POLITECNICO DI TORINO AND UNIVERSITY OF POTSDAM

SCUOLA DI DOTTORATO

Dottorato in Automatica e Informatica – XXVI ciclo

Tesi di Dottorato

People Centered HMI's for Deaf and Functionally Illiterate Users

Nadeem Ahmad

Principal Supervisor
Prof. Dr. Paolo Ernesto Prinetto

Coordinatore del corso di dottorato
Prof. Dr. Pietro Laface

Co-Supervisor
Prof. Dr. Tiziana Margaria-Steffen

March 2014

Published online at the
Institutional Repository of the University of Potsdam:
URL http://opus.kobv.de/ubp/volltexte/2014/7039/
URN urn:nbn:de:kobv:517-opus-70391
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus-70391

Abstract

The objective and motivation behind this research is to provide applications with easy-to-use interfaces to communities of deaf and functionally illiterate users, which enables them to work without any human assistance. Although recent years have witnessed technological advancements, the availability of technology does not ensure accessibility to information and communication technologies (ICT). Extensive use of text from menus to document contents means that deaf or functionally illiterate can not access services implemented on most computer software. Consequently, most existing computer applications pose an accessibility barrier to those who are unable to read fluently. Online technologies intended for such groups should be developed in continuous partnership with primary users and include a thorough investigation into their limitations, requirements and usability barriers.

In this research, I investigated existing tools in voice, web and other multimedia technologies to identify learning gaps and explored ways to enhance the information literacy for deaf and functionally illiterate users. I worked on the development of user-centered interfaces to increase the capabilities of deaf and low literacy users by enhancing lexical resources and by evaluating several multimedia interfaces for them. The interface of the platform-independent Italian Sign Language (LIS) Dictionary has been developed to enhance the lexical resources for deaf users. The Sign Language Dictionary accepts Italian lemmas as input and provides their representation in the Italian Sign Language as output. The Sign Language dictionary has 3082 signs as set of Avatar animations in which each sign is linked to a corresponding Italian lemma. I integrated the LIS lexical resources with MultiWordNet (MWN) database to form the first LIS MultiWordNet(LMWN). LMWN contains information about lexical relations between words, semantic relations between lexical concepts (synsets), correspondences between Italian and sign language lexical concepts and semantic fields (domains). The approach enhances the deaf users' understanding of written Italian language and shows that a relatively small set of lexicon can cover a significant portion of MWN. Integration of LIS signs with MWN made it useful tool for computational linguistics and natural language processing.

The rule-based translation process from written Italian text to LIS has been transformed into service-oriented system. The translation process is composed of various modules including parser, semantic interpreter, generator, and spatial allocation planner. This translation procedure has been implemented in the Java Application Building Center (jABC), which is a framework for extreme model driven design (XMDD). The XMDD approach focuses on bringing software development closer

to conceptual design, so that the functionality of a software solution could be understood by someone who is unfamiliar with programming concepts. The transformation addresses the heterogeneity challenge and enhances the re-usability of the system.
For enhancing the e-participation of functionally illiterate users, two detailed studies were conducted in the Republic of Rwanda. In the first study, the traditional (textual) interface was compared with the virtual character-based interactive interface. The study helped to identify usability barriers and users evaluated these interfaces according to three fundamental areas of usability, i.e. effectiveness, efficiency and satisfaction. In another study, we developed four different interfaces to analyze the usability and effects of online assistance (consistent help) for functionally illiterate users and compared different help modes including textual, vocal and virtual character on the performance of semi-literate users. In our newly designed interfaces the instructions were automatically translated in Swahili language. All the interfaces were evaluated on the basis of task accomplishment, time consumption, System Usability Scale (SUS) rating and number of times the help was acquired. The results show that the performance of semi-literate users improved significantly when using the online assistance.
The dissertation thus introduces a new development approach in which virtual characters are used as additional support for barely literate or naturally challenged users. Such components enhanced the application utility by offering a variety of services like translating contents in local language, providing additional vocal information, and performing automatic translation from text to sign language. Obviously, there is no such thing as one design solution that fits for all in the underlying domain. Context sensitivity, literacy and mental abilities are key factors on which I concentrated and the results emphasize that computer interfaces must be based on a thoughtful definition of target groups, purposes and objectives.

Dedicated to the research community struggling hard to make interfaces available for deaf and functionally illiterate users

Acknowledgements

I would like to thank Prof. Paolo Prinetto who expanded my vision by providing opportunities to work with different partners but kept me aligned with goal of providing useful interfaces for deaf and functionally illiterate users. His suggestions and reviews opened new horizons to think differently with problem solving approach. Under his guidance, I analyzed the requirements of deaf and low literacy users, understood the usability barriers faced by them to access online services and designed easy to use interfaces for them.

I am also thankful to Prof. Tiziana Margaria who helped me to extend my work according to good software engineering practices and I organized my work in a way which enhances its re-usability. I would like to thank Dr. Barry D. Floyd for his feedback to refine my writing, reorganizing the experiments conducted in Rwanda and presenting the results in more rigorous way. I would like to thank Dr. Anna Lena Lamprecht, Alexander Wickert and Samih Alariki who enhanced my understanding about workflow modeling in Java Application Building Center (jABC), a framework for extreme model driven design (XMDD). Special thanks goes to Mr. Umar Shoaib, who is my fellow and we worked together as team members and he has given valuable suggestions to improve the experiments conducted in Rwanda and we jointly extended Italian Sign Language (LIS) lexical resources with MultiWordNet. I am thankful to Marco Elena from University of Torino, Italy who helped in designing pre-experiment questionnaires and Mr. Bilal Saeed from Yuasa Investments Ltd, Rwanda for organizing experiments as a mentor at vehicle refurbishing sites in different districts of Rwanda. I am also thankful to all staff of Department of Computer and Control Engineering, Politecnico di Torino and Service and Software Engineering Chair of University of Potsdam for their full support and cooperation during my stay in Italy and Germany.

Finally I want to thank my family, especially my father who worked hard and supported me but left in peace (May God shower his countless blessings on him) during my 2nd year PhD study. I am greatly thankful for all prayers of my mother and moral assistance of my elder brother. I would like to thank my wife Naila Saleem, her encouragement, support and volunteer duty to look-after our son Muhammad Saim Nadeem during my study timings which ultimately increased my devotion.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The human machine interface (HMI) determines a way in which human beings can interact with machines. Users provide input to the machines through its interface and machines provides output in response. Interfaces are the point of interaction with smart devices which build a relationship with the users. A variety of interfaces from airplane control systems to nuclear plants and automatic teller machines to mobile phones accepting input through mice, keyboards, microphones, eyebrows and brain signals make the process of interaction more complex. Interfaces in sophisticated machines hide the actual implementation from users but effective interfaces must keep the users informed about the progress of the work with different indicators and should map the task with users mental model.

'Deaf' or better to be called naturally challenged users mostly have great hearing loss. For most deaf and hard of hearing users, their communication relies on sign language (SL) as the first language. If a person with congenital deafness has not learned sign language then he may be excluded from social interaction of deaf community [5]. Sign language is based on the combination of movements of hands, arms, body and facial expressions. Italian Sign Language (LIS) is the primary means of communication for about sixty thousand deaf people living in Italy [6]. LIS is a natural language and can convey the range and variety of meanings and undergone a natural development like the spoken languages. Grammatically LIS has its own complex phonological, morphological, and syntactical structure [7]. The LIS linguistic features are different from spoken languages, due to the existence of several components affecting the context, such as the use of facial expressions, head movements and different combinations of hand movements [8]. Inclusion of deaf people in the society aims at providing them access to services in their own language and spreading signed language knowledge among the communities.

