command (user action) and unambiguously interpreted. It can also avoid occlusion ambiguity (Chapter 2), by splitting superposed characteristic structures. When the user edits the visual sentence, the system can impose that the characteristic structures must be spatially distinguishable.

For example, let be given a system for Entity-Relationship Diagram editing. To avoid segmentation ambiguities the user is enabled to superpose two characteristic structures (entity, relationships) (Figure 4.6), and she/he cannot edit isolated relationships (unless they are each connected with two entities).

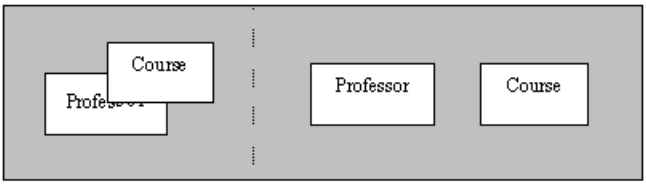


Figure 4.6: Enabled (A) and disabled (B) configurations for Characteristic Structures

The grammar's expressive power reduction can be meaningfully used in the case (chapter 2) of ambiguities due to not required spatial relationships between two of the three characteristic

structures, forced in a bi-dimensional space. To go back to the previous description:

Let be given a line that crosses a rectangle and a rectangle that overlaps an oval. No relationships are expressed between the oval and the line

To avoid the undesired disjointed relationship (see Figure 4.4) between the line and oval, the language can restrict the visual sentences to impose one spatial relationship between at most two elements at the same time.
This ambiguity may be resolved by reducing the grammar's expressive power. If three elements must be considered, as in the above description, then the visual language can be integrated with a textual part to represent the situation described in Figure 4.7.

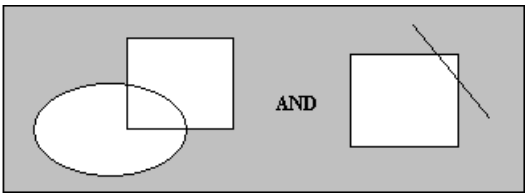


Figure 4.7: Unambiguous expression using a grammar with low expressive power

Grammars with a low expressive power can be found among the Visual Spatial Query Languages. These include Pictorial Query By Example [CaM94] and SVIQUEL [Mey93]. By considering limited kinds of spatial relations (directional relations) only, PQBE avoids multiple query interpretations but reduces the possibility to formulate more complex queries involving topological relationships. SVIQUEL also includes topological operators, but avoids multiple interpretations by limiting the number of objects involved (to just two) and providing a tool with a low expressive power to specify the relative spatial positions.

4.2.3 Improvement of the expressive power of Language grammar

The improvement of the expressive power of the languages grammar is the last method presented to prevent ambiguities. In contrast with the reduction method, improving the grammar's expressive power produces the system is closest to the user's intention. This approach enriches the language by defining new operators and new elements of the language.
Let us consider this description:

Let there be a line that crosses a rectangle and a rectangle that overlaps an oval. No relationships are expressed between the oval and the line.

The ambiguity of expressing spatial relationships between two of the three elements (line, rectangle, oval) in a two-dimensional space can be resolved by introducing a new operator. This solution is proposed by GeoPQL (Geographical Pictorial Query Language) [FeR05], which allows the user to represent only the desired relationships between the classic shapes (or features) "point", "polyline" and "polygon", and it assigns them a precise semantic (for example, "lake" to a polygon, "river" to a polyline, etc.). For example this language defines the *any* operator, which expresses all valid spatial relationships between two characteristic structures. Figure 4.8 shows the image associated with the visual sentence corresponding to the description below:

Find all the regions that are passed through by a river and partially overlap a forest

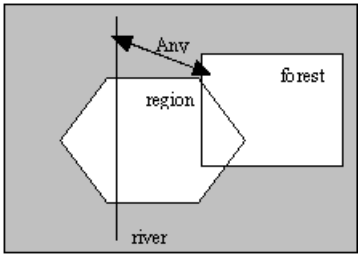


Figure 4.8: The image associated with the Visual Sentence using the Any operator in GeoPQL

The system can reduce interpretation ambiguities by taking account of contextual information, where the context is defined here as information about the application domain (language syntax and semantic), the interaction tool and the user's skill.
The increasing tendency to use different devices for human-computer interaction activities and the pervasive use of mobile devices and PDAs (Personal Digital Assistants) has led to the need for new elements of the language, depending on the different devices and users leading to the Language's personalization. This produces the improvement in the grammar's expressive power.

4.3 A-posterior methods for dealing ambiguities

The a-posterior resolution of ambiguities uses mediation approach [DeM05].
Mediation is the process that facilitates the correct interpretation of the user's actions during interaction according to her/his intention. It involves the user in the disambiguation activity, asking her/him for the correct interpretation. These methods are particularly useful for the resolution of ambiguities caused by imprecision and noises in the human-computer interaction process.
Various mediation techniques exist:

- *repetition*. The user repeats an action until the system is able to interpret it correctly.

- *choice*. The system returns a candidate set of interpretations and the user selects the most appropriate.

These techniques allow a dialog between the user and the system in order to solve ambiguities that appear during the interaction.
This approach can be used in both in the repetition and in the choice methods, which are described below.

4.3.1 Repetition

This approach consists of the repetition of the user's action until the system is able to interpret it correctly.
In [DeM05] the different features of the repetition strategy are classified as:

- Modality,
- Granularity of Repair,
- Undo.

These strategies are detailed below.

4.3.1.1 Modality

The *Modality* feature stresses the repetition method by the perspective of the modality adopted for repetition. If the interaction modality used is mainly gesture (freehand sketching), then repetition using the same modality can be useful. However, repetition can be more effective if a different modality is used to resolve the introduced ambiguity, as use of the same modality frequently results in the same ambiguities. In fact, the user frequently replies the same errors and ambiguities when he/she uses the same modality.
However the use of a different modality can produce a potential conflict.
The need to solve ambiguities can imply the user has to add useful information for the disambiguation process. This information can

be complementary, redundant or concurrent and it can be provided using different modalities.
When repetition combines different modalities, redundancy is mainly used to solve ambiguities. When modalities are redundant, the system integrates the same chunk of information that is transmitted using more than one mode. This information can be jointly used to solve all the ambiguities introduced in the second section and an example of redundancy is provided in the following. Let us consider ambiguities due to the missing closure of a polygon (see Chapter 2). In order to solve this kind of ambiguities, system could allow user to repeat the same input using a different modality, for example voice. Once the user ends her/his drawing, she/he can say the word "polygon". This information allows to the system to interpret the sketch as a polygon (see Chapter 2). Similarly to the previous case, ambiguities due to crosses in a stroke, ambiguities due to intersection of two polygons, ambiguities due to the generation of undesired polygons and polylines and, finally, ambiguities due to the over-tracing of different strokes can be solved using the repetition by another modality.

4.3.1.2 Granularity of Repair

The second sub-method of repetition is the granularity of repair method that can be applied locally to resolve a more general ambiguity.
In this case, as introduced in [MHA00], the granularity of correction is different from the granularity of interpretation. In fact, correction is related only to a component part of the sketched object to be interpreted.

4.3.1.3 Undo

In the dictation System described in [SKG00], the user can introduce or delete some characters in a sentence. Before repeating any action, the user can delete or undo the undesired action. Undo or delete are necessary when the input is a command; i.e. the

repetition must undo this command before repeating the user's action. It should be noted that if no interpretation can be given to the user's action, then undo couldn't be applied, as there is nothing to undo.

4.3.2 Choice

The second method for the a-posteriori resolution of ambiguities is the choice. This method consists of a dialogue with the user that enables the system to identify the correct interpretation of each ambiguity. The system shows the candidate interpretations to the user, which can choose the best one according to her/his intention. This method provides a feedback according to the user's behaviours and preferences.
This method is adopted in some visual spatial query languages, including Sketch [FeR05], Spatial Query By Sketch [Ege97] and Cigales [CaM94]. Spatial Query By Sketch resolves the ambiguity problem by considering and proposing both the exact solution of the query, if possible, and other approximate solutions obtained by relaxing some relationships. In this manner the language includes multiple interpretations in the result, and the user selects the representation that correctly interprets her/his query. Cigales is unable to give a unique interpretation of the visual query representation. Two possible solutions to reduce ambiguity in this language are given by the introduction of various interactions (feedback) with the user, and by the increased complexity of the resolution model. Ambiguities are resolved by detection, attempt at automatic solution and proposal of these solutions to the user.
Moreover, to disambiguate input produced by sketch-based interaction using the choice method, the beautification processes can be useful to obtain the set of candidate solutions. The user then chooses the most correct interpretation.

4.4 Approximation methods for dealing ambiguities

Ambiguities caused by imprecision in Human-Computer Interaction behaviour can also be resolved using approximation resolution methods such as:

- Thresholding,
- Historical Statistics,
- Rules.

These methods do not require any user involvement in the disambiguation process. They can all require the use of some theories, such as Fuzzy Logic, Markov Random Field, Bayesian Networks and Hidden Markov Models.
The main features of these methods and models are briefly described in the following sections.

4.4.1 Thresholding

Thresholding [MaC99] is a method to resolve ambiguities caused by imprecision. The correctness of the user's input is expressed by a probability, which can be compared to a threshold; the recogniser returns a confidence score that measures the probability that a user's input has been correctly recognized. If this confidence measure is below a predefined threshold, the system rejects the interpretation. Thresholding is used in [PWC95].

4.4.2 Historical Statistics

If the confidence score is not available or might be incorrect, probabilities can be generated by performing a statistical analysis of historical ambiguity data. Historical statistics may provide a default probability of correctness for a given interpretation when a recogniser is unable to do so. This approach may use a confusion

matrix, whose values give an estimation of the number of times that the recogniser has confused the Characteristic Structure. So if thresholding is unable to disambiguate the freehand sketches, historical statistical data on correctly interpreted ambiguities can be used.

4.4.3 Rules

Freehand sketches are complex to recognize. In the absence of contextual information, their interpretation is often ambiguous. Their management may require the use of the context, as thresholding and/or statistical approaches may not be sufficient for their resolution. An example of use of rules can be found in [BaH93]. The use of Rules is more sophisticated than thresholding, as it allows the use of context. For example, a rule might use syntactic information to eliminate grammatically incorrect Characteristic Structures.
Let us consider the example of an Entity-Relationship editor. If the user's intention is to draw the entity *Professor* (Figure 4.9A), the entity *Course* (Figure 4.9B) and the relationship *Teach* between them, and the sketch is ambiguous due to ambiguity in the shape of the relationship (Figure 4.9C), then different strategies can be adopted to resolve the ambiguity.

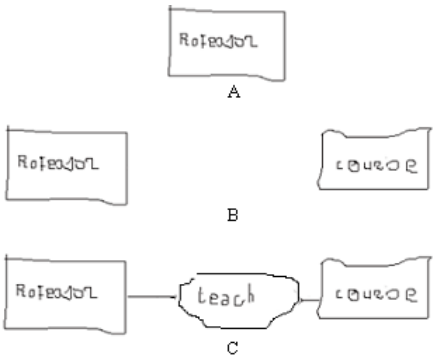


Figure 4.9: Procedural drawing of the E-R

In a sketch-based Interface the interpretation of the image associated with a visual sentence requires that the sketch will be matched to a set of Characteristic Structures and their spatial relationships. If the imprecision in drawing the *Teach* relationship is so high that it could be interpreted as a rectangle or rhombus and the system has to resolve this ambiguity, then thresholding and historical data could be used. However, if the thresholding value is low and historical data does not provide a solution on whether the shape is a rhombus or a rectangle, and then contextual information and the use of rules may be a solution. In fact, given that a connection between two entities can be obtained only by considering relationships, and the characteristic structure for a relationship is a rhombus, then Teach is a relationship, according to the syntactic rules for Entity-Relationship diagrams (Figure 4.9C) establishing that only a rhombus can connect two entities; then the correct interpretation is a rhombus.
These approximation methods can all require the use of some theories, such as Fuzzy Logic, Markov Random Field, Bayesian Networks, and Hidden Markov Models that are briefly described below.

4.4.3.1 Fuzzy logic

Fuzzy logic is widely used for imprecision and/or ambiguity problems, when a classification uncertainty appears. It is based on the fuzzy set concept, and was developed by [Zad65] to provide a general representation of uncertainty considering different degrees of membership values.
In general, when considering sets such as the set of natural numbers, the set of people, and so on, each object may or may not belong to this set. In contrast, the elements of a fuzzy set belong to the set with different graduations. The degree for each one of the elements of the set gives the degree of certainty that each element belongs to the set. “Fuzzy” therefore becomes synonymous with “imprecision”.
Given the imprecision in Human-Computer Interaction, fuzzy logic establishes the appropriate fuzzy set via membership functions,

which are associated with each input in order to resolve the vagueness and ambiguities of the interaction behaviour introduced by imprecision and noise.
CALI [FoJ00] is a freehand drawing information system that uses Fuzzy Logic for sketch recognition. Its recognition method is based on three main ideas: extraction of the geometric properties from input shapes, enhancement of the recognition performance using a set of filters to either identify shapes or remove unwanted shapes and resolution of uncertainty and imprecision in shape sketches by using fuzzy logic [Bez92] to associate a degree of certainty to the recognized shapes.
The thresholding method can be used in combination with Fuzzy Logic; for example, fuzzy thresholding can use entropy as the measure for "fuzziness", or it can use a method that minimizes a "fuzziness" measure of the mean level of greyness in the object and background.

4.4.3.2 Markov Random Field

Recognition of freehand sketches can depend strongly on the drawing's context. The spatial property can be effectively modelled through different aspects such as context, and Markov Random Field (MRF) theory provides a convenient, consistent way to model context-dependences. A Markov network is similar to a Bayesian network (which will be described in section 4.4.3.3) in its representation of dependencies, but can represent dependencies that a Bayesian network cannot, such as cyclic dependencies.

4.4.3.3 Bayesian Networks

A system, which uses Bayesian networks to deal with uncertainty and complexity, can manage ambiguities giving the set of interpretations that a system is considering. Bayesian networks consist of two parts: an acyclic direct graph and a set of probabilistic distributions. Each node of the graph represents one element to be interpreted and each arc gives the relationship between the two connected nodes. The different interpretations are related to different probabilities, which can be influenced by factors

such as context and the process. This approach models the set of interpretations by an acyclic direct graph and a set of probabilistic distribution. In the graph each node represents one interpreted element and each arc gives the relationship between the two connected nodes. Different interpretations are related to different probabilities that can be influenced by factors such as the context.

4.4.3.4 Hidden Markov Models

A HMM is used to model a Markov process (that is a doubly stochastic process) with unknown parameters that can be obtained by observable parameters. A HMM λ(A,B, π) is defined as a set of hidden states $Q = \{q_1, q_2, ...q_N\}$, a set of observation symbols $V = \{v_1, v_2, ...v_L\}$, that are emitted by the hidden states, and three parameters A, B, π that characterize the HMM, where:

- A is the transition probability matrix $a_{ij} = P(q_{t+1} = j|q_t = i)$, where A is a stochastic matrix with each row sums one, and it is often sparse;
- B is the observation probability distribution $B_j(v) = P(O_t = v|q_t = j)$,
- π is the initial state distribution.

A HMM can be represented using a direct graph; its nodes are associated with the HMM states, and arrows are associated to the allowable transitions with non-zero probability.
If the observations are discrete symbols, the observation probability distribution is defined as a matrix:

$$B_j(v) = P(O_t = v|q_t = j). \qquad (4.1)$$

When the observations are vectors in $\Re^L$, the observation probability distribution is defined as a Gaussian: $P(O_t = v|q_t = j) = N(v;\mu_j,\Sigma_j)$, where $N(v;\mu_j,\Sigma_j)$ is the Gaussian density that has mean μ and covariance Σ evaluated at v:

$$N(v;\mu,\Sigma) = \frac{1}{(2\pi)^{l/2}|\Sigma|^{\frac{1}{2}}} esp(-\frac{1}{2}(v-\mu)^{'}\Sigma^{-1}(v-\mu) \qquad (4.2)$$

A focal aspect connected with the HMMs is the assessment of the parameters previously introduced. According to this goal the literature provides some algorithms.
In detail, considering a HMM λ(A,B, π) and a sequence of observations O = o_1, o_2, .., o_k the parameters assessment requires to face three fundamental problems [RaJ86]:

- *Evaluation* that provides a measure of how close a given observation sequence matches the model;
- *Alignment* that provides the most likely state sequence (path) for a given observation sequence;
- *Training* that estimates the model parameter λ to maximise the P(O|λ) against given observation sequences.

In particular, it is possible to determine how well each model λ accounts for the observations by computing P(O|λ) using the Forward algorithm. Moreover the Viterbi algorithm [Vit67] can be used to calculate the best sequence of HMM states transitions for generating O and the Baum-Welch algorithm estimates HMM parameters A, B and π to maximize P(O|λ).
HMMs are a milestone among methods for dealing with sequences of information units, which are pieces of information composing a sequence. A HMM permits to model each unit to be recognized (hidden state) and these units are concatenated to allow their sequences recognition. HMMs provide efficient learning and recognition algorithms, which are able to simultaneously segment an incoming sequence into units and to identify these units.
Therefore HMMs can be used to model stochastic processes and sequences in different scenarios, such as computational molecular biology [KMH94], speech recognition [Rab89], handwriting recognition [NWF85] and so on. Moreover they can be usefully applied to natural language modelling [Jel85].
In detail, the adaptivity of the Hidden Markov Models provides that they can be applied to several pattern recognition applications.