Different definitions for literacy exist and it is an ongoing debate. The National Center for Education Statistics(NCES) of United States defined literacy as "using printed and written information to function in society, to achieve one's goals, and to develop one's knowledge and potential" [9]. Further the literacy is categorized into three scales i.e. prose literacy, document literacy and quantitative literacy. Prose literacy is about searching, understanding and using information from continuous text, while document literacy is about understanding the text from non-continuous texts in various formats. Quantitative literacy is about the ability of arithmetic operations. Functional illiteracy means inadequate reading and writing skills to manage daily life. Functional illiteracy is distinct from illiteracy as illiterate person absolutely can not read or write. A functionally illiterate person might be incapable in using information and communication technologies (ICTs), reading and comprehending job advertisements, past-due notices, newspaper articles, banking paperwork, complex signs and posters.

My research is motivated by objective to provide applications with easy to use interfaces to communities with low literacy and deaf users, which enables them to work independently without acquiring any assistance. Functional illiteracy does not means inability rather it refers towards lack of particular skill which can be developed by providing some assistance [10]. I believe, if the user Interfaces (UIs) are well designed and helpful in nature then users do not require much formal literacy, computer skills, or any external assistance. The research in this context is a multidisciplinary activity consisting of both technical and social-science research.

The chapter is organized into four sections. Section 1.1 highlights the importance of ICT in daily life and provide global statistics about literacy. Section 1.2 discusses the research challenges in designing interfaces for deaf and functionally illiterate users. Section 1.3 discusses the User Centered Design methodology for implementing people centered interfaces. In Section 1.4, the author has discussed his contribution for enhancing e-participation of deaf and functionally illiterate users.

1.1 Information and Communications Technologies for Development

ICT have integrated widely into daily life, including communications, commerce, entertainment and the service sector. These technologies have created new employment opportunities, and provided the countries and individuals with another means for economic development. ICT helps in increasing the education level, alleviation of poverty and improving the health care system. It has a powerful positive impact on human development, health care system, training of medical personnel and teachers, provision of equitable access to education and training facilities, opening opportunities for women, and expansion of the scope for citizens' participation. ICT deals with disadvantaged populations and plays a central role in socioeconomic development [11, 12].

ICT with system applications such as the world wide web (WWW) has significantly contributed towards improving the possibilities for social interaction, activities and integration in the society, particularly for people with disabilities. The participation and accessibility of citizens to the technologies is necessary, but literacy is an obstacle and limits the access to these resources. It is an established fact that real value of life can be provided to low literate communities of developing countries through ICT development in divergent domains. Whereas the experience of transfer of technology from developed countries to under-developed ones had not been very successful due to mismatch of literacy level, user requirements and intended objectives [13].
As computer-based systems proliferate around the world, an increasing number of functionally illiterate and deaf people come into contact with systems they do not understand. Almost 774 million adults and 123 million youths worldwide are illiterate and almost two-thirds of them (62%) are women [14]. Adult literacy rates are concerned with the population aged 15 years and older, while the youth literacy rates cover the population between the ages of 15 to 24 years. More than half of those (412 million) live in Southern Asia while 176 million adults are living in sub-Saharan Africa. Figure 1.1 shows the global distribution of adult and youth literacy rate. On the other hand there are 60 thousand deaf users only in Italy, 3 million in France and 1 million in Germany [15].
Many services including banking, online shopping and booking, taken for granted in the developed countries are often missing and not accessible in the developing world. Computers, being multimedia devices, can play a significant role in the life of deaf and functionally illiterate people exploiting graphics, animation, and audio. Computer applications have the potential to be wholly intelligible to an inexperienced or illiterate users [16].

1.2 Research Challenges

Lack of competency in linguistic and analytical skills put a barrier for functionally illiterate and deaf users to access ICT facilities. Interfaces between technology and society should be designed according to the abilities of end users, as the level of understanding is different among educated, functionally illiterate and deaf users. Availability of technology does not guarantee that users will be able solve their problems by using ICT. Excessive usage of text in applications and world wide web is an accessibility hazard for those whose first or primary language is not English.

1.2.1 Research Issues of Deaf Users

There are significant challenges in addressing the marginalization of persons with disabilities from the mainstream economy and society. These challenges include

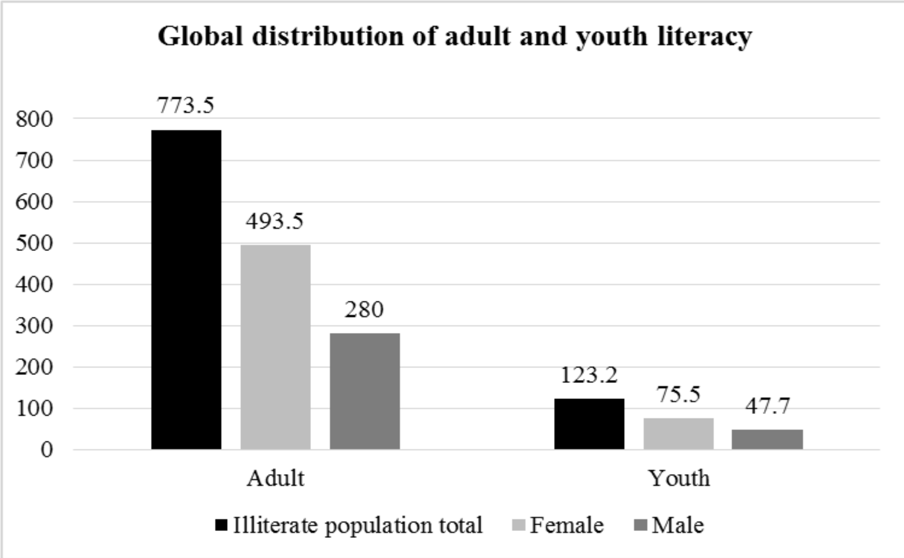


Figure 1.1. Global distribution of adult and youth literacy.

access to information and services for people having some kind of disability. Determining user requirements concerning the characteristics of deaf individuals, their difficulties and needs, is a complex task because the users often find it difficult to express the entirety of their needs and the information provided may also need a psychological and cognitive interpretation to be complete and accurate. For this reason, it has been necessary to draw information from specific literature and studies conducted on deafness.

For normal users voice plays an important role, from childhood voice is a way to get their attention and all information is perceived vocally. The person who got deafness in early childhood or born as deaf, its difficult for him to learn oral languages due to the missing voice modality. As a result vocabulary develops slowly in deaf children and they usually learn concrete words (jumping, eating and running) more easily than abstract words (before, after, honor and respect). In a survey it is revealed that the basic impoverishment of deafness is not lack of hearing but lack of language. In the age of 4, usually the kids learn vocabulary words between 2000 to 3000 and knows little syntax and grammar that how to combine different words in meaningful way, while the deaf kid in same age just have very few words in his vocabulary [17]. Deaf users usually have short term oral language memory skills and its difficult for them to understand verbal analogical reasoning, which requires high-level linguistic skills, and the ability to understand not simple items but complex structures [18]. Its an important challenge to provide the visual material to support deaf users' comprehension and learning in education context.

Sign languages are visual gestural languages and the mother tongue of deaf people.

Sign languages vary among countries, areas and cultures. Sign language is not a universal language; each country has its own sign language such as American Sign Language (ASL), British Sign Language (BSL) and Italian Sign Language (LIS). In some countries more than one sign language is used e.g. in Switzerland, Swiss-German, Swiss-French and Swiss-Italian Sign Language(s) are used [19]. These various forms of SLs have their own vocabulary and grammar, which are quite different from oral languages [20]. Moreover variability is also found in the sign language within the same country as there is no standardized lexicon dataset and set the rules.
An increasing request for sign languages interpretation in educational, legal and health care context is foreseen [21]. This is highly expected to extend to the culture and entertainments. As of today, many language resources available for spoken language users are not available in SLs and there is a significant lack of applications (like complete online dictionaries, databases and translators) which provide access to sign language features and resources. Moreover, if these applications are available, they lack functionalities like the ability to look up LIS lemmas or explore the sign associated with the words of the same or similar sense. When people encounter a word whose corresponding sign is not available, they have to find it in a conventional paper dictionary or rely on online dictionaries having limited scope.
The Radutzky paper dictionary [2] shows pictorial representations of the body position, orientation and movements with the help of arrow keys. These indicate the direction of movements of the sign and are not easy to understand for someone who does not know SLs [22]. The existing LIS dictionaries lack the information about synonyms or related words. In conventional paper lexical resources, it is difficult to search for an SL lemma because these are not listed alphabetically. The existing electronic (e-lis) dictionary allows searching for sign of the word but they are generally very limited. They do not take into consideration the regional variations. For example, the LIS sign for "casa" (house) is different in different areas of Italy.
Research into SLs has grown considerably over the past few years and with time the SL linguistics research community has given more and more importance to SL lexical resources. However, the progress in the field has been slowed down due to some data sharing and processing challenges. The reliance on the intuitions of very few informants and detached textual examples (which are not accessible for peer review) in the past have been problematic in the field of SLs [23]. The research on sign language machine translation deals with several issues, generally related to both the translation and the generation of signs. Most of the translation approaches (e.g., television closed captioning or teletype telephone services) assume that the viewer has strong literacy skills [24] but reading the captioning results is difficult even if deaf users are fluent in sign language. In most TV and entertainment programs, if a live interpreter is not feasible, the translation is avoided for deaf people. A new approach, which takes into account building a system that is able to produce

both the translation and the visualization of signs at the same time, is thus very important.

1.2.2 Research Issues of Functionally Illiterate Users

The normal computer interfaces including operating systems and productivity suite like Microsoft office are relatively complex, with their metaphors of file paths, menus and written text. Functionally illiterate and deaf people are generally less familiar with complex technologies. Functionally illiterate people usually read word by word and they spend more time on word recognition rather understanding their meanings and can not store more information in their short term memory [25]. Functionally illiterate people more concentrate on examples rather their interpretation. They usually skip difficult letters, chunk of sentences or try to avoid reading altogether. Functionally illiterate users have a lower ability to understand and follow-up written instructions, lower cognitive processing speed and lower organizational skills in ICT [26].
Functionally illiterate users become confused when using the web, especially search engines to search for desired results. They believe that the first available option in the search result is the best. Their performance is usually lower as compared to high literate users, since they take longer to finish tasks, spend more time on the same page and visit more pages to find the solution of their problem [27]. Language is a major hurdle for functionally illiterate users to access ICT services such as banking and traveling. Due to the difficulty found in reading and comprehending online text, functionally illiterate users rarely try to find the solution of their problems through world wide web [28]. Success of e-services is dependent on e-participation and access to the common person.
A person who cannot read and understand online information, requires assisting interfaces to compensate his or her limitations. IBM India's research laboratory tested the idea of spoken web [29]. The spoken web is a network of interlinked voice based sites. The users can create their own websites by simply talking and providing the contents to a voice application. The generated sites will be in local language and it was possible to link the newly created site with other sites using hyper-links. It was designed to work with voice as the primary user interface modality and required the ordinary telephone instrument to make calls.
Voice-based interfaces significantly increase users' perceptions of flow and generally helps them in speed of comprehension. Audible instructions should be short and multiple instructions should not be passed together because functionally illiterate users remember either the 1st instruction or the last instruction [30]. In spoken dialogue systems the key challenges include the development of speaker-independent automatic speech recognition (ASR) systems and text-to-speech (TTS) systems that

function reliably as speech recognition and text to speech technology has not been fully developed especially for regional languages.
Lack of continuous user involvement in interface design and weak analysis of end user abilities further distracted already ignored communities of deaf and functionally illiterate users. There is a need to design user-centered interfaces that are best suited for each community.

1.3 People Centered Interfaces

The people who benefit from computers and internet access are generally younger people, instead of the people who are illiterate, disabled or older or working in remote areas. ICT increases educational activity among those who are already learners rather than widening participation to include those who had previously not taken part in formal or informal learning. Technologies used must be adequate to the skills of the poor in order to exploit their potential effectively. The objective of people-centered interfaces is to achieve universal access to information. Usability is an indicator to show the users' experience about a system regardless the user's abilities and the language of the system.
The International Organization for Standardization (ISO) defined six core human centered design principles for interactive systems that communicate the essence of the people-centered approach and serve as the framework for individual methods and techniques [31]:

1. The design is based upon an explicit understanding of users, tasks and environments.

2. Users are involved throughout design and development.

3. The design is driven and refined by user-centered evaluation.

4. The process is iterative.

5. The design addresses the whole user experience.

6. The design team includes multidisciplinary skills and perspectives.

People-centered interfaces improve the system usability, produce less errors during usage, and faster learning times. The user-centered design process focuses on the product's potential users from the very beginning and checks at each step of the way that product designing is according to user's requirements.

1.3.1 User Centered Design

The user-centered design (UCD) approach is both a design philosophy and a process in which the needs, wants, and limitations of end users of a product are given extensive attention at each stage of the design process. It can be characterized as a multistage problem solving process that not only requires designers to analyze and foresee how users are likely to use an interface, but also to test the validity of their assumptions with user's behavior [32].
The primary concern of UCD approach is to discover who the actual users are, what objectives they have and what is the context where the product is used. In other words, in UCD all development proceeds with the user as the center of focus [33]. Although UCD is not a complete methodology for software development when considering the whole life-cycle and activities of a project; it improves the user's experience associated with project and reduces the cost of future redesign. The goal of UCD is to produce products that have a high degree of usability and optimized user's experience of a system, product, or process.
There are four essential activities in a user-centered design approach. These activities describe the general process to include user centered activities throughout software development life cycle.

- Requirements gathering - Understanding and specifying the context of use.
- Requirements specification - Specifying the user and organizational requirements.
- Design - Producing designs and prototypes.
- Evaluation - Carrying out usability testing with actual users.