For example, in speech recognition, isolated word can be recognized using HMMs assuming that each word is modelled by distinct HMMs [Rab89]. Furthermore this approach assumes that there is a training set of occurrences for each word, and each occurrence constitutes an observation sequence, that is the appropriate representation of the characteristics of the word.
Considering the sketch interaction, sketches are often compositional and incremental therefore observation symbols of the HMM respond to the encoding of input strokes in terms of lines of different orientations [SeD05]. In particular the incremental property of the strokes implies to consider the structure of a current stroke taking account of the previous and the next stroke. Therefore this sequence is modelled using a left-to-right HMM [JiS05] in adaptive way because the number of states in multi-strokes sketch recognition is dynamically determined by the structural decomposition of the target pattern. The model is defined by a training stage and the recogniser provides the probabilities and the recognition results in the sequence of probabilities from high to low using the trained HMM.
The problem of handwriting recognition has been treated using different approaches: using words models, using models based on letters, and using sub-character models. The second one of these approaches has been dealt by an HMM based on strokes [HBT94]. This approach considers letters as a concatenation of strokes and each stroke is modelled by one-state HMM.
Finally, considering the gesture recognition, to recognize dynamic gesture HMMs classify human's movements over a sequence of image [EKR98]. This system divides images in meshes and counts the number of pixels that represent the person for each mesh. The system composes a feature vectors that are classified based on discrete HMMs.
In a great part of these applications, an important issue is the difficulty connected with the multiplicity of length scales and recursive nature of sequences. These issues can be managed and overcome using stochastic context free grammars (SCFG) [SBH94] or hierarchical Hidden Markov Models (HHMMs) [FST98]. Considering SCFG, it is difficult to assess the parameters of stochastic processes because typically the likelihood of observed

sequences induced by a SCFG varies dramatically with small changes in the parameters of the model.
Moreover, the inside-outside algorithm [LaY90], which is the algorithm commonly used for parameters estimation of SCFGs, has a cubic time complexity in the length of the observed sequences.
HHMMs are an alternative respect to SCFGs and the following section provides a description of HHMM that analyses how they can be used to manage the hierarchical structure of sequences.

4.4.3.5 Hierarchical Hidden Markov Models

Moreover, this model allows inferring correlated observations over long periods in the observation sequence through the higher levels of the hierarchy. The hierarchical structure of the model permits an arbitrary number of activations of its sub-models. Parameters of the model are estimated through the inside-outside algorithm [FST98]. This estimation procedure is characterized by a quadratic computational time in the length of the observations. In [FST98] the Hierarchical Hidden Markov Models are trained using an unsupervised learning of repeated units (which in that work are strokes defining letters in cursive handwriting).
Hidden states of the HHMMs are autonomous probabilistic models where each state can be a HHMM and each state can emit sequences rather than a single symbol.
Therefore HHMMs generate sequences through a recursive activation of one of the sub-states of a state, and each sub-state can include sub-states and activate one of its sub-states. When the recursive activation process reaches a production state, the activation ends because the production state is the only state that emits output symbols. A production state chooses the output symbol to emit according to a probability distribution over the set of output symbols. Hidden states, which do not emit observable symbols, are internal states.
A transition between states can be vertical or horizontal.
A *vertical transition* activates a sub-state by an internal state, and returns the control to the activation state. When a vertical transition is completed because the level is reached, then the state that originated the recursive activation chain takes again the control.
A *horizontal transition* is a transition within the same level.

The vertical transitions and the set of states define a tree structure that has the root state in the node at the top of the hierarchy and that has the production states in the leaves.
HHMMs are defined by the n-ple [Q, O, (A, B, π)] where the set of the hidden states Q of the HHMM becomes composed of a set of production states P and the new set of internal states Q.
Let be N the total number of production states p, D the total number of internal states q, and M the total number of observation symbols v_i (with i=1…M);
Therefore a state of an HHMM is denoted by q_i^d ($d \in \{1..D\}$ where i is the state index and d is the hierarchy index (associated with the hierarchy level). In particular, the hierarchy index of the root is 1 and the production states can have at most D. Moreover, the internal states do not need to have the same number of sub-states (in Figure 4.10 the D value is 4)
An HHMM is also characterized by the state transition probability between the internal states and the output distribution vector of the production states.
For each internal state q_i^d there is a state transition probability matrix $A^{q^d} = (a_{ij}^{q^d})$, with:

$$a_{ij}^{q^d} = P(q_j^{d+1} | q_i^{d+1}) \qquad (4.3)$$

is the probability of making a horizontal transition from the *ith* state to the *jth*, which are both substates of q_d.

Considering the same formalism

$$\Pi^{q^d} = \{ \pi^{q^d}(q_i^d) \} = P(q_i^{d+1} | q^d) \qquad (4.4)$$

that is defined as the initial distribution vector over the sub-states of q^d, which is the probability that state q^d will initially activate the state q^{d+1}, and in the case that q^d is an internal state this probability can be considered as the probability of a vertical transition.
In addition, each production state q_D is exclusively defined by its output probability vector

$$B^{q^D} = \{ b^{q^D}(k) \} = P(v_k | q^D) \qquad (4.5)$$

that is the probability that the production state q^D will produce the symbol v_k.

An example of HHMM, with horizontal and vertical probabilities transitions, on four levels is presented in Figure 4.10.

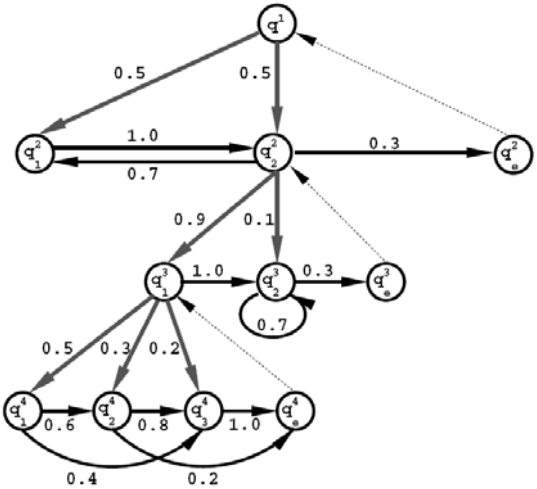


Figure 4.10: Example of HHMM of four levels [FST98]

One example of application of hierarchical hidden Markov Models for representing the grammatical structure of the sentences, which are extracted from texts about scientific literature, is presented in [SCR03]. Authors use machine-learning methods to induce models for extracting relation instances from biomedical articles using a shallow parser to construct a multi-level representation of each sentence being processed. The training process of these hierarchical HMMs captures the regularities of the parsers for both positive and negative sentences.

Fine [FST98] in order to resolve the complex multi-scale structures that characterize natural language, such as speech [RaJ86], handwriting [NWF86], and text used HHMMs. In these works the main idea is to allow HHMMs to correlate structures that are arbitrarily far apart and to handle the statistical inhomogeneities for different sub-models. The idea of using sub-models has been applied to identify frequent letter combinations, punctuation and the ending of sentences in natural language processing.

Starting from the Fines' work, in [SCR03] HHMM is described as multiple "levels" of HMM states, where the lower levels represent

each individual output symbol, and upper levels represent the combinations of lower level sequences.
A further extension of HHMM is introduced in [BPV04] that has presented a general HHMM where the state hierarchy can be a lattice allowing the arbitrary sharing of a sub-structure at the lower levels of the model. This work [BPV04] extends the tree structure, defined by Fine [FST98], which does not allow the sharing of common sub-structures in the model.

4.4.4 Examples of applications of approximation methods

Approximation methods are widely applied in the Literature for solving ambiguous interpretation of the user's input. This section shows some examples of approaches that are mainly based on approximation techniques.
In particular, this section shows how these techniques are applied in:

- Graph-based approaches;
- Finite state mechanisms;
- Formal theory of context;
- Parse trees-based approaches.

4.4.4.1 Graph-based approaches

Starting from the description of Lexical ambiguities (Chapter 2) this section underlines how attributed relational graphs introduced by [CHZ04] deal this kind of ambiguities. In this approach, the speech graph is considered as the referring graph (Figure 4.11A), while gesture graph (Figure 4.11B) and the history graphs (Figure 4.11C) are combined in the referent graph adding new edges to connect every gesture nodes to all history nodes.

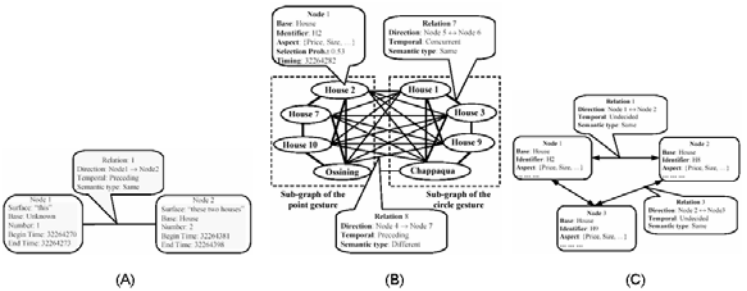


Figure 4.11: Examples of the speech, the gesture and the history graphs [CHZ04]

This approach aims to find the most probable association among referents and referring expression taking into account semantic, temporal and contextual constraints. Therefore, the referent resolution problem is solved finding the best mach between the referring graph and the referent graph and satisfying temporal, semantic and contextual constraints.
The referent resolution problem has been also handled to solve deictic and anaphoric expressions [HCB95].
Modal input can be inaccurate and in particular when user uses gesture modality displayed objects can be too small for human finger so she/he can selects more than one object using only one gesture input generating an ambiguity. To deal this ambiguity [CHZ04] proposes the definition of a history graph that is composed of a list of elements that are in focus during the last interaction of the user. The ambiguities' problems are solved as a graph-matching problem that aims to define the best match between the history graph and graphs generated by modal input optimising the satisfaction of temporal, semantic and contextual constraints.

4.4.4.2 Finite state mechanisms

Lexical ambiguities can be also efficiently dealt using finite-state mechanisms [JoB05] that is based on weighted finite-state automaton with multimodal grammar. This method provided lattice representations for gesture and speech input.

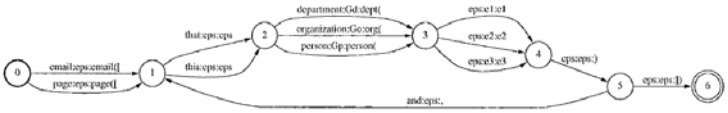


Figure 4.12: Example of finite-state automaton [JoB05]

If the input is ambiguous then the input stream is represented by a lattice that indicates all its possible interpretations. In order to solve ambiguities this method provides a transducer that represents the relationship between gesture and speech, and in particular, it represents the relationships between a particular gesture input stream and all the possible word sequences that could co-occur with the specific input stream. This transducer is composed with gesture interpretations providing a mutual compensation among the input modalities.

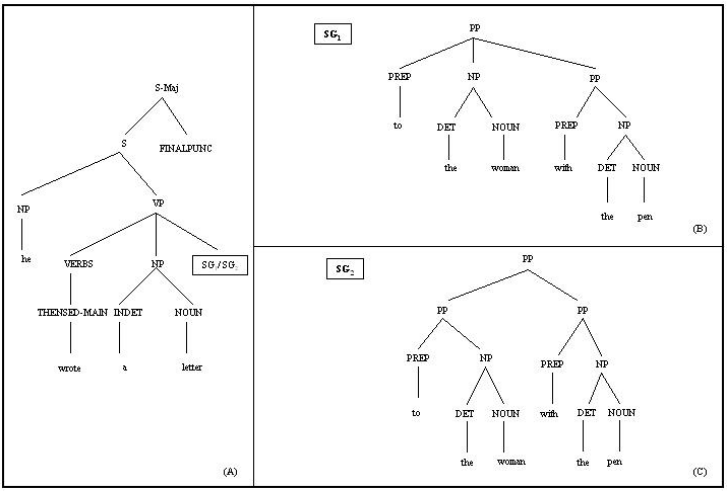
4.4.4.3 Formal theory of context

A further method for dealing Lexical ambiguities is provided in [Buv96] and it is based on a Formal Theory of Context. This approach uses the logic of context representing fact about the context and reasoning with context. This method uses first order structures to describe what is true in the context. These structures describe two types of context: knowledge base context that refers to possible states n the context; and discourse context that refers to particular states in the discourse. In particular, discourse states consist of: states that refer to facts defined in the discourse or that are known in the discourse; and interpretations of predicate symbols in the discourse context considering them according to predicate symbols in knowledge base context.
In the case that a predicate symbol is interpreted differently in different discourse contexts, so ambiguities appear. Therefore ambiguities are related to interpretations of predicate symbols in the discourse context. For solving ambiguities this approach uses common sense knowledge or it directly asks users what is the particular meaning of his/her input.

4.4.4.4 Parse trees-based approaches

Considering syntactical ambiguities, they are connected with the structural properties of the user input and in particular and they are mainly detected analysing the structure of possible parse trees of the user input [Col97].
Syntactic ambiguities are dealt integrating a semantic construction process into a parser analysing the structural properties of the user input and mapping parse trees to different logical representations [Har94].
Considering an example of attachment ambiguity when the user says the sentence “he wrote a letter to the woman with the pen” using the speech modality, the system is not able to decide between these cases: 1) in the first interpretation “with the pen” is attached to the verb “wrote” (Table 4.1B); 2) in the second interpretation “with the pen” is attached to the element “woman” (Table 4.1C). These two interpretations define different meaning for the sentence because in the first case, the man is using a pen to write a letter; while in the second, the woman has the pen.

Table 4.1: Two possible parse trees for he wrote a letter to the woman with the pen



In this case this approach produces a number of parse trees connected with the possible interpretations of the user input (Table 4.1); therefore a problem of this approach is the quite great number of parse trees to represent. To overcome this problem Literature provides a method based on highest preference choice [AlC92] that selects the most likely parse that cannot necessarily be the correct parse.

To avoid the issue connected with the great number of possible parse trees the Description Theory has been proposed [Hir87]. This approach does not provide a whole description of the trees but a partial one representing only those relations that are common to all consistent trees. However this approach is only syntactical and it has to be combined with a semantic interpreter that recognizes if inputs, which have the same structure, are different at the semantic level.

Moreover to overcome problems connected with the great number of structural representations of the user input the shared-packed parse forest approach has been proposed [Har94]. This method stores all parses of user input in a compact form using a data structure that is defined by terminal and non-terminal nodes. These

nodes contain lists of node numbers of the children that build a parse of that constituent and, in particular, nodes, which participate in multiple parses, have multiple arcs that enter in the node. This structure allows detecting ambiguities by traversing the forest of trees analysing the paths in the structure.

4.5 Conclusions and discussion

The analysis of methods, for representing and managing processes characterized by ambiguity and uncertainty, has underlined the suitability of different methods to face the ambiguity problem, underlining the relevance of approximation methods for dealing with uncertainty, which is characteristic in natural interaction approaches.
This chapter, starting from a description of the main different methods described by the literature to solve ambiguities, concludes describing some examples of applications of approximation methods.
Starting from ambiguities arising in the interaction process and their solutions strategies, some repetition methods and all approximation methods are useful applicable to solve ambiguities arising in the multimodal interaction process.
Hidden Markov Models (HMMs) for disambiguation process of multimodal ambiguities can be useful as they have the flexibility for dealing with different levels during the interpretation process and they allow facing the difficulties of combining different kinds of information.
Considering information at different levels is a focal point that matches the aim of this thesis to deal with different classes of ambiguities, which can appear both at syntactic and lexical levels.
The HHMM approach well than HHM or Bayesian Networks (BN) models the different stochastic levels and length scales that characterise natural language, speech, handwriting, and text. Similarly they are used in this thesis dissertation to model multimodal ambiguities and their solutions.
Chapter 5 presents how this thesis uses Hierarchical Hidden Markov Models (HHMM) to represent the semantic and the

syntactic classes of ambiguities for multimodal sentences (described in Chapter 3). In fact, this model permits the semantics tagging of each element of the sentence and to identify the correct syntactic tree starting from the syntax graph of the ambiguous multimodal sentence.
Indeed, HHMM can be used to extract, in a similar manner respect to the Natural Language Processing, a multilevel representation of a multimodal sentence, and using a training process to solve the ambiguities of sentences. This approach operates at different levels of modelling: from the terminal elements level to the multimodal sentence level. These different levels are due to the fact that they operate on the combination of complex information conveyed by different modalities in a multimodal sentence.

Chapter 5

Multimodal Ambiguities Resolution

5.1 Introduction

This chapter copes with the problem to define the correct interpretation of a Multimodal Sentence, i.e. an unambiguous and a meaningful interpretation. It deals with the resolution of the syntactic and semantic ambiguities analysing the syntax-graph and the terminal elements of a Multimodal Sentence, which have been introduced in the Chapter 3. Syntactic ambiguities can produce multiple interpretations of a Multimodal Sentence connected with multiple alternative paths on the syntax-graph; the resolution of this ambiguity implies selecting one multiple paths on the syntax-graph of the ambiguous Multimodal Sentence.
In particular, this chapter discusses how HHMMs match the goal of this thesis modelling repeated processes and sub-processes related to the disambiguation of Multimodal Sentences.
The presented method has been described for the analytic ambiguity and lexical ambiguity, respectively as an example of syntactic and semantic ambiguity.
In particular, considering syntactic ambiguities, methods of solution using HHMMs permit to select the syntax tree of the Multimodal Sentence starting from the syntax graph of the ambiguous Multimodal Sentence.
Considering semantic ambiguities connected with a Multimodal Sentence, these models work on the aspect to assign the most probable sense to the elements of the Multimodal Sentence. This second aspect has been considered in literature on Natural Language Processing as a problem of word-sense disambiguation;

it has been dealt using a probabilistic model of the interdependences among components of the language and a set of input features, such as for example in [OWB00]. This problem is coped assigning to each component (i.e. each word for in natural language sentences) the sense tag that has the highest estimated probability in the given context. For dealing issues of univocally assigning the sense tag this thesis uses a HMM; it identifies the most probable sense tag for each element corresponding to the concept connected with it. In detail, the estimation probability is calculated using information about the context.
In the following sections the model, which describes and manages the different classes of multimodal ambiguities, is presented.