In UCD all features should be excluded that provide no real benefit or value to the user experience. In UCD prototypes are used to translate requirements into usable interactive systems. The UCD approach is adopted to enhance e-participation of deaf and functionally illiterate users of the developing countries. This dissertation presents an idea to develop interfaces in such a way that in 1st contact with the computer the users can recognize useful interaction and enable them to work independently without any human assistance.
The classical software development approach is no longer adequate for heterogeneous, cross organizational systems which must adapt to rapidly changing requirements [34]. An eXtreme Model Driven Design(XMDD) is the agile methodology which puts the user-level process in the center of the development. It provides liberty to end users to design and control their processes in the whole software development life cycle, starting from requirement analysis to demand driven process evolution [35]. The UCD methodology is used to design interactive interfaces for

functionally illiterate users while XMDD is followed in modeling translation process from Italian text to sign language for deaf users.

1.4 Thesis Contribution

The dissertation focuses on two case studies, one is about deaf users in which they were offered a gestural visual interface with LIS interpreter that reproduces the LIS movements and the second is about functionally illiterate users, in which different helpful interfaces were designed to compare their usability from end user's perspective. In this research existing tools in voice, web and multimedia technologies were investigated to identify learning gaps.

1.4.1 Enhancing Sign Language Lexical Resources for Deaf Users

There is a variety of simple lexical resources for SLs available in the form of SL dictionaries, containing a basic set of written words, each word accompanied by a line drawing or a photograph of person signing [36]. Clearly, there is need for the development of lexical resources containing more signs and additional information about the signs. Dictionaries of Italian language bring indicatively 250 thousand words, while there are few thousand signs in the LIS. For example, the Radutzky dictionary [2], considered a reference in Italy, defines about 2500 signs.
To overcome the need of lexical resources, a new Sign Bank is developed that contains hundreds of animated video sequences for distinct sign language signs [1]. The sign language lexical resources can be accessed through web to support deaf people in learning of both sign language and written language. The interface of the platform-independent LIS dictionary which is based on Sign Bank is developed for deaf users [37, 38]. The sign language dictionary accepts Italian lemmas as input and provides their representation in the LIS as output. The sign language dictionary contains 3082 signs as set of avatar animations in which each sign is linked to a corresponding Italian lemma. The LIS Dictionary is implemented in such way that its functionality is enhanced according to the availability of resources. It is available in the form of a web application online and as a mobile application for Android [39].
A great problem of research is "mapping and linking of knowledge networks". Multilingual sign language knowledge bases address the development of language resources by capturing the relationship between words and their corresponding concepts in sign language and hence making the semantic connection between lemmas of both languages. There are WordNets in more than 50 different spoken languages including, English, Italian, Spanish and German but there is no one that provides information

about the corresponding sign language.
The LIS lexical resources were integrated with MultiWordNet (MWN) database to form the first LIS MultiWordNet(LMWN) [40]. LMWN contains information about lexical relations between words, semantic relations between lexical concepts (synsets), correspondences between Italian and sign language lexical concepts and semantic fields (domains). The LMWN not only provides the LIS meanings of the lemma but also provides the set of words with similar meanings and the sign associated with each lemma in the database. All semantical and syntactical relationships of LIS signs with MWN are defined and the words existing in both databases are determined. The approach enhances the deaf users' understanding of written Italian language and shows that a relatively small set of lexicon can cover a significant portion of MWN. Integration of LIS signs with MWN made it useful tool for computational linguistics and natural language processing.
Machine translation is a hard task because languages are very complex. Thus for the sake of accuracy most translation systems in sign languages are domain dependent. In Italian text to sign language translation, the the main focus was on weather forecast domain in which the translation is performed resorting to rule-based and statistical translators. The rule-based translation process from Italian written text to LIS is transformed into service oriented approach. The translation procedure is graphically represented by Service Logic Graphs (SLGs) which are executable control-flow models composed of so-called Service Independent Building Blocks (SIBs). A SIB can encapsulate a single service, but also another SLG as subgraph, thus serving as a macro that hides more detailed and basic steps [35]. A service is any piece of functionality that is directly executable and can be published for use in complex applications [41]. The translation process is composed of various modules including parser, semantic interpreter, generator, and spatial allocation planner. All these modules have been implemented as services and encapsulated in SIBs. The Java Application Building Center (jABC), which is a framework for service oriented application development based on XMDD, is then used to compose the services (represented by SIBs) into a process model (in the form of an SLG) and execute it.
The process of transformation addresses the heterogeneity challenges of linguistic translation modules and integrates them into translation workflows. These workflows will improve the flexibility to entertain languages other than Italian. In addition offering translation modules as loosely coupled components will improve the re-usability. It also provides greater flexibility since the workflows can be adapted, extended and changed at a user-accessible level.

1.4.2 Enhancing E-Participation of Functionally Illiterate Users

For enhancing the E-Participation of functionally illiterate users, two detailed studies were conducted in Republic of Rwanda. In first study, traditional (textual) interface was compared with the virtual character based-interactive interface. Both the interfaces, the original one (based on text instructions) and the newly designed (based on verbal instructions in regional language pronounced by a virtual character to provide the help at each step), were tested on two separate groups composed of 50 functionally illiterate users in each group. The study helped to highlight the usability barriers and users ranked both interfaces according to three fundamental areas of usability i.e. effectiveness, efficiency and satisfaction. Although the participation of functionally illiterate users is encouraged by virtual character-based interfaces, which specific feature played the key role (whether it was translation in local language, vocal help or sense of virtual human assistant) remained unclear.

In another study four different interfaces were developed to analyze the effects and usability of online assistance (consistent help) on functionally illiterate users and compared different help modes including textual, vocal and virtual character on the performance of semi-literate users. In the experiment, 250 functionally illiterate users participated, users qualification and IT exposure is measured in pre-experiment questionnaire, 70 subjects were discarded due to inappropriate skills, either found completely illiterate or having already sufficient ICT skills. The remaining 180 users were randomly divided into four different groups. Each group is further subdivided into three subcategories of basic, moderate and expert users. The task involved filling online form by providing basic information such as name, gender and level of education to apply for an internship program.

In the newly designed interfaces the instructions were automatically translated in Swahili language. All the interfaces were evaluated on the basis of task accomplishment, the time consumed, System Usability Scale (SUS) rating and number of times help required. System Usability Scale is a reliable, low-cost usability scale that can be used for global assessments of systems usability [42]. The results show that the performance of semi-literate users improved significantly when using the online assistance. Moreover, the percentage of completed tasks increased from 52% to 94%. Statistical analysis of the System Usability Scale (SUS) indicates that their average subjective usability score boosted from 39 to 80.

The proposed virtual character based interfaces not only assisted deaf users but also enhanced the applications utility for functionally illiterate users by translating contents in their own language. The interfaces presented in this thesis are less cluttered with text compared to most of the complexity of normal applications. The interfaces should be designed in a way that it should enhance the leranability of low literate users and should help them to complete their tasks with more ease and comfort.

1.4.3 Thesis Structure

The thesis is organized into following chapters.

- Chapter 2 provides some background information about scarcity of sign language linguistic resources, discusses their limitations and describe implementation details of platform independent Italian Sign Language Dictionary based on rich collection of signs.

- Chapter 3 discusses the expansion of linguistic resources by developing multilingual sign language knowledgebase. It covers the approach which links the Italian lemmas and LIS signs to extract and display bilingual information from the collected lexicon and the semantic relationships of LIS Signs with MultiWordNet.

- Chapter 4 discusses the transformation of translation from Italian written text to LIS as service oriented paradigm. It also describes how the service oriented implementation of core workflows enhances the re-usability of the system and produces the flexibility to integrate it with languages other than Italian.

- Chapter 5 provides the in depth study of usability barriers and challenges faced by illiterate or low literate communities in using ICTs. It analyzes the previous recommended design methodologies with user centered design and describe some initial results of virtual character based interfaces tested by slums of Rwanda.

- Chapter 6 presents the idea of localized helpful interfaces in which multimedia help is provided on each screen. Different textual, oral and visual help modes are compared and user performance is evaluated with different usability methods. The chapter summarizes the guidelines, effective in designing interfaces to enhance the e-participation of functionally illiterate and deaf users

Chapter 2

Extending Italian Sign Language Lexical Resources

The sign languages(SLs) are based on the gestural-visual modality with signs as lexical units instead of words. All SLs, as oral languages, exhibit a grammatical structure at all linguistic levels including phonology, morphology and syntax, but differ from spoken languages on several aspects. In particular, all visual-gestural languages possess a rich morphosyntactic structure organized in space, which differs from the sequential ordering of the sentence elements in verbal languages [43]. Respect to spoken languages, which feature a sequential modality, sign languages communicate several information spatially and simultaneously.
This chapter provides insight about communication ways adopted by deaf users in section 2.1, different sign writing notations are discussed in section 2.2, Italian Sign Language, the Radutzky dictionary, introduction of ATLAS project and limitation of existing lexical resources are discussed in section 2.3, the development of platform independent LIS Dictionary and its business use cases are discussed in section 2.4, in section 2.5 the details about lexicon creation process will be discussed, finally section 2.6 discusses the architecture of LIS dictionary and its deployment on web and mobile platforms.

2.1 Importance of Sign Language for Deaf Users

Normally people use their hearing skills to monitor their environment and focus on tasks visually in response to hearing alert, while deaf users perform both tasks visually. Home sign or kitchen sign is the communication system that is developed automatically in deaf users while communicating with hearing deaf family members. Home sign gestures are quite different from sign language and mixture of oral and sign language. More sophisticated kitchen sign language is developed if there are more children deaf in a family. In kitchen sign language no consistent relationship is found between gestures and word meanings. Every child grows with own gestures so there are communication problems with other deaf people as no large group adopts the same signs. Kitchen sign language can be considered a starting point to learn sign language and it helps in faster learning of SLs.

Many studies document the linguistic difficulties among deaf children when compared to their hearing peers [17, 44, 45, 46]. The deaf users get comparatively lower learning opportunities, routinely they are unable to perceive the information which is being discussed in their home environment and they also suffer from isolation and loneliness. Deaf persons can not listen to their own words when they speak and have lower ability to mimic the pattern of voices. Most deaf individuals do not achieve the same degree of educational attainment as hearing people do [47, 48, 49].

When deaf people read and process a written text, their lower ability to recognize and process the basic grammatical morphemes of the written language (such as articles, prepositions, conjunctions and pronouns) leads to a poor exploitation of the semantic and the pragmatic information necessary to reconstruct the meaning of the global sentences [50]. As result of the hearing loss and aural degraded linguistic input, deaf children since birth have more difficulties in acquiring both the language lexemes and the morphosyntactic aspects of the oral language.

Deaf individuals consider their sign language a crucial aspect of their cultural identity. The same consideration holds for hearing children born to deaf parents, who are therefore native signers hearing, and for deaf individuals who had not the chance to acquire a sign language in their first years of life but later on, usually in adolescence, develop signing skills. The deaf community provides to its members more social and emotional support compared to the hearing community [51]. Deaf people can easily integrate in deaf communities by learning sign language .

2.1.1 Communication in Deaf Users

In the past only face to face communication was possible among deaf people. After introduction of telephone services for hearing people, there was widespread demand for similar service for deaf users. In last 50 years deaf people have been able to communicate through a particular telephone system called teletypewriter (TTY, sometimes TDD, telecommunications device for the deaf) [52]. The deaf person with TTY can send and receive messages to other users but the limitation of this service is that both users must have TTY and the process is also quite slow. Although TTYs are useful for deaf to deaf conversation, deaf users are unable to communicate with hearing people and Government agencies with TTY. Many countries started relay services in 1990, in which an operator was receiving messages from deaf users through TTY, and then conveying that message to hearing person vocally and finally, sending the response back to deaf users [53]. This system was widely accepted by

deaf individuals who, either with moderate or severe hearing loss, had no chances to use the telephone: for the first time deaf individuals were enabled to communicate with deaf and hearing individuals.
A more advanced form of relay services is video relay service (VRS). With the advent of computer technology, the computers have taken the place of TTYs. In VRS, the deaf users communicate with sign language interpreter by web-cam via high speed DSL connection. With both the operator/interpreter and the user in view of each other through cameras linked to monitors, manual communication improves the functional needs of deaf users. This device employs a visual system which is preferable for deaf people in comparison to the written language used with TTY [54]. VRS has been provided in most of European countries, but still some countries have problems of legislation or the financing for large-scale VRS and to provide necessary facilities to use VRS.
In summary. the computer mediated communication revolutionized the deaf users' life and now they are able to communicate with other deaf users in sign language too, by using computers and web cameras.

2.2 Representing Sign Languages

Oral languages have clear representation which facilitates linguistic processing, storage and comparison with other knowlegebases. The problem to represent SLs is tedious and has a variety of challenges. Due to the multidimensional visual-gestural forms of Sls, the writing system of oral languages is not sufficient for representing SLs. It is also not possible to devise a new writing system for SLs which is not widely accepted or adopted in deaf communities [55].
The Stokoe Notation System (SNS) [56] was a first attempt to describe the American Sign Language (ASL) as set of symbols and still widely used. Later on extensive expansions and modifications were suggested in Stokoe Notation System. Friedman's Notation System [57] considered description of SNS inadequate and presented alternative phonetic and phonemic analysis for hand configuration, place of articulation and orientation. An abstract view of some modern sign writing systems including SNS, Sutton's SignWriting [58] and Hamburg Notation System [59, 60] is discussed below.

2.2.1 Stokoe Notation System

Stokoe notation is first groundbreaking work in which different symbols were defined by combining Latin letters, numerals and different shapes to define the hand movements of the ASL. Language. The dictionary on ASL [61] was a major turn in SL studies which employed Stokoe's insight that lexical items can phonologically

be decomposed just like the words of spoken languages. SNS is written horizontally from left to right but Sutton's SignWriting is written vertically from top to bottom. In SNS, the three dimensions of ASL (location, shape, and movement) are considered to categorize features of sign-language units known as cheremes. Overall 55 different symbols are organized into three groups of tab ("tabula" or sign location), dez ("designator" handshape), and sig ("signification" motion).
Signs are described according to the formula TDs, where T represents tab, D for dez and s for sig similar to the distinction of consonant, vowel, and tone used in oral languages. SNS gives iconic symbols for 12 distinct locations showing the position of head, arm, and torso. SNS recognizes 19 different hand shapes and their orientation. SNS classified 24 types of movements and a subset of these symbols is also used for hands orientation whereas the multiple movements are arranged linearly. In Figure 2.1, the translation of a sentence "the woods, there is a house sitting on a hill. (If you) go in" is shown with the help of SNS.