5.2 The HHMMs-based disambiguation method for Multimodal Sentences

In several works on HMMs for natural language interpretation, sentences to be processed are considered as sequences of tokens. Using a similar approach, the present thesis models a Multimodal Sentence as a sequence of tokens that involves the syntax graph and the elements of Multimodal Sentences defined in Chapter 3. As described in the Chapter 3, when a Multimodal Sentence is ambiguous there are two or more different sentences in natural language that can correspond with the user's input forming the candidate interpretations. Therefore, the natural language sentences related to each Multimodal Sentence are combined into a syntax-graph that incorporates the syntactic roles defined by the natural language parser. Therefore, in this section the syntactic roles of the syntax-graph connected with the ambiguous Multimodal Sentence are considered.
The present thesis uses HHMMs for taking into account of semantic and syntactic information to represent and manage ambiguities of Multimodal Sentences.
In particular, semantic information stand for a sequence of concepts connected with the elements of the language belonging to the Multimodal Sentence. Syntactic information stands for a sequence of syntactic roles of the language belonging to the syntax-graph.

A HHMM incorporates semantic and syntactic information from the syntax-graph and concepts associated to its elements and enable to provide a detailed description of processes and sub-processes that characterize the identification of the un-ambiguous interpretation.

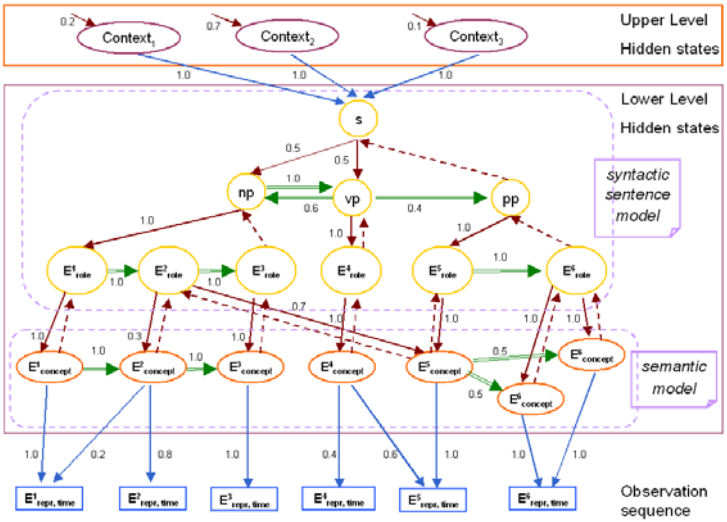


Figure 5.1: An example of an HHMM for a Multimodal Sentence

The proposed representation contains three connected models of a Multimodal Sentence, the *context model*, the *semantic model* and the *syntactic sentence model* (Figure 5.1), respectively structured in an *Upper Level* and a *Lower level*, which are over the Observation sequence.

The *Upper level* consists of a HMM that associates the context (i.e. what the Multimodal Sentence concerns) to the Multimodal Sentence according to the sequences of concepts that the sentence contains

The *Upper level* affects the resolution process of the semantic ambiguity, because the knowledge of the different meanings that can be associated with the ambiguous words, their sequences as well as the typical contexts in which they occur is necessary. This context is obtained by the *context HMM* (Upper level) used to

associate each terminal element of the Multimodal Language with a semantic tag representing the meaning of the element. In section 5.2.1.1 this chapter will present the example 1, where a corpus of Multimodal Sentences will be referred to two different contexts (in the example to the context of *transportation system* and to the context of *water system*). The states of the context HMM are showed by ovals ( ).

Starting from considering that the meaning of each Multimodal Sentence is represented as a sequence of concepts connected with terminal elements of a Multimodal Language, each Multimodal Sentence can be achieved by HMM (*semantic model)* that refers to concepts connected with terminal elements and that can be defined as *semantic elements HMMs* and their states as *semantic element states* ($E^i_{concept}$) as in Figure 5.1.

In particular, *semantic element states* have observation sequences ($E^i_{repr,\ time}$), i.e. their representation and temporal information as direct emissions.

The presented hierarchical HMMs splits a Multimodal Sentence into disjoint observations $E^i_{concept}$ where each $E^i_{concept}$ is the concept connected with the *ith* element. Each Multimodal Sentence represents a sequence of observations associated with all the attributes (except for *concepts* and *roles* that are modelled with nodes of the HMM at syntactic and semantic level) of each terminal element of the Multimodal Language. In particular, according to the definition provided in (section 3.3.1.1) the two attributes considered in the following examples are *representation* and *time* ($E^i_{repr,\ time}$). Each observation $E^i_{repr,\ time}$ consists of a representation of the concept $E^i_{concept}$ connected with the *i* element and time information E^i_{time}, which enables to consider the temporal relations among the different elements in the ambiguity solution process.

The *syntactic sentence model* represents each sentence as a sequence of syntactic roles and they are the internal states of the HHMM; the *semantic model* represents concepts connected with the terminal elements that are the production states of the HHMM.

The *syntactic sentence model* contains states that emit the syntactic structure of the Multimodal Sentence; therefore these states are the syntactic roles of the syntax-graph, and the *syntactic sentence states* (E^i_{role}).

In this work *syntactic sentence states* identify the syntactic roles that each element can have in the Multimodal Sentence. These roles are included in the syntactic roles of the syntax-graph that are defined in Chapter 3 as tag symbols of the Penn Treebank Tag set [MSM94].

The semantic element states are depicted with ovals and the syntactic sentence states in Figure 5.1 are depicted with circles. Labels of each semantic element state are shown in the oval connected with the element state, and the syntactic roles of the Multimodal Sentence are shown into the circles.

The observation sequence is depicted with rectangles that contain the label of each representation of the element, and attributes (different from roles and concepts) represented in the Multimodal Language such as the temporal attributes.

Figure 5.1 contains several paths that characterize the HHMM: 1) paths that correspond to *forward* and *backward* transitions (Figure 5.1) that are defined by the edges in green (⟹) and represent the probability of making a horizontal transition; 2) path variables that define *downward* and *upward* transitions (Figure 5.1) that are defined by the edges in red (⟶) that represent the initial distribution vector over the sub-states and the probability for a vertical transition; 3) the edges in blue represent the probability that each production state produces a symbol of the observation.

This structure can be efficiently used to solve classes of multimodal ambiguities defined in the Chapter 3.

The *semantic model* and *syntactic sentence model* fit for the resolution of the *multimodal semantic ambiguities,* and the *syntactic sentence model* answers to the need of solving the classes of *multimodal syntactic ambiguities*.

In particular, this approach based on multi-levels stochastic process using HHMMs can be applied for dealing with semantic multimodal ambiguities, because it provides a multilevel description of a Multimodal Sentence from the terminal elements interpretation to the Multimodal Sentence interpretation and its context (see Figure 5.1), connecting the different Hidden Markov models.

Classes of syntactic ambiguities concern different candidate structure of the Multimodal Sentence and they refer to the possible different paths on the syntax-graph. For this reason the problem

consists of identifying the correct path on the syntax-graph and consequently the focus is only moved on the *syntactic sentence model*. In fact, in this case the disambiguation process consists of identifying univocally the sequence of the *syntactic sentence states*. The following sections details how these models can be trained and used to solve multimodal ambiguities.

5.2.1 Estimating the disambiguation method parameters and identifying un-ambiguous sentences

Identifying an un-ambiguous interpretation for a given Multimodal Sentence consists of two main steps: the first one is the step of training the HHMM of the ambiguous sentence that enables estimating the disambiguation parameters, while the second step consists of identifying the most probable sequence of states in the HHMM that is associated with the correct interpretation of an ambiguous Multimodal Sentence.

The training process of the *Upper* and the *Lower* levels is firstly described. Let be given the set *U* of users, a set *SM* of Multimodal Sentences, a set *I* composed of the *IC* sets of candidate interpretations for each ambiguous Multimodal Sentence belonging to *SM*.

Let be given a Multimodal Sentence belonging to SM and one only interpretation. Each user is required to input that sentence (thinking to the assigned meaning). The request for the input is repeated for each meaning of the Multimodal Sentence, for each Multimodal Sentence belonging to SM and for all users belonging to the U set. This process enables to observe and to set all parameters characterising the multimodal interaction according to the different candidate interpretations and meanings for the Multimodal Sentences.

The *context HMM* (*Upper level*) is learned using corpus of Multimodal Sentences, tagging each element representation of E^i with its meaning, and then referring it to its context. The context for each concept can be identified considering WordNet taxonomy. When tagging a terminal element representation, then the

probability to give that meaning to a representation in the sequence of terminal elements forming a Multimodal Sentence is updated. This problem will be better explained in the example 1 of this chapter (see section 3.2.1.1) and in the learning process described in Chapter 7.

Training the HHMMs (*Lower level* in Figure 5.1) for ambiguous Multimodal Sentences involves the sequence of elements of that Multimodal Sentences. During this process training instances for these hierarchical HMMs are sequences of attributes, taking into account of representations, of temporal information and all the attributes of Multimodal Sentences that characterize the user's input. The disambiguation problem can be faced training the model in order to univocally define paths for reaching each syntactic role of syntax-graph elements and their concepts.

In detail, for training models at *Upper* and *Lower* levels in order to solve a class of ambiguities, an extension of the Baum-Welch [FST98] algorithm is used. The Baum-Welch training algorithm in its original formulation [BaT66] has the purpose to define model parameters in order to maximize the conditional likelihood of the syntactic structure and the semantic of the elements of the sentence, given their representation.

The multi-level nature of the applied HHMM implies that the parameters of this method are complex to estimate, as it is necessary to calculate the observational and transitional probabilities for each level redefining the transition matrix by including estimated values for each level. This procedure implies to calculate the observation distribution for each level and, then, redefining the transition matrix by including estimated values for each level.

The hierarchical structure of the HHMM implies that several paths have to be considered: paths that correspond to *forward* and *backward* transitions (green in Figure 5.1) and other path variables that define *downward* and *upward* transitions (red in Figure 5.1).

The training procedure uses the Maximum Likelihood Estimation (MLE) method for fitting a mathematical model to data captured during the learning process for tuning the parameters of the model to obtaining probabilities. For the maximum-likelihood parameter estimation procedure of HHMMs a generalization of the Baum-Welch algorithm (Table 5.1) is applied as defined in [FST98]; they

imply stochastic horizontal and vertical transitions which recursively generate observations.

Table 5.1: Generalized Baum-Welch Algorithm steps

ALGORITHM: BAUM-WELCH:

Initialisation: Pick arbitrary model parameters
•**Recurrence**:
–Set all the A and E variables to their pseudocount values r (or to 0)
–For each Sequence j=1..n:
•Calculate for sequence j using the forward algorithm
•Calculate for sequence j using the backward algorithm
•Add the contribution of sequence j to A and E
–Calculate new model parameters
–Calculate new log likelihood of the model
•**Termination**:
–Stop if the change in log likelihood is less than some predefined threshold or the maximum number of iterations is exceeded

In particular, the *semantic model* is trained to recognize the correct concepts of the Multimodal Sentences ($E^i_{concept}$). The *syntactic sentence model* is trained to recognize the correct sequence of the syntactic roles (E^i_{role}) in the Multimodal Sentences.
During this training process the algorithm is trained considering the *syntactic sentence states* as internal states, the *semantic element states* as production states and the representations of the elements as observation sequence.
This model is trained in order to solve each class of multimodal ambiguities introduced in the Chapter 3. Each class of multimodal ambiguities has a HHMM that is able to solve it and has its general schema presented in Figure 5.1.
After the training of the model, the second step to define the correct interpretation of ambiguous Multimodal Sentences has to identify the most probable sequence of the HHMM states; the Viterbi algorithm [Vit67] (Table 5.2) is used to this purpose i.e. the most probable interpretation.
In particular, a correct interpretation of a Multimodal Sentence that contains one of the classes of the semantic and syntactic ambiguity is extracted considering a given observation sequence of representations and temporal intervals of its elements and obtaining

the most probable path among the concepts and the syntactic roles of its elements applying the Viterbi algorithm on the model connected with the ambiguity.

Table 5.2: Viterbi Algorithm

ALGORITHM: VITERBI

```
def forward_viterbi(obs, states, start_p, trans_p,
emit_p):
   T = {}
   for state in states:

       ##          prob.           V. path  V.
prob.

       T[state] = (start_p[state], [state],
start_p[state])
   for output in obs:
       U = {}
       for next_state in states:
           total = 0
           argmax = None
           valmax = 0
           for source_state in states:
               (prob, v_path, v_prob) =
T[source_state]
               p = emit_p[source_state][output] *
trans_p[source_state][next_state]
               prob *= p
               v_prob *= p
               total += prob
               if v_prob > valmax:
                   argmax = v_path + [next_state]
                   valmax = v_prob
           U[next_state] = (total, argmax, valmax)
       T = U

   ## apply sum/max to the final states:

   total = 0
   argmax = None
   valmax = 0
   for state in states:
       (prob, v_path, v_prob) = T[state]
       total += prob
       if v_prob > valmax:
           argmax = v_path
           valmax = v_prob
   return (total, argmax, valmax)
```

The output of the model for the disambiguation process of the semantic ambiguities is the sequence of roles and concepts contained in the Multimodal Sentences with the highest rank among all sentence models tighter with its score value. Obtaining a sequence of concepts can imply resolving lexical ambiguities, i.e. ambiguities of meanings for singular elements. It is a problem of word sense disambiguation. This dissertation deals with this problem considering it as a semantic tagging problem.
The output of the model for the disambiguation process of the syntactic ambiguities is the sequence of syntactic roles contained in the Multimodal Sentences with the highest rank among all sentence models tighter with its score value and their connected concepts.
To clarify the disambiguation process two examples will be given in the next sections.

5.2.1.1 Example 1

The first example considers the lexical ambiguity presented in the Chapter 3. This specific structure is trained by a training set of examples of the same class of ambiguities.
Suppose a user is interacting with a map by sketch and speech. Using the speech modality the user says:

 *"show this in Rome"*

and the user simultaneously draws the sketch in Figure 5.2 that is the element of the language connected with two different concepts, respectively street and river:

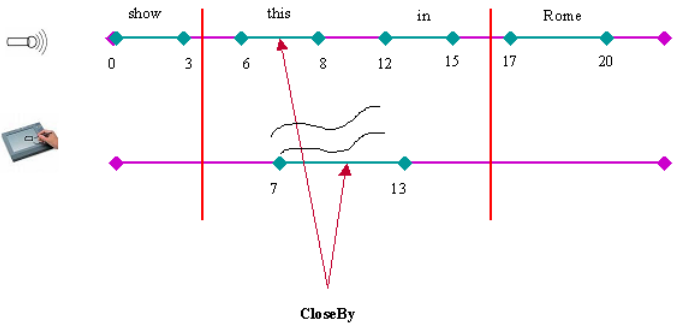


Figure 5.2: Elements that compose the Multimodal Sentence

As explained in Chapter 3, the sketch of Figure 5.2 can be interpreted both, as a river and a street.
And the syntax-graph connected with this Multimodal Sentence is the following:

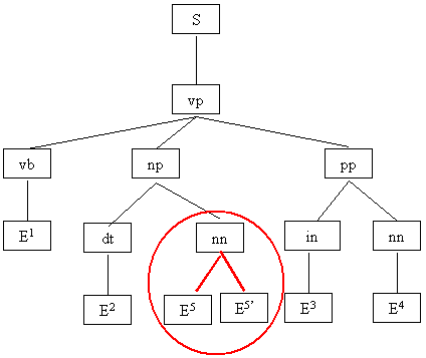


Figure 5.3: syntax-graph of the user's input defined by the example of lexical ambiguity

Here elements defined by the speech modality are:

- E^1 *is* $!\ (E^1_{mod}=speech) \otimes !\ (E^1_{repr}=$  *"show")) ⊗ !* $(E^1_{time}=(0,2)) \otimes !\ (E^1_{concept}=(verb)) \otimes !(E^1_{role}=(vb))$

- E^2 *is* ! (E^3_{mod}=*speech*) ⊗ ! (E^2_{repr} =  *"this"*)) ⊗ ! (E^2_{time}=(5,7)) ⊗ ! ($E^2_{concept}$=(deictic)) ⊗ !(E^2_{role}=(dt))
- E^3 *is* ! (E^3_{mod}=*speech*) ⊗ ! (E^3_{repr} =  *"in"*)) ⊗ ! (E^3_{time}=(11,12)) ⊗ ! ($E^3_{concept}$=(adverb)) ⊗ !(E^3_{role}=(in))
- E^4 *is* ! (E^4_{mod}=*speech*) ⊗ ! (E^4_{repr} =  *" Rome"*)) ⊗ ! (E^4_{time}=(15,18)) ⊗ ! ($E^4_{concept}$=(city)) ⊗ !(E^4_{role}=(nn))

and elements defined by the sketch modalities are:

- E^5 *is* ! (E^5_{mod}=*sketch*) ⊗ ! (E^5_{repr} = )) ⊗ ! (E^5_{time}=(7,13)) ⊗ ! ($E^5_{concept}$=(river)) ⊗ !(E^5_{role}=(nn))
- $E^{5'}$ *is* ! ($E^{5'}_{mod}$=*sketch*) ⊗ ! ($E^{5'}_{repr}$ =)) ⊗ ! ($E^{5'}_{time}$=(7,13)) ⊗ ! ($E^{5'}_{concept}$=(street)) ⊗ !($E^{5'}_{role}$=(nn))

As defined in Chapter 3, the alignment of the element E^2 with the element E^5 detects a lexical ambiguity due to the fact that the element E^5 can have two different meanings, river (E^5) and street ($E^{5'}$), according to the two different contexts (*Water system Context* and *Transportation system Context*).
In this example the two possible interpretations are:

- "show this river in Rome"
- "show this street in Rome".