$\overline{B}_{\alpha}\surd B_{\wedge\omega}$	$G^{\perp}$	$B_{\wedge}{}^{I}B_{\wedge\vee}^{\div}$	$\square\dot{A}^{@\times}$	$\underline{B}_{p}B_{p}^{\perp}$
WOODS	UP	HOUSE	SITTING.THERE	ENTER

Figure 2.1. A sample sentence represented in Stokoe Notation System: "the woods, there is a house sitting on a hill. (If you) go in".

In spite of the remarkable work, SNS cannot be used for segmenting individual signs and there is no provision to represent signs in natural flow. A serious deficiency of SNS is that it does not provide symbols for many phonetic details, such as uneven rate of signing, tenseness or laxness and other non-manual components including facial expression, mouthing, eye gaze, and body posture that are crucial to the actual formation of a sign. SSN consider every movement which is not straight as circular and unable to properly present curved movements. SSN provide 12 iconic symbols for locations while Liddell and Johnson found 56, similarly SSN can represent 19 different hand shapes while Liddell and Johnson found 150 different hand shapes [62]. Furthermore SNS is unable to deal with spatial locations as theoretically there are infinite relations in hand and head movements and SNS can not provide infinite symbols while the pictorial nature of Suttons's SignWriting (SSW) deals with such situations [63].

2.2.2 Sutton Sign Writing

Valerie Sutton initially defined a notation for dance choreography in which palm orientation, signing space, facial expressions, punctuation, and grammar are all depicted pictorially. SSW is a highly iconic featural script and its spatial arrangement does not necessarily follow sequential order like in oral languages. No phonemic or semantic analysis is required to write SSW as it follows physical representations of signs as signer conceives it. Contrasting to SNS, the SSW facilitates all three representations of a sign, the initial position, the movement and ending position while SNS only represent initial position and describe the ending position if required.
In SSW there are multiple ways to write a single sign and any sign language of the world can be represented with it. SSW covers 10 dimensions of sign languages including hand shapes, orientation, movement, speed of movement, face, upper-body, full-body, head, limb, location, dynamics, and punctuation. The 55 symbols of SSW can be configured in multiple ways. The SSW provide more phonetic details than SNS. Figure 2.2 shows SSW notation of a nursery rhyme, “Jack and Jill”, translated into ASL. The SSW represented the complete nursery rhyme (‘Jack and Jill went up the hill, to fetch a pail of water, Jack fell down and broke his crown and Jill came tumbling after”) in two dimensional layout, usually the symbols are organized linearly in sign and oral languages.

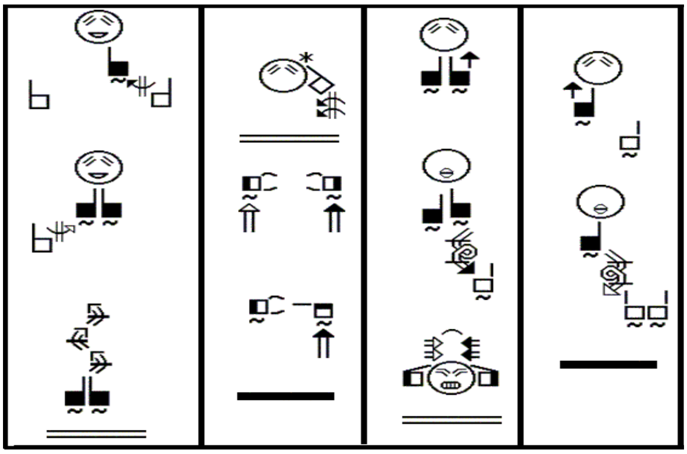


Figure 2.2. Sutton SignWriting Notation(SSW) for the nursery rhyme, “Jack and Jill”, translated into ASL.

In SSW the symbols can be repositioned, twisted or shaded to represent slight variation of SLs [64]. For each new shape, SSW reproduces the sign while SNS force to rely on existing set of symbols. SNS chooses arbitrary shapes based on ASL alphabets and numbers for hand shape parameters, which are not valid for such sign languages that have no manual alphabets like Chinese Sign Language or completely

different one like Australian Sign Language [63]. In SSW the hand-shapes can be dictated according to the grammar which solve the problem of representing signed pronouns.
Both SNS and SSW have a problem that none of them can be directly typed on keyboard, SNS has provided a way to transcribe its symbols in ASCII while SSW has more worldwide recognition and it is the first sign writing language which has been proposed for inclusion in Unicode.

2.2.3 Hamburg Notation System

The Hamburg Notation System (HamNoSys) is published in 1987 by the University of Hamburg, having roots in SNS and describing signs mostly on a phonetic level [59]. The HamNoSys was intended to be used for worldwide sign languages and does not rely only on ASL conventions. Another aim of HamNoSys was to introduce a notation system which can be easily represented and processed at computer machines, so the HamNoSys symbols are available as a Unicode font.
A HamNoSys notation represents the initial posture of the sign and actions which are changing the state of the sign. HamNoSys describes 6 hand shapes as base forms which are combined with 3 diacritical symbols for thumb position and 4 hand shapes for bending. In addition, intermediate forms can also be described if involving fingers deviate. This approach covers all possible hand shapes of existing SLs. HamNoSys has been used first time in the Hamburg Animated Dictionary of Signs in 1987. In this dictionary the two dimensional animations are created on the basis of notations [65]. HamNoSys uses almost 200 iconic symbols to represent hand shape, hand orientation, location relative to other parts of the body, and movement. HamNoSys provides a symmetry operator to describe two handed-shapes and also support non manual components. Figure 2.3 represents a sample sentence ("while reading the newspaper, I had a cup of coffee") in HamNoSys.

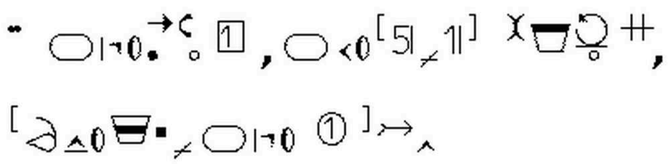


Figure 2.3. A sample sentence represented in HamNoSys: "while reading the newspaper, I had a cup of coffee".

HamNoSys has not given extensive support to facial expressions, which is being considered an important phonological base of many SLs. Signing Gesture Markup Language (SiGML) which is an extension of Extensible Markup Language (XML) is

developed on the basis of HamNoSys. SiGML can produce animated signing avatars from the text being provided as input.
The notation systems based on SNS including HamNoSys have some limitations as it is difficult to use these notations for transcribing sequence of signs and actual signed discourse. Moreover they underestimate the role of non manual components in SLs [66].

2.3 Italian Sign Language

The European Parliament approved a resolution in 1988, in which it insisted the member states to officially recognize sign language as the language of deaf users [67]. Italian Sign language (Lingua dei Segni Italiana, LIS) is not yet officially recognized, but it is source of education and communication in Italian deaf communities. Although symbolic recognition does not ensure any provision of services, official recognition will encourage researchers to develop it like other linguistic domains. As the research is progressing in sign languages, many countries are taking initiative to recognize sign language as national language. In 2011, the Parliament of Iceland unanimously declared Icelandic Sign Language as an official minority language. LIS is used by deaf communities not only in Italy bust also in Ticino a region in the south of Switzerland and in San Marino. Lexical variations are also reported in different parts of Italy.
LIS is characterized by precise grammatical, morphological and syntactic rules, very different from Italian oral language rules. Some of these, referable to typical characteristics of the visual-spatial communication, are common to other sign languages, others are peculiar to LIS. In LIS the agreement among nouns, adjectives and verbs is not based on gender rather it is based on the place in which the sign is performed. The types of the signs are categorized according to the following four functional parameters.