This lexical ambiguity is related to the lower level of the HHMM (*semantic model*- Figure 5.1) and it is trained for solving them.
In this example the set of hidden states is defined by:

- Internal states *SS* = {s, vp, np, pp, vb, dt, nn, in, nn}
- Production states *LS* = {show, this, river, street, in, Rome}

The observation sequence of this sentence is:

O_r= 

In the Figure 5.4 there are also represented the values of the transition probabilities defined by $A^{q^d} = (a_{ij}^{q^d}) = P(q_j^{d+1}|q_i^{d+1})$ (see section 4.4.3.5) that gives the probabilities of making a horizontal transitions for each q_i^d where *i* is the state index and *d* is the hierarchy index.
Moreover, Figure 5.4 shows the initial distribution probabilities over the sub-states of q^d that represents the probability that state q^d will initially activate the state q^{d+1}, and the output probabilities.
The probabilities contained in the matrices of the HHMM are calculated starting from the information contained in the syntax-graph connected with the ambiguous Multimodal Sentence, and the temporal alignments of the temporal intervals of the elements. If this specific class of ambiguity appears for the first time and there is not meaningful information for defining the most probable interpretation, then the alternative sequences of states, which define the two different interpretations, have the same probability.
The Figure 5.4 shows that before the training the element E^5 has a probability equal to 0.5 to be interpreted as "*river*" and a probability equal to 0.5 to be interpreted as "*street*". The probabilities connected with the concepts "*river*" and "*street*" are connected with the probabilities to associate the sentence to the two possible contexts, *Water system* and *Transportation system Contexts* showed on the Upper level of Figure 5.4.

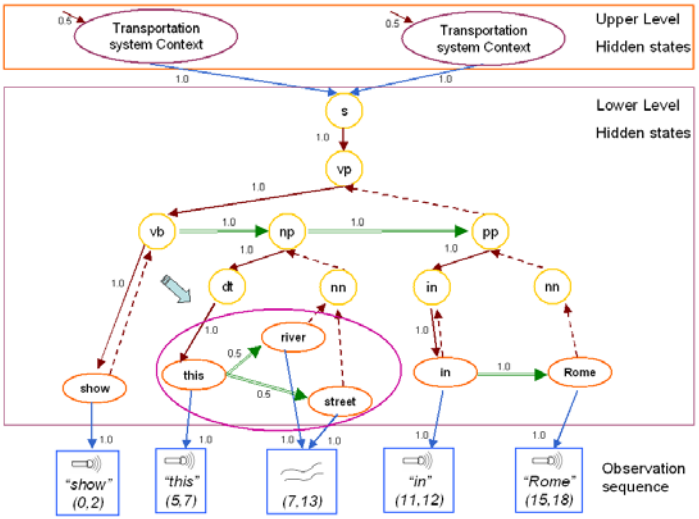


Figure 5.4: Initial HHMMs defined by the example of lexical ambiguity

The probabilities defined in Figure 5.4 are updated using a training set of sequences that change the probabilities.
In detail, the Baum-Welch Algorithm updates these probabilities during the learning process of the model using other observation sequences that are positive examples (correct interpretations) for this class of multimodal ambiguities.
The Figure 5.5 shows the updated HHMM for lexical ambiguity.

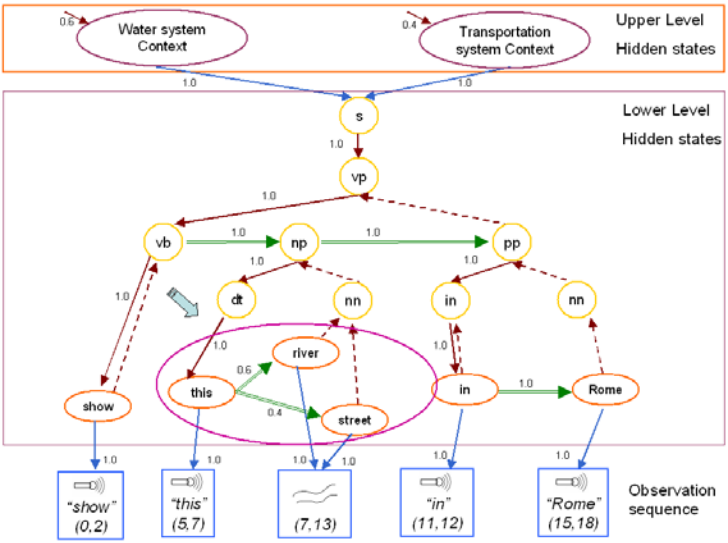


Figure 5.5: Updated HHMMs defined by the example of lexical ambiguity

The transition matrix for the HHMM in Figure 5.5 is expressed in the following table:

Table 5.3: Transition matrix of the HHMMs defined by the example of lexical ambiguity

	s	**vp**	**vb**	**np**	**pp**	**dt**	**nn**	**in**	**nn**	**show**	**this**	**river**	**street**	**in**	**Rome**
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
vp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
vb	0	0	0	1.0	0	0	0	0	0	0	0	0	0	0	0
np	0	0	0	0	1.0	0	0	0	0	0	0	0	0	0	0
pp	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dt	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
in	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
show	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
this	0	0	0	0	0	0	0	0	0	0	0	0.6	0.4	0	0
river	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
street	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
in	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0
Rome	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Considering the initial distribution matrix and therefore the probability of making a vertical transition, the Table 5.4 shows the values of this matrix defined for lexical ambiguity of the given example.

Table 5.4: Initial distribution matrix of the HHMM defined by the example of lexical ambiguity

	s	**vp**	**vb**	**np**	**pp**	**dt**	**nn**	**in**	**nn**	**show**	**this**	**river**	**street**	**in**	**Rome**
s	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0
vp	0	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0
vb	0	0	0	0	0	0	0	0	0	1.0	0	0	0	0	0
np	0	0	0	0	0	1.0	0	0	0	0	0	0	0	0	0
pp	0	0	0	0	0	0	0	1.0	0	0	0	0	0	0	0
dt	0	0	0	0	0	0	0	0	0	0	1.0	0	0	0	0
nn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
in	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0	0
nn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0
show	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
this	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
river	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
street	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
in	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rome	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The output probability matrix has values equal to zero for internal states, as shown in the Table 5.5, because the production states are the only states that emit output symbols. This matrix represents the probability that each production state associated with E_{conc} will produce the E_{repr} and E_{time} of the terminal element.

Table 5.5: Matrix of the production probability defined by the example of lexical ambiguity

	 "show" (0,2)	 "this" (5,7)	 (7,13)	 "in" (11,12)	 "Rome" (15,18)
s	0	0	0	0	0
vp	0	0	0	0	0
vb	0	0	0	0	0
np	0	0	0	0	0
pp	0	0	0	0	0
dt	0	0	0	0	0
nn	0	0	0	0	0
in	0	0	0	0	0
nn	0	0	0	0	0
show	1.0	0	0	0	0
this	0	1.0	0	0	0
river	0	0	1.0	0	0
street	0	0	1.0	0	0
in	0	0	0	1.0	0
Rome	0	0	0	0	1.0

Considering the matrices of this example, they are obtained by the training process using the Baum-Welch Algorithm. Table 5.3 shows that the element E^5 has a probability equal to 0.6 to be interpreted as "*river*" and it has a probability equal to 0.4 to be interpreted as "*street*".

For every internal and production state q_i and observation sequence, the single best sequence can be found using the Viterbi algorithm, which maximises $P(O|Q,\lambda)$.

The output of the method is the correct sequence of the internal states, given the observation sequence.

The output of the model is the sequence of concepts contained in the Multimodal Sentences with the highest rank among all sentence models tighter with its score value.

In this case the output of the Viterbi algorithm is the sequence that defines the following interpretation:

"*show this river in Rome*"

because the connected sequence maximises $P(O|Q,\lambda)$.

5.2.1.2 Example 2

Considering the example of analytic ambiguity provided in the Chapter 3 where the user says the sentence:

"***show Italian river***"

and immediately after she/he write the word "name" (name) using the handwriting modality (Figure 5.6).

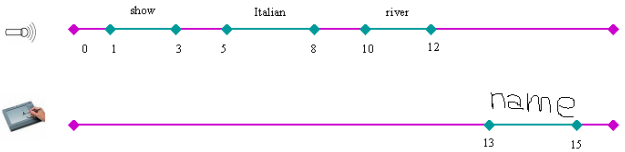


Figure 5.6: Elements that compose the Multimodal Sentence

The Multimodal Sentence is composed by the following elements defined by the speech modalities:

- E^1 *is* ! $(E^1_{mod}$=*speech)* ⊗ ! $(E^1_{repr}$= "*show*")) ⊗ ! $(E^1_{time}$= (1, 3)) ⊗ ! $(E^1_{concept}$=(verb)) ⊗ !$(E^1_{role}$=(vb))
- E^2 *is* ! $(E^2_{mod}$=*speech)* ⊗ ! $(E^2_{repr}$ ="*Italian*")) ⊗ ! $(E^2_{time}$=(5,8)) ⊗ !$(E^2_{concept}$=(adjective)) ⊗ !$(E^2_{role}$=(jj))
- E^3 *is* ! $(E^3_{mod}$=*speech)* ⊗ ! $(E^3_{repr}$ ="*river*")) ⊗ ! $(E^3_{time}$=(10,12)) ⊗ !$(E^3_{concept}$=(river)) ⊗ !$(E^3_{role}$=(nn))

While the element defined by the handwriting modality is:

- E^4 *is* ! $(E^4_{mod}$=*handwriting)* ⊗ ! $(E^4_{repr}$ =name)) ⊗ ! $(E^4_{time}$=(13,15)) ⊗ ! $(E^4_{concept}$=(name)) ⊗ !$(E^4_{role}$=(nn))

As explained in the Chapter 3 the element E^2 can be associated both to the element E^3 and E^4 because the Multimodal Sentence can be interpreted as:

1) "show the Italian name of the river"; and
2) "show the name of the Italian river"

The corresponding graph tree is showed in Figure 5.7.

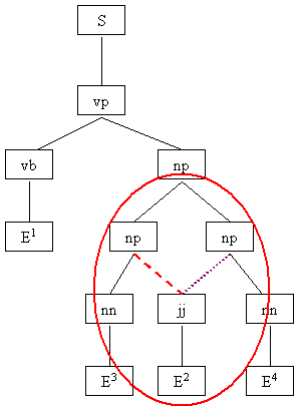


Figure 5.7: Syntax-graph associated to the Multimodal Sentence that defines the example of analytic ambiguity

In the syntax-graph obtained by this Multimodal Sentence (Figure 5.7) there is more than one edge that reaches the element E^2. In fact, the element E^2 has two different roles in the syntax-graph because there are two different paths that allow reaching this element.
The following figure represents Hierarchical HMMs for an analytic ambiguity.
The analytic ambiguity is a syntactic ambiguity, thus it is related to the Upper model of the HHMM (*syntactic sentence model*- Figure 5.1) that is trained for solving them.

In this example the set of hidden states is defined by:

- Internal states *SS* = {s, vp, np, vb, jj, nn, nn}, and

- Production states LS = {show, Italian, river, name}

The observation sequence defined by this sentence is:

O_r= { "show" (1,3) "italian" (5,8) "river" (10,12) name (13,15) }

In Figure 5.8 the transition probability matrix defined by $A^{q^d} = (a_{ij}^{q^d}) = P(q_j^{d+1}|q_i^{d+1})$ for each inner state $q_i^d \in SS \cup LS$ is represented and it defines the probability of making a horizontal transition.
Moreover, in the figure the initial distribution vector $\Pi^{q^d} = \{\pi^{q^d}(q_i^d)\} = P(q_i^{d+1}|q^d)$ is represented and it defines the probability that state q^d will initially activate the state q^{d+1} (vertical transitions).
Finally, for each production state the figure represents its output probability vector $B^{q^D} = \{b^{q^D}(k)\} = P(v_k|q^D)$ that is the probability that the production state E_{conc} will produce the E_{repr} and E_{time} for the terminal element.

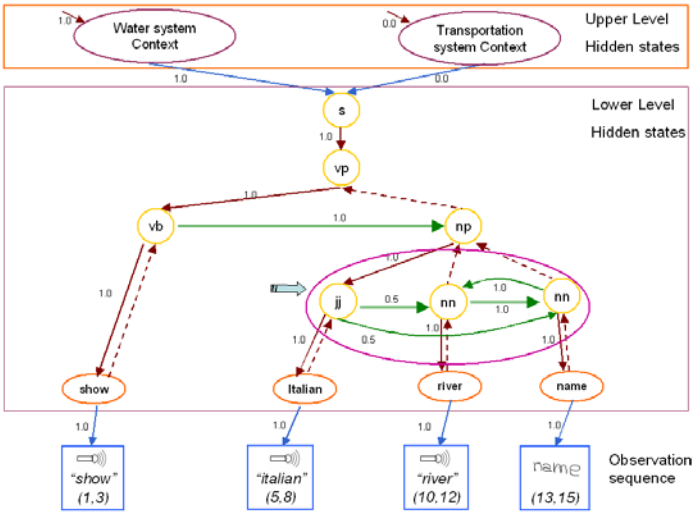


Figure 5.8:Initial HHMMs defined by the example of analytic ambiguity

As explained in the example 1 probabilities contained in the figure are updated during the learning process of the model using other observation sequences that are positive examples for this class of ambiguity.

If the analytic ambiguity appears for the first time and there is not meaningful information to update these probabilities then the alternative sequences of states, which define the two different interpretations, have the same probability.

In fact, Figure 5.9 shows that before the training process the sequence "*Italian river"* has a probability equal to 0.5 and the sequence "*Italian name"* has a probability equal to 0.5.

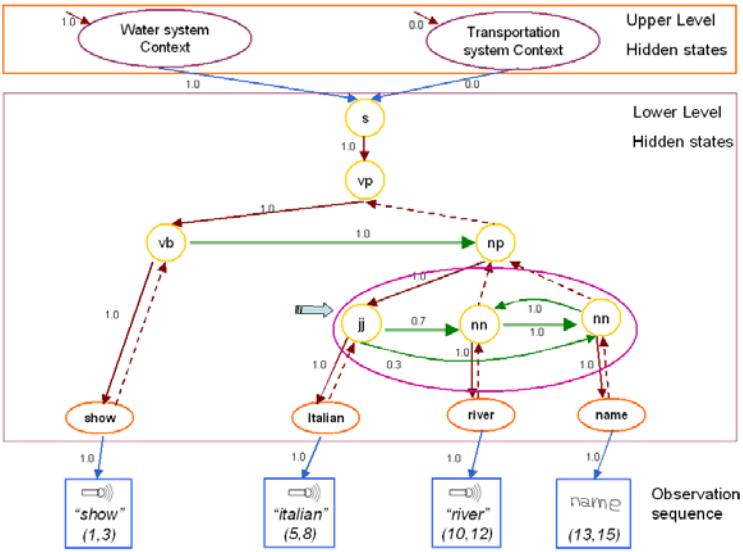


Figure 5.9: HHMMs defined by the example of analytic ambiguity

The probabilities, defined in Figure 5.9, are modified using a training set of sequences that change the probabilities of the sequences. In this example, values in the Table 5.6, Table 5.7 and Table 5.8 are obtained training the model.
In particular, the transition probability matrix is shown in the following table:

Table 5.6: Transition matrix of the HHMM defined by the example of analytic ambiguity

	s	**vp**	**np**	**vb**	**jj**	**nn**	**nn**	**show**	**Italian**	**river**	**name**
s	0	0	0	0	0	0	0	0	0	0	0
vp	0	0	0	0	0	0	0	0	0	0	0
np	0	0	0	0	0	0	0	0	0	0	0
vb	0	0	1.0	0	0	0	0	0	0	0	0
jj	0	0	0	0	0	0.7	0.3	0	0	0	0
nn	0	0	0	0	0	0	1.0	0	0	0	0
nn	0	0	0	0	0	0	0	1.0	0	0	0
show	0	0	0	0	0	0	0	0	0	0	0
Italian	0	0	0	0	0	0	0	0	0	0	0
river	0	0	0	0	0	0	0	0	0	0	0
name	0	0	0	0	0	0	0	0	0	0	0

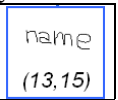
The initial distribution matrix connected with this example is shown in the following table:

Table 5.7: Initial distribution matrix of the HHMM defined by the example of analytic ambiguity

	S	**vp**	**np**	**vb**	**jj**	**nn**	**nn**	**show**	**Italian**	**river**	**name**
s	0	0	0	0	0	0	0	0	0	0	0
vp	0	0	0	1.0	0	0	0	0	0	0	0
np	0	0	0	0	1.0	0	0	0	0	0	0
vb	0	0	0	0	0	0	0	1.0	0	0	0
jj	0	0	0	0	0	0	0	0	1.0	0	0
nn	0	0	0	0	0	0	0	0	0	1.0	0
nn	0	0	0	0	0	0	0	1.0	0	0	1.0
show	0	0	0	0	0	0	0	0	0	0	0
Italian	0	0	0	0	0	0	0	0	0	0	0
river	0	0	0	0	0	0	0	0	0	0	0
name	0	0	0	0	0	0	0	0	0	0	0

Finally, for each production state the figure represents its output probability. Therefore, the states that are not production states have the values of the matrix equal to zero, as shown in Table 5.8.

Table 5.8: Matrix of the production probability defined by the example of analytic ambiguity

	 "show" (1,3)	 "italian" (5,8)	 "river" (10,12)	 name (13,15)
s	0	0	0	0
vp	0	0	0	0
np	0	0	0	0
vb	1.0	0	0	0
jj	0	1.0	0	0
nn	0	0	1.0	0
nn	0	0	0	1.0
show	0	0	0	0
Italian	0	0	0	0
river	0	0	0	0
name	0	0	0	0

After the training, this model is able to define the correct interpretation finding the best sequence using the Viterbi algorithm, which maximises P(O|Q,λ). This method returns the correct sequence of the internal states given the observation sequence.
The output of the model is the sequence of syntactic roles contained in the Multimodal Sentences with the highest rank among all sentence models tighter with its score value. In this case the sequence that maximises P(O|Q,λ) is *"show the name of the Italian river"* because Table 5.6 shows that the sequence "*Italian river"* has a probability equal to 0.7 and the sequence "*Italian name"* has a probability equal to 0.3.

5.3 Discussions

This chapter has proposed an approximation method for coping the problem to identify the correct interpretation of a Multimodal Sentence when a semantic or a syntactic ambiguity arises. The method is proposed and discussed using one example of semantic and one example of syntactic ambiguity: respectively the lexical and analytic ambiguities.

This method this chapter has used a HHMMs to model the disambiguation process as repeated processes and sub-processes.
In particular, ambiguous Multimodal Sentences have been considered as sequences of tokens that include information about the context, the syntax graph and the elements of Multimodal Sentences. Each class of multimodal ambiguities, introduced in the Chapter 3, can be modelled using an instance of the general model proposed. This model consists of an Upper level that permit to identify the context and a Lower level, containing an HHMM that models the syntax and the semantic of the Multimodal Sentence, permitting to represent and solve its ambiguities.
The methods proposed for classifying multimodal ambiguities in the Chapter 3, and in this chapter for solving them have been adopted in the design and implementation of the two modules *Multimodal Ambiguity Classifier* and *Multimodal Ambiguity Solver* that will be described in Chapter 6, and that will be evaluated in Chapter 7.

Chapter 6

Multimodal Ambiguities Classifier and Multimodal Ambiguities Solver Design

6.1 Introduction

This chapter, starting from methods proposed in Chapter 3 and Chapter 5 to classify, detect and solve multimodal ambiguities, describes the design process of the *Multimodal Ambiguities Classifier* and *Multimodal Ambiguities Solver* modules.
The description of the general *MultiModal Language Processing framework* architecture (M2LP) is given; it shows the platform supporting multimodal interaction in which the *Multimodal Ambiguities Classifier* module and the *Multimodal Ambiguities Solver* module are included. The design of the two SW modules is respectively provided in sections 6.3.1 and 6.3.2. An example of use of the two modules is given in section 6.3.3.

6.2 The general MultiModal Language Processing framework architecture

This section aims to provide a vision of the general framework architecture M2LP [DFG08] including the two SW modules (*Multimodal Ambiguities Classifier module* and the *Multimodal Ambiguities Solver*) designed and implemented for this thesis. The (M2LP) framework [DFG08] is a platform that aims to be

integrative, configurable, scalable, and adaptive in order to efficiently manage multimodal communication between people and computational systems.

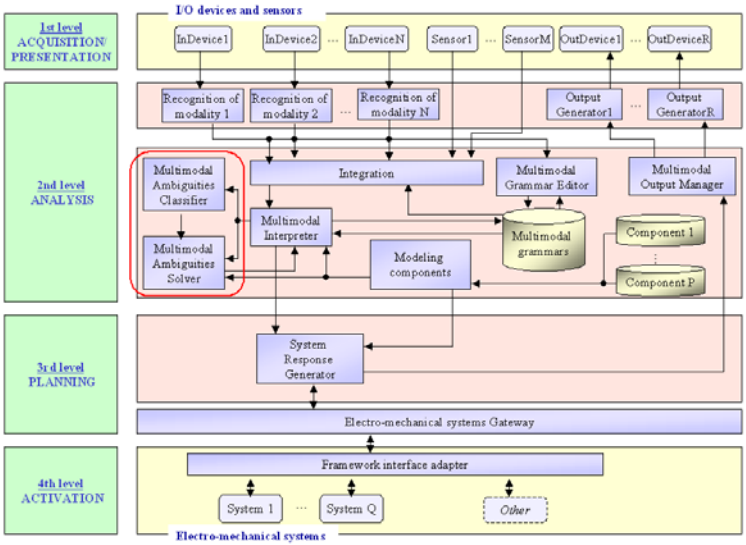


Figure 6.1: Multimodal Platform Architecture

Figure 6.1 shows the M2LP architecture, which consists of the following four different architectural levels:

- The *acquisition/presentation level*: it includes the specific I/O devices (for example, display, cameras, microphone, loudspeakers, and input sensors);
- The *analysis level*: it includes both the unimodal input recognisers (for example the *Automatic Speech Recogniser* and the *gesture recogniser*) and the output generators (for example the *Speech Synthesizer*). This level also contains the *Multimodal Interpreter* component that integrates the recognized inputs, assigning them the appropriate values for the attributes, as required by the multimodal grammar

notation; it applies the production rules stored in the *Multimodal Grammar Repository* and the set of production rules of the grammar through the *Multimodal Grammar Editor*, respectively to interpret Multimodal Sentences and to define new grammars. When a Multimodal Sentence can have more than one interpretation, the Multimodal Interpreter asks to the *Multimodal Ambiguities Classifier and Solver* modules, whose design and development is the focus of this thesis, to detect the class of ambiguity and to solve it. As the interpretation is a complex activity that needs to take into account of many aspects such as context and user information, the Interpretation module and the Ambiguity Solver module are connected with the *Modeling components* module, which capture the contextual features used during the interpretation and disambiguation phases for leading up to the most probable interpretation of the user input (for example user, content and context modeling components). Finally, the *Multimodal Output Manager* defines the generation (multimodal fission) of appropriate output information, through the available output modalities.

- The *planning level*: it is composed by the *System Response Generator* that has the role to plan the better way to react to the user input (either directly intervening on the electro-mechanical systems, through the electro-mechanical systems Gateway, or providing specific audio/visual feedback) and the consequent adaptation of the human-machine interaction, taking into account also of the outputs of the Modeling Components. This level contains also the *Electro-mechanical systems Gateway* that provides the link with the electro-mechanical systems. Proper solutions shall be applied to ensure safe interfacing and communication between the two levels.
- The *activation level*: it is composed by the electro-mechanical components offering specific functionalities to the user. It includes a framework interface adapter offering

specific functions such as communicating to the framework through the electro-mechanical systems gateway.

In Figure 6.1 the red bounded area focuses on the *Multimodal Ambiguities Classifier* and the *Multimodal Ambiguities Solver* modules, which have been designed and implemented in this thesis. Figure 6.2 shoes the Data flow among the *Multimodal Ambiguities Classifier*, the *Multimodal Ambiguities Solver* and the other components of the architecture that directly communicate with them.
The purposes of these modules are:

1) taking as input the Multimodal Sentence and its ambiguous interpretation given by *Multimodal Interpreter*,
2) classifying and solving ambiguities also using information coming from the *Modeling components* module on context and user's modeling.

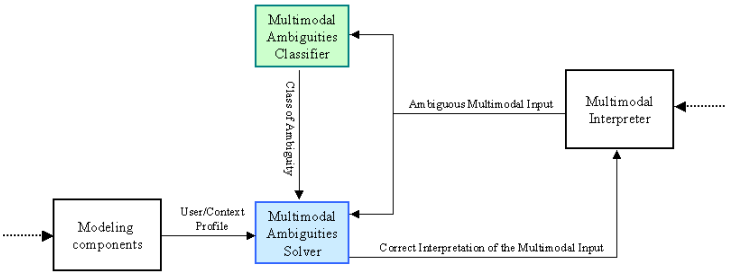


Figure 6.2: Data flow among components

The *Multimodal Ambiguities Classifier* and the *Multimodal Ambiguities Solver* modules have two main goals:

1) to analyse ambiguous interpretation of Multimodal Sentences in order to detect the ambiguity classes they

belong to, according to classes and approaches described in Chapter 3 for classifying multimodal ambiguities;

2) to use the knowledge about the detected ambiguity classes in order to solve ambiguities using the HHMMs models as defined in the Chapter 5.

The following sections present in detail the design and development of the *Multimodal Ambiguities Classifier* and the *Multimodal Ambiguities Solver* modules, and one example of use is provided.

6.3 Design of the Multimodal Ambiguities Classifier and Solver modules

This section describes the Multimodal Ambiguities Classifier and Solver modules, starting from identifying requirements of the SW design and implementation. Once identified these requirements, a general and abstract description of the SW modules and their connections satisfying these requirements has been provided using a UML Class Diagram representation. Sub-sections 6.3.1 and 6.3.2 provide a detailed description of the Multimodal Ambiguities Classifier and Solver modules (on which this thesis dissertation efforts were focused).
Classifying and solving ambiguities of Multimodal Sentences require:

- Multimodal Sentences and their candidate interpretations as inputs.
- information about the context (i.e. what the Multimodal Sentence or a set of Multimodal Sentences concern and mean),
- information about the user's behaviour in the multimodal interaction process collected to define the user's profile,
- visualizing the identified ambiguity class and its solution.

The SW design at a high level of abstraction, which satisfies these requirements, is now presented. It describes the SW components of the platform that directly interact with the Multimodal Ambiguities Classifier and Solver modules, the JAVA libraries involved and their connections (Figure 6.3).

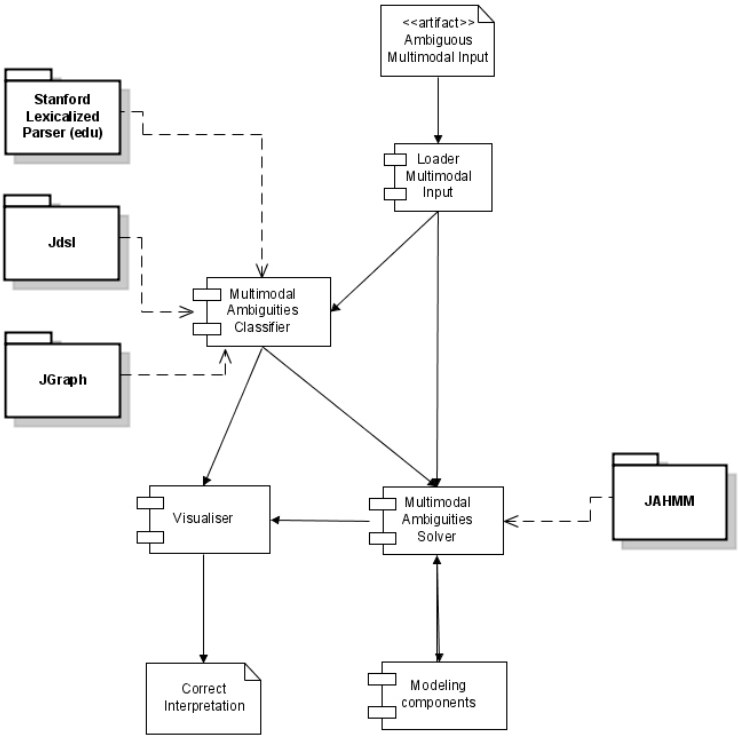


Figure 6.3: Component Diagram of the Multimodal Ambiguities Classifier and Multimodal Ambiguities Solver modules

The main components that answer to the requirements before described are:

- *Loader Multimodal Input*: it is a module devoted to load the multimodal input. As the Multimodal Ambiguity Classifier as well as the Multimodal Ambiguity Solver modules are internal modules of the M2LP platform this module acquires the Multimodal Sentences and produces an XML file that describes it according to the attributes defined for the Multimodal Language;
- *Modeling Component*: it is devoted to manage information on the user's profile and on the context with which Multimodal Sentences are related to;
- *Stanford Lexicalized Parser*: it is a natural language parser that works out the grammatical structure of sentences; it is a probabilistic parser that uses knowledge of language gained from hand-parsed sentences and it tries to produce the most likely analysis of new sentences;
- *Multimodal Ambiguities Classifier*: it is the module for the classification and the recognition of the ambiguities connected to the multimodal input (defined in this work);
- *Multimodal Ambiguities Solver*: it is the module for the resolution of the multimodal input ambiguities (defined in this work);
- *JAHMM*: it is a Java implementation of Hidden Markov Model (HMM);
- *Visualiser*: it is the module for visualizing the detected class of ambiguity, its syntax-graph, and the correct interpretation for an ambiguous Multimodal Sentence.

In particular, these modules mainly use the following java libraries:

- *jgraph*: this library is used in order to visualise the syntax-graph of the multimodal sentence;
- *jdsl:* this library is used to create and manage complex data such as list, queue, tree, graph and priority queues; it is used in order to manage all structures that implies managing graphs;
- *edu:* this is the java implementation of the Stanford Parser; it is applied for obtaining the syntactic tree connected with

the sentence in natural language that represents the candidate interpretation of the Multimodal Sentence.

Figure 6.4 is a "zoom in" on the Multimodal Ambiguities Classifier and Multimodal Ambiguities Solver modules.

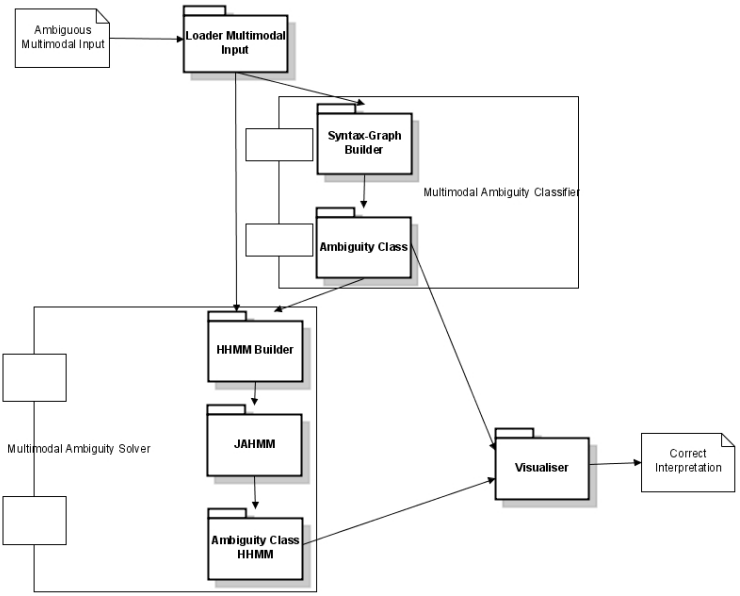


Figure 6.4: Packages that compose the Multimodal Ambiguities Classifier and Multimodal Ambiguities Solver modules

The figure shows that the ambiguous multimodal input is loaded by the *Loader Multimodal Input* and it is conveyed to the *Multimodal Ambiguities Classifier* module that manipulates this multimodal input creating the syntax-graph connected to this ambiguous input. The defined syntax-graph is analysed by the *Ambiguity Class* module that classifies the ambiguous Multimodal Sentence according to a specific class of ambiguity. The loaded Multimodal Sentence and its class of ambiguity are conveyed to the *HHMM*

Builder that uses them in order to build the HHMM instance for the identified class of ambiguity. The HHMM model needs to be trained in order to solve ambiguities. When the training is complete, the ambiguous Multimodal Sentences can be correctly (i.e. unambiguously) interpreted using the package *Ambiguity Class HHMM*.
The following two sections provide a more detailed description of the *Multimodal Ambiguities Classifier* and *Multimodal Ambiguities Solver* modules and their diagram of classes.