- Configuration: form assumed by the hands.
- Location: palm orientation, the spatial property of a place with respect to body and position of the hands in two handed signs.
- Articulation place: region of space where the sign is reproduced.
- Movement: direction, manner, contact and interaction of the hands in performing the sign, for example upward, circular, undulated etc.

Facial expressions, head tilting, shoulder movements and other gestures which support hand movements are non-manual markers. The functional parameters are very important to classify the LIS signs. For example, two very similar signs that have only a different functional parameter, as direction of the movement or use of the space, could assume two different meanings.

2.3.1 The Radutzky Dictionary

The LIS Dictionary was created by Elena Radutzky [2] who spent years collecting LIS signs and their variants. The Radutzky dictionary is officially used for Italian Sign Language, but the resource is very limited and contains less than three thousand signs. Each sign is encoded resorting to symbols that explicitly transcribe each parameter. This particular transcription system of the sign is called "Radutzky Notation" (RN). The RN is a notation system developed on the basis of SNS [56] which makes it possible to write LIS signs by means of special characters. It is considered a de facto standard in the deaf community. Generally, the RN of a sign can be divided into a base and an exponent. As mentioned above, the parameters of the sign are phonological components, so with the RN it is possible to transcribe the phonological components of the sign. Figure 2.4 describes the parameters of the sign: configuration (hand shape), location of the signing and direction and angle of movements are shown with the arrow keys. The signs are associated to the "cheremes", which are the phonological units of SL, as we have phonemes for the spoken languages. The phonological information includes different characteristics of the sign. The movements have varying values and its value may vary during the sign or remain static. The information about these variations may result in more complex notations as the complexity of the signs increases.
Figure 2.4 shows a signer performing the gesture of the sign for "nemici" ,"enemies" in English, while the Radutzky Id is used to identify the sign. It can be seen from Figure 2.4 that the signing person is pointing two hands towards both shoulders which is the initial position of the sign. The arrow keys pointing downwards indicate the direction of the subsequent movements. There is another hand shape beneath, where the arrows are actually pointing to the final position of the hands. This means that to perform the sign, hands are moved from the shoulders to the chest with three fingers and the thumb closed and the index finger pointing inwards. In the RN, "G" shows the configuration of the hands. The symbol "T>" and "T<" describe the orientation of the hands in order to perform the sign. The symbol "∅" provides information about where the specific sign should be performed. Finally, the symbols in the top right position of the notation describe the movements of the sign.

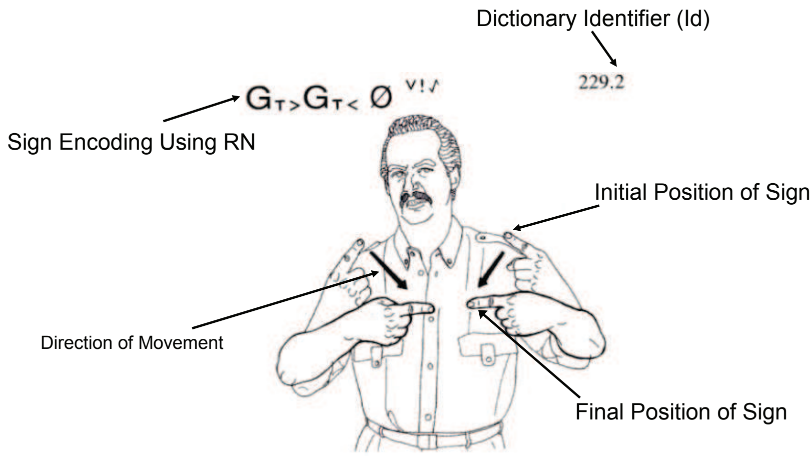


Figure 2.4. Radutzky dictionary sign for: “nemici”, “enemies” [2]

2.3.2 The ATLAS Project

The Automatic Translation into sign Languages (ATLAS) project was a three year project and its main purpose was to develop tools which enhance the deaf people’s access to IT content and services [68]. It provides a system for the automatic translation from Italian written text to LIS. The LIS signs performed by a virtual character are shown as output [69]. The project includes various software modules for linguistic translation for virtual character animation. In ATLAS the main focus was on a specific domain (i.e., weather forecast). The translation is performed resorting to rule based and statistical translators, the details of translation process are discussed in chapter 4.
In Atlas, a mixed approach was adopted for the generation of signs, using an animation language to compose and parameterize pre-captured and hand animated signs. The interpretation process is performed in two steps. First, the text is translated into an intermediate representation called “ATLAS extended written LIS (AEWLIS)” then these sentences are translated into characters gestures Animation Language (AL) that is generated within the Virtual Character Animator and represents the LIS translation of the sentence itself. A LIS sentence is basically a sequence of annotated glosses carrying a set of additional syntactic information. Figure 2.5 shows the architecture of the ATLAS system. The semantic-syntactic interpreter resorts to an ontology modeling the weather forecasting domain. The interpreter performs the analysis of the syntactic tree and builds a logical formula by looking at the semantic role of the verbal dependents. This information is collected within the

AWLIS formalism, which is sent to the animation module. This module is composed by:

1. A Planner: it gets the signs and sets them into the signing space, according to relocation strategies.
2. An Animation Interpreter (AI): it takes the AWLIS and the sign planning to render the signs by smoothing them together.

The animation engine retrieves a basic form of the sign for each lexical entry specified in the AWLIS from a database. Once the basic shape is done, it applies a set of modifiers to the shape, e.g., facial expression, automatic relocation, body movements. The Signary is the ATLAS lexical database that collects all the basic forms of the signs, performed by the avatar, in a video format.