6.3.1 Multimodal Ambiguities Classifier

The Multimodal Ambiguity Classifier Module identifies the classes of ambiguities connected with the ambiguous multimodal input using the rules defined in the Chapter 3.
It is an internal module of the overall architecture of the M2LP; in fact, when a user inputs a Multimodal Sentence, it is interpreted by the *Multimodal Interpreter*. If the sentence is ambiguous the Multimodal Interpreter produces at least two interpretations. These interpretations (expressed by Natural Language Sentences) and the Multimodal Sentence are stored into an XML file (using XML as standard for exchanging information). This approach makes the input independent from the different input devices and from the architecture of the Multimodal System. Figure 6.5 represents an example of XML input file.

```
<?xml version="1.0"?>
<input>

  <mminput mod="speech" repr="every"  ts="2008-01-15 18:05:15"  te="2008-01-15 18:05:17" conc="every"/>
  <mminput mod="speech" repr="man"  ts="2008-01-15 18:05:18"  te="2008-01-15 18:05:20" conc="man"/>
  <mminput mod="speech" repr="saw"  ts="2008-01-15 18:05:21"  te="2008-01-15 18:05:23" conc="saw"/>
  <mminput mod="speech" repr="the"  ts="2008-01-15 18:05:24"  te="2008-01-15 18:05:25" conc="the"/>
  <mminput mod="speech" repr="boy"  ts="2008-01-15 18:05:26"  te="2008-01-15 18:05:29" conc="boy"/>
  <mminput mod="speech" repr="with"  ts="2008-01-15 18:05:30"  te="2008-01-15 18:05:31" conc="with"/>
  <mminput mod="speech" repr="this"  ts="2008-01-15 18:05:32"  te="2008-01-15 18:05:35" conc="this"/>
  <mminput mod="gesture" repr="binoculars"  ts="2008-01-15 18:05:31"  te="2008-01-15 18:05:35" conc="binoculars"/>

  <nlsentence sent="every man saw the boy with this binoculars"/>
  <nlsentence sent="every man saw, the boy with this binoculars"/>
</input>
```

Figure 6.5: Example of XML file connected with the ambiguous multimodal sentence

The Multimodal Ambiguity Classifier module takes as input XML files similar to the file of Figure 6.5 and, it asks to the Stanford Parser to parse the information contained in the XML file in order to obtain the syntactic roles it needs to build the syntax-graph.
The Multimodal Ambiguity Classifier uses the parsed information for detecting the multimodal ambiguity class and displaying the syntax-graph connected with the ambiguous Multimodal Sentence. For demonstrative purpose this module has been designed and implemented to visualize the syntax-graph even if this graph is produced to be the input of the *Multimodal Ambiguity Solver Module*.
The *Loader Multimodal Input* package reads information contained in the input XML file and sets using them the Multimodal Ambiguity Classifier parameters.
The Figure 6.6 shows the classes contained in this package. In particular the *Multi modal Input* class is used to set the attributes of each element of the Multimodal Sentence. The *Loader*, *ThreadButton* and the *Wait* classes permit to manage the information loading process from the XML file.

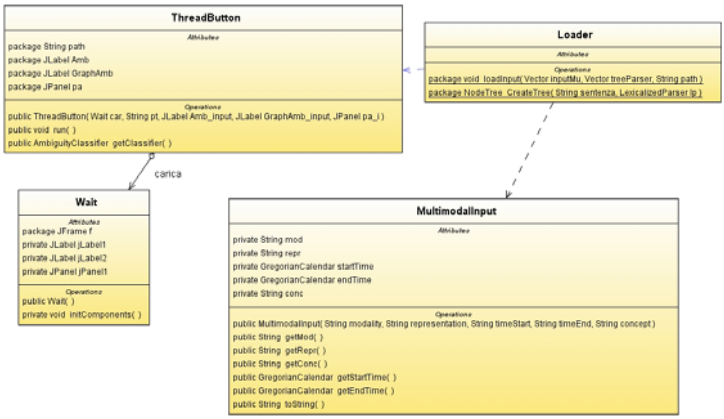


Figure 6.6: Loader Multimodal Input package

Once information is loaded, the Stanford Parser library parses the natural language sentences representing the candidate

interpretations of the ambiguous Multimodal Sentence producing a syntax-tree for each different interpretation. These syntax trees are merged into a syntax-graph (see Chapter 3), and the terminal nodes of the syntax-graph are produced with elements of the multimodal sentences.

The Figure 6.7 shows the package *Syntax-Graph Builder*. It contains the classes used to define the structure of the multimodal elements and used to build the syntax-graph combining the structures of the syntax trees given by the Stanford Parser. The classes *Nodo*, *Vertice*, *Pozzo* and *Arco* are the classes that merge respectively, the nodes and the arcs of the involved syntax-trees into vertices and arcs of the syntax graph. The SyntaxGraph class manages the building process of the syntax-graph.

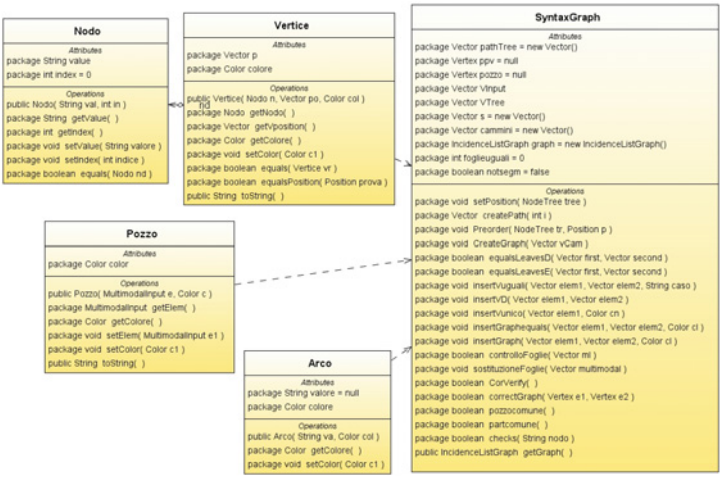


Figure 6.7: Syntax-Graph Builder package

The syntax-graph connected to the ambiguous multimodal input is analysed by the *AmbiguityClassifier* class (Figure 6.8), that is the most important class of *Ambiguity Class* package.

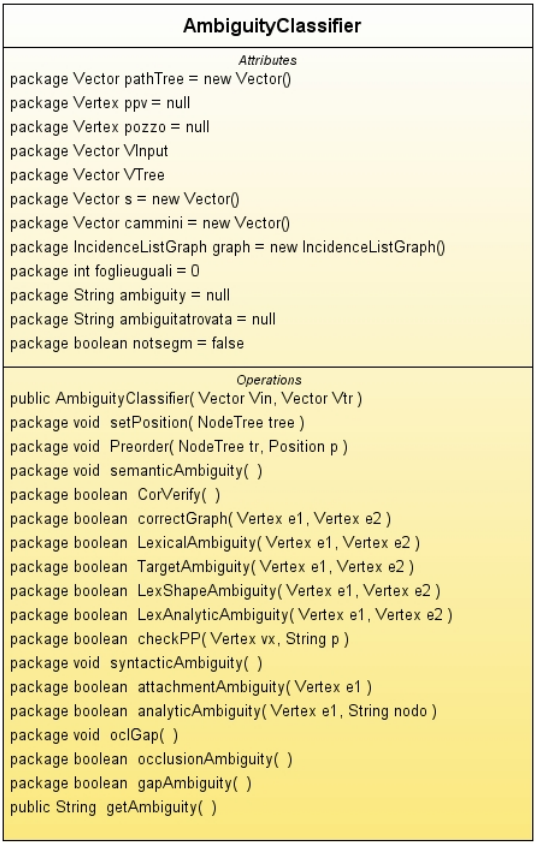


Figure 6.8: Ambiguity Class package

The *AmbiguityClassifier*, using the syntax-graph produced by the *Syntax-Graph Builder* and rules defined in Chapter 3, detects the class of ambiguity that each ambiguous Multimodal Sentence refers to.

Once the class of ambiguity has been identified, the *Ambiguities Solver* solves the ambiguous multimodal input using the knowledge about the ambiguity class.

The next section will describe the *Multimodal ambiguity solver* module, and the chapter will end with an example of use involving both software modules.

6.3.2 Multimodal Ambiguity Solver

When the Multimodal Ambiguity Classifier module detects the classes of multimodal ambiguities, this information is sent to the ***Multimodal Ambiguity Solver*** module that uses it to set the HHMM model according to the class of ambiguity.
This module manages information about ambiguous multimodal input contained in the XML file and the class of multimodal ambiguity detected by the *Multimodal Ambiguity Classifier* module. Starting from this information and the information about the user, this module computes the correct interpretation using HHMMs (see Chapter 5).
Figure 6.9 shows the classes *Loader*, *Build_Model* and *hddModel*, used to read information contained in the XML file. Moreover it receives by the *Multimodal Ambiguity Classifier* information about the class of multimodal ambiguity. It sends the syntax-graph, the ambiguity class and the information about the multimodal input to the *hddModel* that uses this information in order to manage the HHMM connected to the detected ambiguity for building the HHMM.

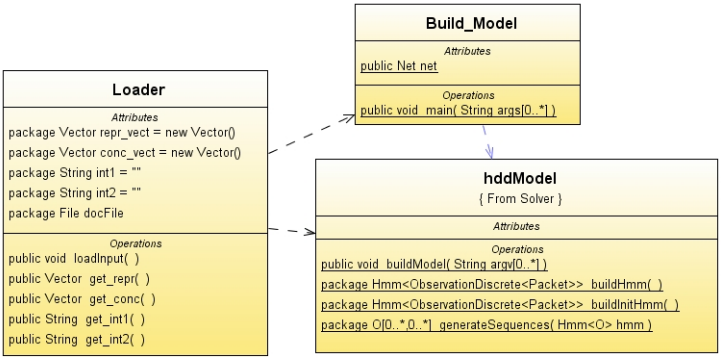


Figure 6.9: HHMM Builder package

The Multimodal Ambiguity Solver module is HHMM oriented. Its results deeply depend on the training process.
The use of HHMM needs of a training process, and usually, the more the module is trained the better is the result of the disambiguation process. During the training process of the HHMMs some ambiguous multimodal inputs are clustered according to their specific classes of ambiguities into sets. Elements, which are contained in the same set, share the same ambiguity class and they are used for training the HHMM considering the specific class of ambiguities.
Once the model is learned, the system is able to disambiguate the ambiguous multimodal input using information connected with the class of ambiguity and the information about the user profile as shown in the following figure.

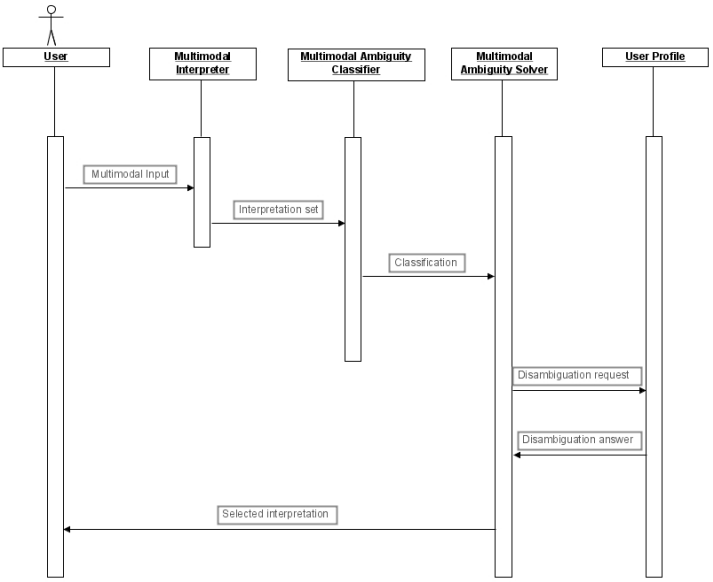


Figure 6.10: System sequence diagram for the solution of the ambiguous input

The next section presents an example of use of the Multimodal Ambiguities Classifier and the Multimodal Ambiguities Solver modules.

6.3.3 Example of use of Multimodal Ambiguity Classifier and Multimodal Ambiguities Solver modules

For the sake of clarity this section describes an example of use of the Multimodal Ambiguity Classifier and of the Multimodal Ambiguities Solver. It considers the example for lexical ambiguity. The input of these modules is an XML file representing a Multimodal Sentence transmitted by the loader multimodal input. In order to test these modules an interface has been designed, which

allows selecting ambiguous Multimodal Sentences from a significant set of defined XML files (Figure 6.11).

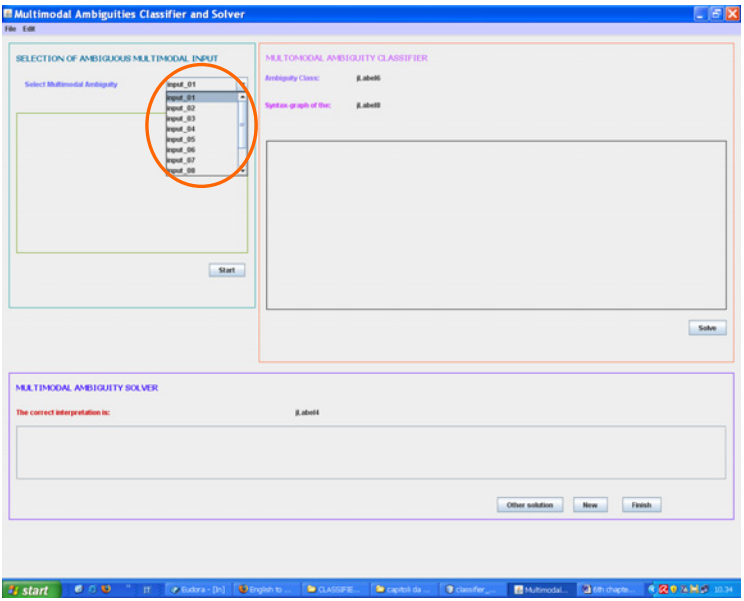


Figure 6.11: List of the ambiguous multimodal input

Once the user has selected one input (for example input_01 of Figure 6.11) the system returns the representation of the multimodal inputs as shown in Figure 6.12. This figure graphically shows the modalities defining the selected multimodal inputs and their coordination.

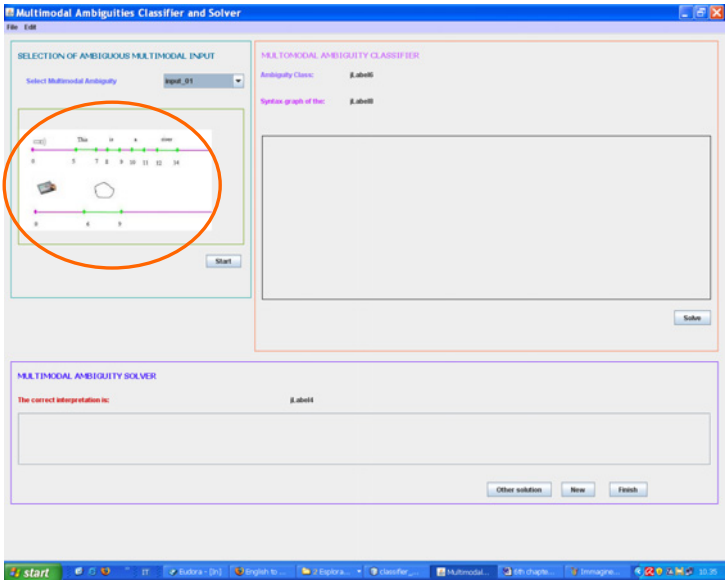


Figure 6.12: Selection of ambiguous multimodal input

When the multimodal input has been selected, the user selects the *start* command and the system starts to analyse the multimodal input.

The syntax classifier produces the syntax-graph connected with the selected ambiguous multimodal input and it returns the class of ambiguity (in this case lexical ambiguity class), as shown in Figure 6.13.

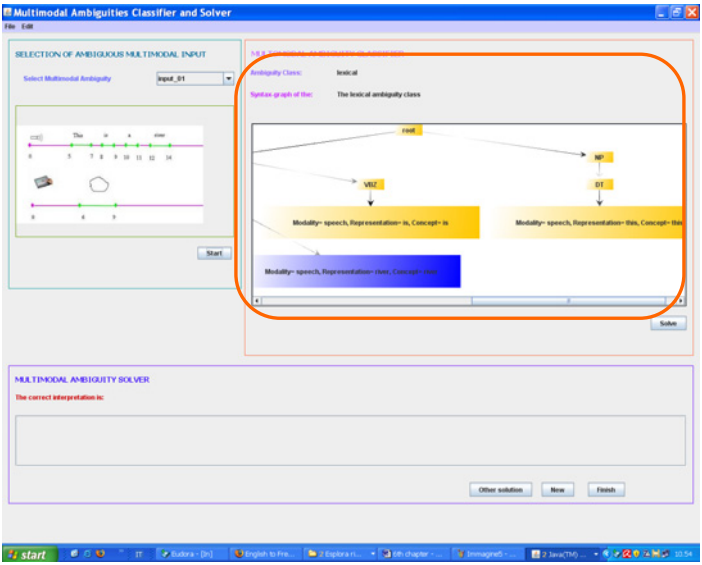


Figure 6.13: Syntax-graph and ambiguity class connected with the ambiguous input

The detected class of ambiguity (lexical in this example) and the elements that compose the ambiguous Multimodal Sentence are sent to the Multimodal Ambiguities Solver that solve it using the HHMMs of the identified class of ambiguity by choosing the best path using the Viterbi algorithm.

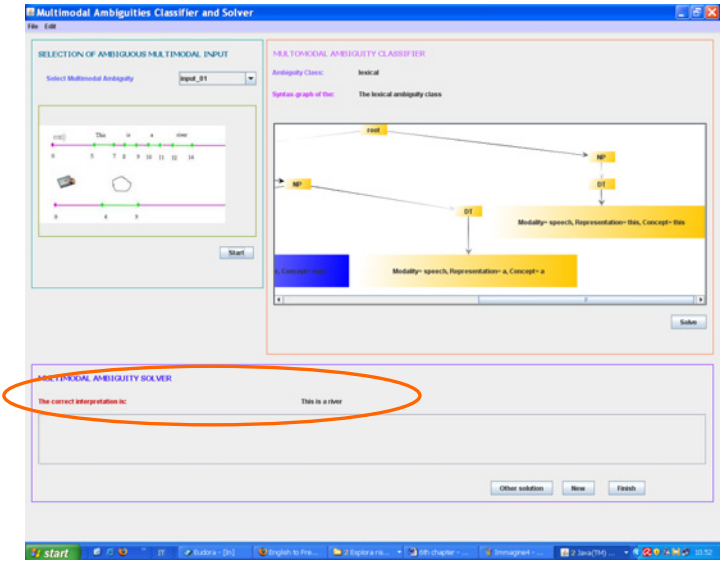


Figure 6.14: Correct interpretation of the ambiguous multimodal input

6.4 Conclusion

This chapter has described the design of the Multimodal Ambiguities Classifier module and of the Multimodal Ambiguities Solver module produced by the work carried out for this thesis. Their evaluation process and its results are described in Chapter 7.

Chapter 7

Evaluation and Discussion

7.1 Introduction

This chapter presents the evaluation process of the Multimodal Ambiguities Classifier and of the Multimodal Ambiguities Solver. It aims at validating methods and models proposed in this thesis dissertation to face problems of classifying and solving multimodal ambiguities.
The evaluation process has required to identify the scopes of the evaluation, to identify metrics and measures involved and to define the test set used in the experiment carried out to implement the evaluation.
The evaluation process has allowed:

1. to validate the classification proposed for ambiguities of Multimodal Sentences and the set of rules defined to detect them in terms of accuracy of the multimodal ambiguities classification;

2. to validate the model defined in order to solve ambiguities based on the use of the Hierarchical Hidden Markov Models in terms of accuracy of the ambiguities resolution process.

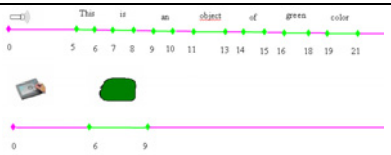
Section 7.2 describes criteria adopted to identify the test set consisting in the ambiguous Multimodal Sentences. The test set is

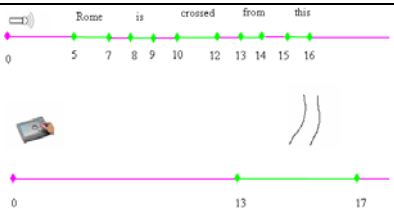
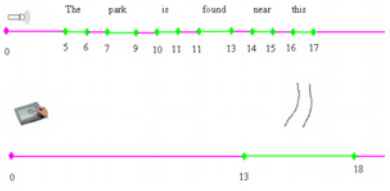
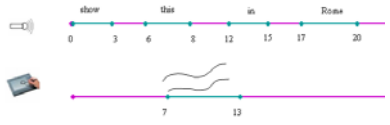
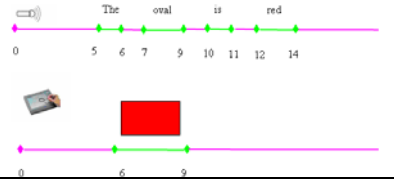
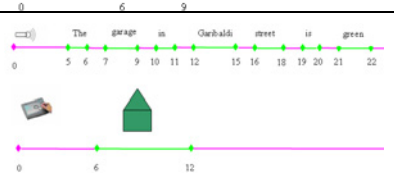
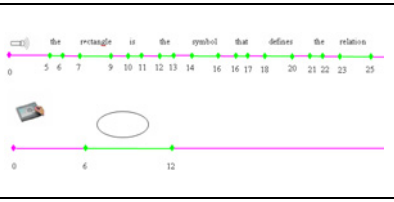
used for both the Multimodal Ambiguities Classifier and the Multimodal Ambiguities Solver module.

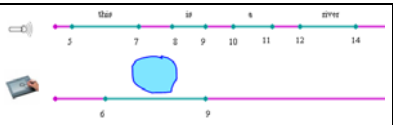
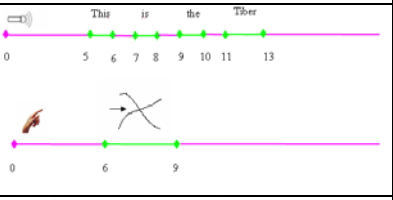
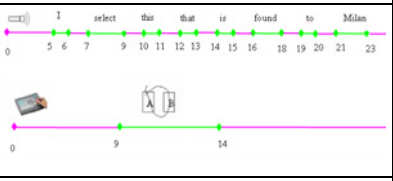
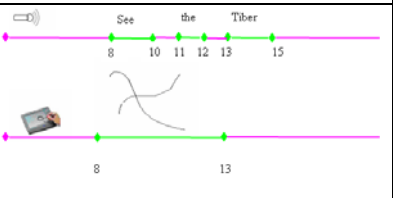
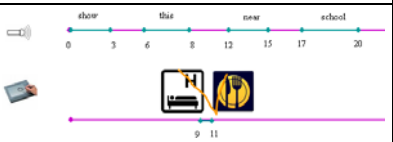
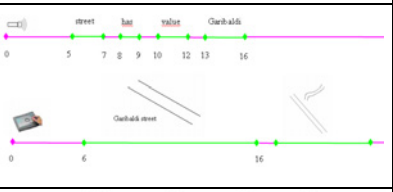
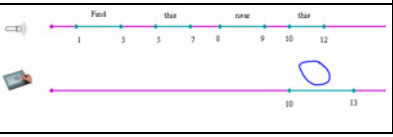
7.2 Definition of the test set

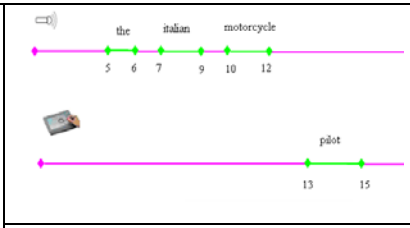
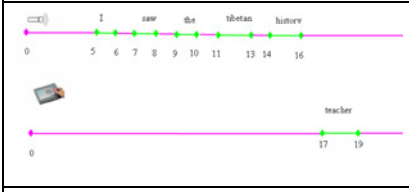
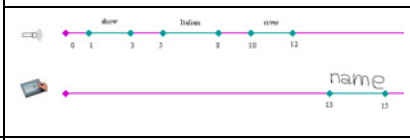
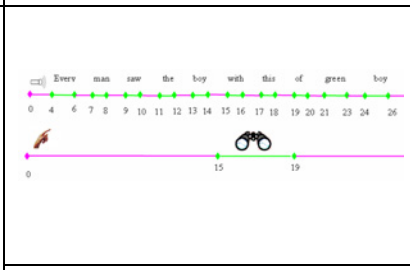
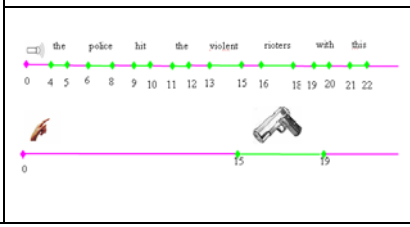
This section describes criteria adopted to identify the test set used for evaluating the Multimodal Ambiguities Classifier and the Multimodal Ambiguities Solver.
A set of 60 ambiguous Multimodal Sentences has been defined. Thirty (30) of them contain a semantic multimodal ambiguity (10 lexical, 10 temporal-semantic and 10 target ambiguities); the remaining 30 contain syntactic multimodal ambiguities (10 gap, 10 analytic and 10 attachment ambiguities).
The set of ambiguous multimodal inputs, presented when describing the classes of multimodal ambiguities in Chapter 3, has been enriched by other ambiguous multimodal inputs for each class of multimodal ambiguities.
The following table presents some of the ambiguous multimodal inputs used to test the Multimodal Ambiguities Classifier and the Multimodal Ambiguities Solver modules.
In the Table 7.1 ambiguous multimodal inputs are clustered according to their classes of ambiguities, and a set of their possible interpretations are presented.

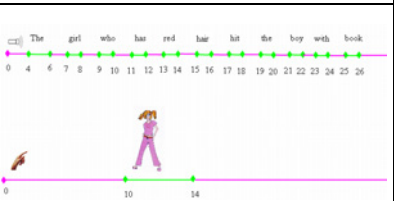
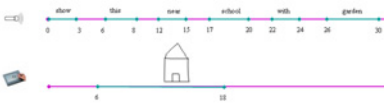
Table 7.1: Examples of inputs for the testing process

Ambiguity Class		Multimodal input	**Possible Interpretations in NL**
Semantic ambiguity	Lexical		▪ This rectangle is an object of green colour ▪ This oval is an object of green colour

			▪ Rome is crossed from this river ▪ Rome is crossed from this road
			▪ The park is found near this river ▪ The park is found near this road
			▪ show this river in Rome ▪ show this street in Rome
	Temporal-Semantic		▪ The oval is red ▪ The rectangle is red
			▪ The garage in Garibaldi street is green ▪ The home in Garibaldi street is green
			▪ The rectangle is the symbol that defines the relation ▪ The oval is the symbol that defines the relation

			▪ This is a river ▪ This is a lake
	Target		▪ This river1 is the Tibe ▪ This river2 is the Tiber
			▪ I select this hotel-A that is found to Milan ▪ I select this hotel-B that is found to Milan
			▪ See the Tiber river1 ▪ See the Tiber river2
			▪ Show this hotel near school ▪ Show this restaurant near school
Syntactic Ambiguity	Gap		▪ street that has value Garibaldi arrives close to the river (…)
			▪ Find this (…) near this lake

	Analytic		▪ The italian motorcycle pilot ▪ The italian motorcycle, pilot
			▪ I saw the Tibetan teacher of history ▪ I saw the teacher of Tibetan history
			▪ Show the Italian name of the river ▪ Show the name of the Italian river
	Attachment		▪ Every man saw the boy with this binoculars of green colour ▪ Every man saw, the boy with this binoculars of green colour
			▪ The police hit the violent rioters with this gun ▪ The police hit, the violent rioters with this gun

			▪ the girl who has red hats hit the boy with this book ▪ the girl who has red hats hit, the boy with this book
			▪ the girl who has red hair hit the boy with this book ▪ the girl who has blond hair hit the boy with this book
			▪ Show this house near school with garden ▪ Show this house near, school with garden

This set of multimodal sentences has been used to evaluate the performances of the two modules presented in the section and in section 7.4.

7.3 Multimodal Ambiguities Classifier evaluation

The evaluation process of the Multimodal Ambiguities Classifier consists of evaluating its performances when, having an ambiguous multimodal sentence of the set test as input, it has to correctly assign the multimodal sentence to the ambiguity class.
Let be given the set test described in section 7.2. The Multimodal Ambiguity Classifier will receive in input the XML files of the Multimodal Sentences defined in the test set, but it doesn't know the class of ambiguity of each sentence of the test set (which represents the expected class of ambiguity).

Starting from this consideration it is possible to calculate the performance in terms of accuracy of the classification process of multimodal ambiguities considering, for each Multimodal Sentence, how the expected class of ambiguity matches the detected class of ambiguity.
The test has been performed on the test set defined in section 7.2.1. Given this test set the Multimodal Ambiguities Classifier has been able to achieve classification accuracy in the 90% for the classes of semantic ambiguities, and 96,7% for the classes of syntactic ambiguities. In detail, considering 30 semantic multimodal ambiguities (10 lexical, 10 temporal-semantic and 10 target ambiguities), and 30 syntactic multimodal ambiguities (10 gap, 10 analytic and 10 attachment ambiguities), the module has correctly classified 27 examples of semantic multimodal ambiguities, and 29 examples of syntactic multimodal ambiguities.
The next section describes the evaluation for the Multimodal Ambiguities Solver.

7.4 Multimodal Ambiguities Solver evaluation

The evaluation process of the Multimodal Ambiguities Solver consists in the analysis of its performances to solve the different classes of multimodal ambiguities in terms of accuracy of interpretation obtained by the Multimodal Ambiguities Solver and the expected one.
The purpose is to analyse how much the interpretation identified as correct by the Multimodal Ambiguities Solver moves away from the expected interpretation.
The evaluation process of the Multimodal Ambiguities Solver has been structured in two phases:

- the training phase of the Multimodal Ambiguities Solver; and
- the test phase of the Multimodal Ambiguities Solver.

The training phase has involved: 6 people, 3 men and 3 women, from 25 to 65 years old. The used set of ambiguous multimodal sentences is the test set.
During the training each participant has had to input the Multimodal Sentences of the set test (by the input SW module).
Each person has input each Multimodal sentence two times:

- a first time, she/he knows the Multimodal Sentence she/he has to insert and the first interpretation according to the set test (except for the gap ambiguity);
- a second time, she/he has had to input each Multimodal Sentence knowing its second interpretation according to the set test (except for the gap ambiguity).

This training process allows capturing the key parameters (such as class of cooperation among modalities and temporal relations among elements of the Multimodal Sentence) according to the correct interpretation to assign to the multimodal sentence.
The trained Multimodal Ambiguities Solver has been tested on the test set.

In the test phase, the evaluation of the performances of the Multimodal Ambiguity solver has used two measures: the accuracy of the ambiguities resolution process.
The accuracy of the ambiguities resolution process measures how the expected interpretation of a Multimodal Sentence matches the interpretation of a Multimodal Sentence produced by the software module.
For the syntactic ambiguities, when the sequence of hidden states, obtained by the HHMM, is not the same of the expected interpretation of a Multimodal Sentence than the error rate evaluates the distance between the two sequences; if this distance is lower than a predefined threshold than the obtained sequence of hidden states is considered correct.
The error rate is defined as follows:

$$Error(S', S) = \frac{Diff(S', S)}{L} \qquad (7.1)$$

Where *S', S* and *L* are:

- *S'* that is the hidden states sequence, generated by the HHMM of the Multimodal Ambiguities Solver module, that corresponds to the emission sequence, which coincides with one example presented in Table 7.1,
- *S* that is the correct hidden states sequence (the actual hidden states sequence defined during the definition of the training examples) that corresponds to the emission sequence, which coincides with the same example used for *S'*,
- *L* that is the length of the interpretation sequence that corresponds to the number of elements that are obtained as output of the HHMM of the Multimodal Ambiguities Solver module.

It is a measure that computes the differences between two states sequences (*Diff(S', S))* as the number of corresponding hidden states that do not agree with each other, and S' and S are sequences of the same length L.
This parameter is calculated for each Multimodal Sentence of the set test connected with syntactic ambiguities and the smaller is the error rate and the better is the solution. For this reason, during the experiment has been defined a threshold for the syntactic ambiguities. A sequence of hidden states is defined correct if its error rate is lower than the defined threshold. In the experiments this threshold has been established as $\frac{1}{L}$.
For lexical ambiguity, when the sequence of hidden states, obtained by the HHMM, is not the same of the expected interpretation, then the accuracy has been evaluated using the Lin semantic similarity measure [Lin98], which is defined as the maximum information content shared by the two concepts divided by the information content of the compared concepts. Let be given two concepts c_1 and c_2; considering WordNet as lexical taxonomy and the least upper bound (*lub*) of the two concepts c_1 and c_2, the semantic similarity is:

$$SiSem(c_1, c_2) = 2\frac{\log p(\text{lub}(c_1, c_2))}{\log p(c_1) + \log p(c_2)} \qquad (7.2)$$

This value is calculated for each Multimodal Sentence of the set test connected with lexical ambiguities referred to the WordNet lexical taxonomy [Wor2.1]. In this case a sequence of hidden states is defined correct if its Lin measure is upper than 0,8.

For the remaining sub-classes of semantic ambiguities a threshold has not been considered, and only when the sequence of hidden states obtained by the HHMM matches the expected interpretation then the sequence is considered correct.
The Multimodal Ambiguities Solver module has been tested on the set test that, as defined in section 7.2, consists of 30 samples of semantic multimodal ambiguities (10 lexical, 10 temporal-semantic and 10 target ambiguities), and 30 samples of syntactic multimodal ambiguities (10 gap, 10 analytic and 10 attachment ambiguities).
It has achieved a resolution accuracy of 80% on the classes of semantic ambiguities, and of the 93,3% on the classes of syntactic ambiguities.
The Multimodal Ambiguities Solver module is capable of learning incrementally because it can improve its performances gradually as it is learned with more and more examples. The module has correctly interpreted 24 examples of semantic multimodal ambiguities, and 27 examples of syntactic multimodal ambiguities.

7.5 Discussion

The evaluation of the Multimodal Ambiguities Classifier and the Multimodal Ambiguities Solver modules has served to provide the accuracy of the multimodal ambiguities classification, and the error rate of the resolution process.
In particular, the Multimodal Ambiguities Classifier has achieved an accuracy of 90% for the classes of semantic ambiguities, and 96,7% for the classes of syntactic ambiguities. These values will be

improved extending the rules for classifying multimodal ambiguities.

Considering the Multimodal Ambiguities Solver, the evaluation process has underlined a resolution accuracy of 80% on the classes of semantic ambiguities, and of the 93,3% on the classes of syntactic ambiguities. These accuracy values can be improved training the HHMM with a wider set of examples. Moreover, the training process will be refined involving more people. Future directions will be discussed in the Chapter 8.

Chapter 8

Conclusions

8.1 Summary

This dissertation has faced the problem of ambiguity, which usually arises when a user interacts with a multimodal system.
The communication process between user and system has been dealt by the linguistic point of view focusing the problem on the interpretation of the language sentences.
An overview of the relevant studies on ambiguities related with the Natural Language and Visual Languages has been carried out and, the most relevant results of these studies have been extended and generalised for Multimodal Languages. A general framework for classifying multimodal ambiguities has been defined.
Once an ambiguity has been identified and classified it needs to be solved. For this purpose the thesis has analysed the most frequently used methods proposed in literature for solving ambiguities. The analysis of these methods has underlined the relevance of approximation methods for dealing with uncertainty, which is characteristic in natural interaction approaches. For this reason a resolution methods based on Hierarchical Hidden Markov Models (HHMM) has been proposed and adopted in this dissertation to solve semantic and syntactic multimodal ambiguities.
The results obtained for classifying and solving multimodal ambiguities have been implemented and validated in the design and developed of two software modules: the Multimodal Ambiguities Classifier and the Multimodal Ambiguities Solver.

The evaluation of these two modules has provided a good level of accuracy of classification and solution of multimodal ambiguities.

8.2 Contribution

The work in this dissertation was motivated by two main challenges in the treatment of the interpretation problems connected to the *Multimodal Language*:

- the analysis and the classification of ambiguities connected with the *Multimodal Language*;
- the definition of a method devoted to solve ambiguities connected with the *Multimodal Language*.

The main results of the thesis are represented by:

- a general classification framework of ambiguities for *Multimodal Languages* that represents an extension of the Natural Language and Visual Languages ambiguities;
- a method for coping the problem to define the un-ambiguous interpretation of a *Multimodal Sentence* based on the use of HHMMs.
- the design and implementation of two software modules for classifying and solving ambiguities of Multimodal Sentences.

The foundation of all the contributions in this dissertation is the definition of a method for dealing ambiguities connected with the *Multimodal Language* based on the integration of multiple information types.
The thesis has given a classification of multimodal ambiguities and the set of rules to identify them, using the notions of *Multimodal Grammar*, *terminal element* of the Multimodal Grammar, *Multimodal Sentence* and *Multimodal Language*.
The provided classification of ambiguities extends classifications defined by studies on the Natural Language and Visual Languages. In particular, ambiguities can be distinguished in *Semantic* and *Syntactic*. The *Semantic ambiguities* have been classified into:

lexical ambiguity; temporal-semantic ambiguity; and target ambiguity. *Syntactic ambiguities* have been divided into: gap ambiguity; analytic ambiguity; and attachment ambiguity.
A method to face the problem to define the correct interpretation of a *Multimodal Sentence* has been defined; it incorporates semantic and syntactic information connected with the *Multimodal Sentence* and is based on HHMMs. The proposed method contains three connected models of a Multimodal Sentence: the *context model*, the *semantic model* and the *syntactic sentence model*.
The *context model* has been used to associate each terminal element of the *Multimodal Language* with a semantic tag representing the meaning of the element. The *semantic model* has been referred to concepts connected with terminal elements of the *Multimodal Sentence*, and the *syntactic sentence model* has represented each *Multimodal Sentence* as a sequence of syntactic roles.
This method allows modelling a multi-levels stochastic process using HHMMs. It provides a multilevel description of a Multimodal Sentence from the terminal elements interpretation to the Multimodal Sentence interpretation and its context, connecting the different Hidden Markov models.

8.3 Future Research

The original contribution of this thesis dissertation consists in facing the problem of ambiguities, extending and generalising classifications and solutions methods arising from the literature to the multimodal ambiguities.
As results of the evaluation of this work (Chapter 8) suggest, this study, methods for classifying and solving multimodal ambiguities need to be tested, and consequently evolved in a wide corpus of Multimodal Sentences.
Moreover the solution process will be improved using a wider set of examples for training the HHMM. Involving more people in the training process can be useful in order to define a user's model, acquiring knowledge on the users' behaviour.

Bibliography

[AAA94] Agirre, E., Arregi, X., Artola, X.,. Díaz de Ilarraza X.,A, Evrard, F. and Sarasola K. (1994) Lexical Knowledge Representation in an Intelligent Dictionary Help System *In International Conference On Computational Linguistics.* pp 544-550.

[AlC92] Alshawi, H. and Crouch, R.. (1992). Monotonic semantic interpretation. *In The Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics.*

[AlF94] Allen, J. F. and Ferguson, G. (1994), `Actions and events in interval temporal logic', *Journal of Logic and Computation 4(5)*, pp.531-579.

[BaH93] Baber, C., Hone, K. S. (1993). Modelling error recovery and repair in automatic speech recognition. *International Journal of Man-Machine Studies 39, 3*. (pp. 495–515).

[BaT66] Baum L. E. and Petrie T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Annals of Math. Statistics, vol. 37*, pp.1554-1563.

[BCL95] Bottoni, P., Costabile, M. F., Levialdi, S., Mussio, P. (1995). 'Formalizing Visual Languages'. *VL 1995*. pp. 45-52.

[BCM99] Bottoni, P., Costabile, M. F., Mussio, P. (1999): Specification and dialogue control of visual interaction through visual rewriting systems. *ACM Trans. Program. Lang. Syst. 21(6)*: pp.1077-1136.

[Bel01] Bellik Y. (2001). Technical Requirements for a Successful Multimodal Interaction. *International Workshop on Information Presentation and Natural Multimodal Dialogue*, Verona, Italy. 14-15 December 2001, 5 pages.

[Bez92] Bezdek, J. C. (1992). Computing with Uncertainty, *IEEE Communications Magazine, 30(9).*pp. 24-36.

[BKK01] Berry, D.M., Kamsties, E., Kay, D.G., and Krieger, M.M., (2001). From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity, *Technical Report, University of Waterloo, Waterloo, ON, Canada*.

[Buv96] Buvac, S. (1996). Resolving lexical ambiguity using a formal theory of context. *In Van Deemter, K., Peters, S., eds.: Semantic Ambiguity and Underspecification, CSLI Publications.*

[CaM94] Calcinelli, D., Mainguenaud, M. (1994). Cigales, a visual language for geographic information system: the user interface. *Journal of Visual Languages and Computing 5(2).* pp. 113-132.

[Car92] Carpenter, B. (1992). The Logic of Typed Feature Structures. *Cambridge University Press*.

[CFG07] Caschera M.C., Ferri F., Grifoni P. (2007). An Approach for Managing Ambiguities in Multimodal

Interaction. *OTM 2007 Ws, Part I, Springer-Verlag LNCS 4805*. pp. 387–397.

[CFG07b] Caschera M.C., Ferri F., Grifoni P., (2007). Multimodal interaction systems: information and time features. *International Journal of Web and Grid Services IJWGS - Vol. 3, No.1* pp. 82 – 99.

[CHZ04] Chai, J. Y., Hong, P., & Zhou, M. X. (2004). A probabilistic approach to reference resolution in multimodal user interfaces. *In Proceedings of 9th International Conference on Intelligent User Interfaces (IUI)*, pp. 70–77.

[CJM97] Cohen, P. R. Johnston, M., McGee, D., Smith, I., Oviatt, S., Pittman, J., Chen, L. and Clow, J. (1997) "Quickset: Multimodal interaction for simulation set-up and control" In *proceedings of the Fifth Applied Natural Language Processing meeting*.

[ClC79] Clark, E. V., & Clark, H. H. (1979). When nouns surface as verbs. *Language, 55*, 767–811.

[Col97] Collins, M.. (1997). Three generative, lexicalised models for statistical parsing. *In Proceedings of the 35th Meeting of the Association for Computational Linguistics and the 7th Conference of the European Chapter of the ACL.* pp. 16-23.

[CPQ06] Chai, J.Y., Prasov, Z., Qu, S. (2006). Cognitive Principles in Robust Multimodal Interpretation. *J. Artif. Intell. Res. (JAIR) 27:* pp.55-83.

[DeM05] Dey, A. K. Mankoff , J., 2005. Designing mediation for context-aware applications. *ACM Trans. Comput.-Hum. Interact. 12(1)*, pp. 53-80.

[Ege97] Egenhofer, M. J. (1997). Query Processing in Spatial-Query- by-Sketch. *Journal of Visual Languages and Computing 8(4)*. pp. 403-424.

[EKR98] Eickeler, S. Kosmala, A. and Rigoll, G. (1998). "Hidden Markov Model based continuous online gesture recognition". *Proceedings of international conference on pattern recognition, 2,* pp.1206-1208.

[FaA00] Favetta, F. Aufaure-Portier, M.A. (2000). About Ambiguities in Visual GIS Query Languages: a Taxonomy and Solutions. *Proceedings of the 4th International Conference on Advances in Visual Information Systems, Springer-Verlag*, pp. 154-165.

[FeR05] Ferri, F., Rafanelli, M. (2005). GeoPQL: A Geographical Pictorial Query Language That Resolves Ambiguities in Query Interpretation. *J. Data Semantics III*. pp.50-80.

[FoJ00] Fonseca, M. J., Jorge J. A. (2000). CALI : A Software Library for Calligraphic Interfaces. *INESC-ID, http://immi.inesc-id.pt/cali/*.

[Fos98] Fosler-Lussie, E. (1998). Markov Models and Hidden Markov Models - A Brief Tutorial. *International Computer Science Institute Technical Report TR-98-041*.

[FST98] Fine S., Singer Y., Tishby N., (1998). The Hierarchical Hidden Markov Model: Analysis and Applications, *Machine Learning, vol. 32*, p. 41-62.

[Fut99] Futrelle, R. P. (1999). Ambiguity in Visual Language Theory and its Role in Diagram Parsing. *IEEE Symposium on Visual Languages, Tokyo, IEEE Computer Soc.*, pp. 172-175.

[GaW98] Gavalda, M. and Waibel, A.. (1998). Growing Semantic Grammars. *Proceedings of ACL/ Coling 1998, Montreal, Canada*.

[Gir87] Girare, J.-Y. (1987). Linear logic. Theoretical Computer Science, 50:1-102.

[Har94] Harper, M. P. (1994). Storing Logical Form in a Shared-Packed Forest. *Computational Linguistics 20(4)*. pp. 649-660.

[HaS04] Harper, M. P., Shriberg,E. (2004). Multimodal model integration for sentence unit detection. *ICMI 2004*. pp.121-128.

[HBT94] Hu, J., Brown, M.K., and Turin, W., (1994). "Handwriting recognition with hidden Markov models and grammatical constraints", *Fourth International Workshop on Frontiers of Handwriting Recognition*, Taipei, Taiwan.

[HCB95] Huls, C., Claassen, W.and Bos, E. (1995). Automatic referent resolution of deictic and anaphoric expressions. *Computational Linguistics*, 21(1): pp.59–79.

[Hir87] Hirst,G. (1987). Semantic Interpretation and the Resolution of Ambiguity. *Cambridge University Press, Cambridge, England.*

[JBV02] Johnston, M., Bangalore, S.,, Vasireddy, G., Stent, A., Ehlen, P., Walker, M. A., Whittaker, S., Maloor, P., MATCH: An Architecture for Multimodal Dialogue Systems. *ACL 2002*: pp.376-383.

[JiS05] Jiang, W., Sun, Z.. (2005). Hmm-Based On-Line Multi-Stroke Sketch Recognition. *Proceedings of the Fourth International Conference on Machine*

Learning and Cybernetics, Guangzhou, 18-21 August 2005. pp.4564-4570.

[JoB05] Johnston, M. and Bangalore, S. (2005). "Combining Stochastic and Grammar-based Language Processing with Finite-state Edit Machines". *In Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop.*

[Knu68] Knuth. D. E., (1968). Semantics of context-free languages, *Mathematical Systems Theory 2*, pp.127–145.

[LDR04] Landragin, F., Denis, A., Ricci, A. & Romary, L. (2004), Multimodal meaning representation for generic dialogue systems architecture, *in `Proceedings of the International Language Resources Conference (LREC04)', Lisbon, Portugal*.

[LeC95] Lee, Y.C., Chin, F. (1995). An Iconic Query Language for Topological Relationship in GIS. *International Journal of geographical Information Systems 9(1)*. pp. 25-46.

[Lin98] Lin, D. (1998). An Information-Theoretic Definition of Similarity. *Proceedings of the 15th Intern. Conference on Machine Learning (ICML'98), Madison, WI*, pp.296-304.

[MaC99] MacKenzie, I.S., Chang, L. (1999). A performance comparison of two handwriting recognisers. *Interacting with Computers 11*. pp. 283-297.

[Mar97] Martin J.C. (1997) 'Toward Intelligent Cooperation Between Modalities: The Example of a System Enabling Multimodal Interaction with a Map'. *Proceedings of International Joint Conference on Artificial Intelligence(IJCAI'97) Workshop on "Intelligent Multimodal Systems."* Nagoya, Japan.

[MCO98] McGee, D.R, Cohen P.R. and Oviatt S.L. (1998) "Confirmation in multimodal systems" In *Proc. of COLING-ACL '98*, Montreal, Canada.

[Mey93] Meyer, B. (1993). Beyond Icons: Towards New Metaphors for Visual Query Languages for Spatial Information Systems, *Proceedings of the International Workshop on Interfaces to Database Systems, Glasgow, Scotland.* pp.113-135.

[MHA00] Mankoff, J., Hudson, S.E., Abowd, G.D. (2000). Providing integrated toolkit-level support for ambiguity in recognition-based interfaces. *Proceedings of ACM CHI'00 Conference on Human Factors in Computing Systems*. pp. 368 – 375.

[MMW98] Marriott, B. Meyer, and K. Wittenburg, (1998). A survey of visual language specification and recognition. *In K. Marriott and B. Meyer, editors, Visual Language Theory, Springer, New York.* pages 5–85.

[MSM94] Marcus, M. P., Santorini, B. and Marcinkiewicz, M. A. (1994). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics, 19(2)*, pp 313-330.

[NiC93] Nigay L. and Coutaz J. (1993). 'A design space for multimodal systems - concurrent processing and data fusion'. *INTERCHI '93 - Conference on Human Factors in Computing Systems, Addison Wesley*. pp. 172-178.

[NiC95] Nigay L. and Coutaz J. (1995). 'A generic platform for addressing the multimodal challenge'. *International Conference on Computer-Human Interaction*, ACM Press. pp. 98-105.

[OCW00] Oviatt, S.L., Cohen, P., Wu, L., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. and Ferro, D. (2000) "Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions" In *Human-Computer Interaction* pp.263-322.

[OvC00] Oviatt, S.L. Cohen, P. R.. (2000). Perceptual user interfaces: multimodal interfaces that process what comes naturally**.** *In Commun of the ACM* pp. 45-53.

[Ovi03] Oviatt, S.L. (2003). 'Multimodal interfaces'. *In The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, J. JACKO AND A. SEARS, Eds. Lawrence Erlbaum Assoc., Mahwah, NJ, 2003, chap.14*. pp. 286-304.

[OWB00] O'Hara, Thomas, Wiebe, Janyce, & Bruce, Rebecca F. (2000). Selecting decomposable models for word-sense disambiguation. *Computers and the Humanities 34* (1-2).

[PWC95] Poon, A., Weber, K., Cass, T. (1995). Scribbler: A tool for searching digital ink. *In Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, volume 2 of Short Papers: Pens and Touchpads.* pp. 252-253.

[Rab89] Rabiner, L. R., (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedins of the IEEE, vol 77*. pp. 257-285.

[RSH05] Russ, G., Sallans B., and Hareter H. (2005). Semantic Based Information Fusion in a Multimodal Interface"In *CSREA HCI* . pp: 94-102.

[SeD05] Sezgin, T.M., and Davis, R., (2005), "HMM-Based Efficient Sketch Recognition," *Proc. 10th Int'l Conf. Intelligent User Interfaces (IUI 05), ACM Press*, pp. 281–283.

[SKG00] Spilker, J., Klarner, M., Görz, G. (2000). Processing Self Corrections in a speech to speech system. *COLING 2000*. pp. 1116-1120.

[TsF79] Tsai, W.H. and Fu, K.S. (1979), Error-correcting isomorphism of attributed relational graphs for pattern analysis. *IEEE Trans. Sys., Man and Cyb., vol. 9,* pp. 757−768.

[VaA97] Van den Anker, F.W.G. and Amold A.G. (1997). Mobile multimedia communication: a task- and user-centered approach to future systems development. *7th International Conference on Human Computer Interaction - HCI International '97 Conference*. pp.651-654.

[Vit67] Viterbi, A., (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Transactions on Information Theory*. pp. 260-269.

[VoM98] Vo M.T. (1998). 'A Framework and Toolkit for the Construction of Multimodal Learning Interfaces'. *PhD thesis, Carnegie Mellon University*.

[VoW96] Vo M.T. and Wood C. (1996). 'Building an application framework for speech and pen input integration in multimodal learning interfaces'. *International Conference on Acoustics, Speech and*

Signal Processing, IEEE Computer Society. pp. 3545-3548.

[VoW97] Vo M.T. and Waibel A. (1997). 'Modeling and interpreting multimodal inputs: A semantic integration Approach'. *Technical Report CMU-CS-97-192, Carnegie Mellon University*.

[W3C03] W3C (2003) NOTE 8 January 2003, "Multimodal Interaction Requirements", http://www.w3.org/TR/2003/NOTE-mmi-reqs-20030108/.

[WeH98] Wessel, M., Haarslev, V. (1998). VISCO: Bringing visual spatial querying to reality. *IEEE Symposium on Visual Languages*. pp. 170-177.

[WOC02] Wu, L., Oviatt, S. L. and Cohen, P.R. (2002). From members to teams to Committee: A robust approach to gestural and multimodal recognition. In *IEEE Transactions on Neural Networks.* pp. 972 -982.

[WOC99] Wu, L., Oviatt S. L. and Cohen, P. (1999) "Multimodal integration: A statistical view" *In IEEE Transact. Multimedia* pp: 334-342.

[Wor2.1] WORDNET 2.1: A lexical database for the English language; http://www.cogsci.princeton.edu/cgi-bin/webwn, 2005.

[Zad65] Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control 8(3)*. pp. 338-353.