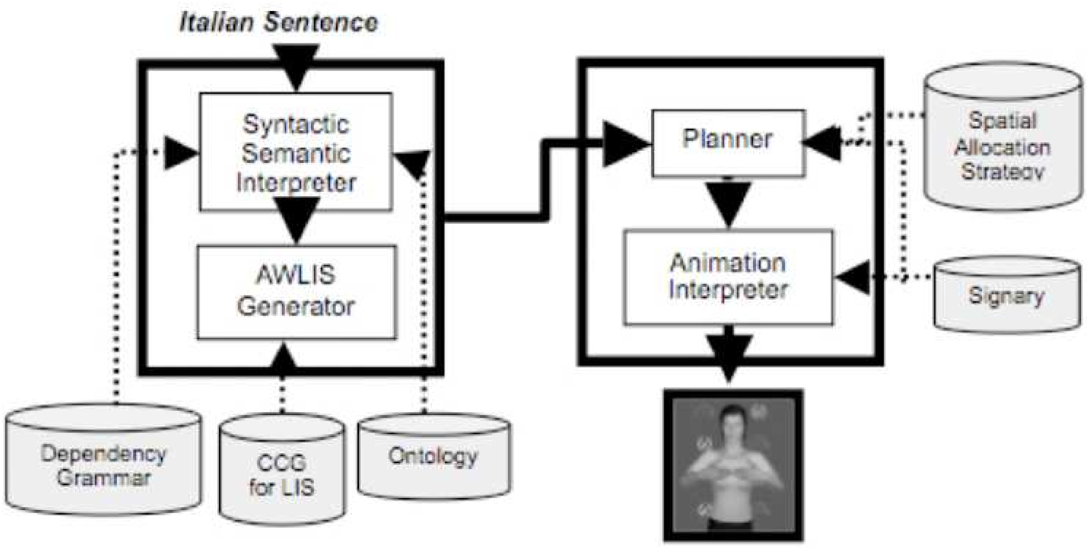


Figure 2.5. Architecture of the ATLAS system

2.3.3 The ATLAS Extended Written LIS (AEWLIS)

It is important to point out that AEWLIS is not a written form of sign language but an intermediate form of the sentence that supports the ATLAS processes and helps us to store and exchange data between the ATLAS modules. AEWLIS is machine-readable and conveys information about all the sign peculiarities. It contains phonological, semantic and syntactic information. In the AEWLIS mechanism, all the typical sign parameters such as configuration, orientation, place and sequences of movements are not considered. Only the following Communication Channels (CCs) are considered to be relevant: hands, direction, body, shoulders, head, facial expression, labial position and gaze. An AEWLIS unit can be considered as a sentence or sequence of Italian words. Each of them is a lexical unit. For each lexical unit, a description is given in terms of elements organized as concurrent tracks [6]. For each lexical sign, a single track expresses additional information with respect to the lexical unit. The following four tracks are used:

1. Lemma: used to store attributes related to the lemmas, including the lemma id, the identifier and its syntactic number.

2. Sentence Structure: used to store information items related to the structure of the sentence, including the proper pointer to the related syntax tree, the sentence syntactic and semantic roles, and parts of speech.

3. Modifiers: used to express information about the relocation of the sign and the modifications with respect to the lexical sign.

4. Time stamps: used to encode information related to the sign performance timings.

The AEWLIS is used in three situations i.e. it is produced as output in translation process from Italian text to AEWLIS, as the output of an annotation step from LIS video to AEWLIS and in modification and correction phase of AEWLIS (Annotation details are discussed in section 2.5.2).
The ATLAS services have the focus on lexical resources and on a LIS corpus. These resources are used in order to train the translator and also as the basis for the development of the online LIS dictionary.

2.4 Development of Platform Independent LIS Dictionary

The development of a new LIS dictionary started to overcome the lack of LIS resources especially in electronic format. The purpose of the platform-independent LIS Dictionary is to facilitate deaf users on each platform by providing the same facility with the same interfacing thus developing a platform-independent application. Users should not be restricted to a single platform in the sense of operating systems or hardware. There should be equal opportunity for everyone, everywhere. The Platform Independent LIS Dictionary (henceforth referred to as Dictionary) is proposed to support deaf learning of both Sign Language and written language. The Dictionary is linked with MultiWordNet (MWN), a lexical and semantic database

which includes several languages (MultiWordNet is discussed in detail in chapter 3). The Dictionary provides the full set of lexemes and video data sets of Italian sign language with an online interface attached to the MWN. Each sign in the LIS dictionary is explained with pictorial representation providing the body position, orientation and movements. The dictionary provides the correspondent lemma and an example of its usage in the discourse.
Before going into details of the developmental process of the dictionary, the following section covers some details about previous work related to SL translation and development of SL dictionaries.

2.4.1 State of the Art

There is considerable work in progress on Sign Language dictionaries, but many of them have some limitations and provide only limited functionalities, i.e. only for a word-to-sign search, and only a limited lexicon data set is available, usually pre-recorded with the help of a signer [70, 71]. A large lexicon project was started for American Sign Language, which facilitated a large lexicon data set but the purpose was to find similar signs in database by taking new video signs [72].
As shown by Al-Ohali [73], more than fifteen tools (e.g. 3D animation, video based representation, sign editor, dictionary, text analysis and speech recognition) have been developed to assist deaf in their lives and increase their learning opportunities. Virtual Characters (or Avatars) are obvious candidates for the representation of signs. Many avatar-based signing systems have been developed for automatic interpretation of text into sign language [69, 74]. The VISICAST translator and eSIGN projects [75] defined the architecture for sign language processing and the gesture visualization with a virtual character by using motion capture and synthetic generation to create signs.
Vcom3D developed and evaluated a proof of concept American Sign Language and Signed English dictionary for handheld computer and media player devices [76]. The dictionary included signed animations of English terms that could be difficult for Deaf learners to understand, including idioms, scientific terms, and words with multiple senses. Mobile ASL is a project of Washington University on Sign Language communication over cell phones in which they are checking limitations, especially low bandwidth constraints of the current cell phone networks, which created many video compression challenges because even today's best video encoders cannot yield intelligible ASL at such low bit rates [77]. These SL avatar animations may be scripted by the user [78] or can be generated automatically by text-to-SL machine translation systems [79, 80, 81, 82].
The statistical machine translation of text to sign language needs a suitable corpus of SL data to train with. Unfortunately existing corpora are either too small or too

general, so they are not suitable for specific domain. The corpora generation mechanism needs to be given importance because it influences both the quality of annotations and the related information. BlueSign (Blue sign partners) and ZARDOZ (an English to SL translation System) are some examples of working systems that take into consideration statistical or example based techniques to perform translation [82, 83]. Blue Sign doesn't take into consideration the grammar for LIS translation. Recent researches on corpora creation emphasized on the representation, proper documentation and machine readability [84]. The DGS Corpus project has started in January 2009 and will be completed in 2023, in which a corpus based electronic dictionary for German Sign Language will be produced [85]. The DGS project corpus is based on 328 informants and 350 to 400 hours of footage in which 2.25 million tokens will be involved. Almost 6000 sign entries including information on form, meaning, grammar, variants and usage are planned along bidirectional search (through sign form or written word) in final product. Swiss-German Sign Language dictionary is based on Multimedia Bilingual Databank that collects 60h of informal group discussions on given arguments (medicine, sport, schooling, etc.) [86]. Almost 3000 signs were collected and stored in multimedia bilingual DSGS-German database. The research aimed at producing electronic and print dictionaries for Swiss deaf people. The British sign language Corpus Project aims at creating a machine-readable digital corpus of spontaneous and elicited British sign language (BSL) collected from deaf native signers and early learners across the United Kingdom. Although the recruitment of participants is being balanced for gender and age, it focused only on signers exposed to BSL before the age of 7 years, and adult deaf native signers are disproportionately represented [87]. The Corpus NGT is another effort to record and archive video data from Sign Language of the Netherlands [88]. The NGT Corpus for Dutch Sign Language has collection of 2375 videos (as per September 2010) for about 72h of data on dialogues and narrations signed by 92 native signers. The corpus is fully transcribed at gloss level. The users of the dictionary face the problem of extracting information on meanings and grammatical characteristics about the signs from the glosses. They also have the difficulty in finding information from (Dutch) example sentences, the grammar overview and from the word list. Instead, this information should have been provided in a clear way in the entries and according to grammatical categories.

Specific groups of people demanded dictionaries containing the signs for the terminologies they usually use, resulting in printed dictionaries with a focus on different domains. Spoken languages are basically one-dimensional in nature (sounds of the words are pronounced one after another). Some spoken language dictionaries also contain information about lemma pronunciation using different symbols. The problem for sign notation is that signs are represented using three-dimensional space and have sequential (chronological) structure. This is difficult to encode in one- or two-dimensional paper representations. The compilers of SL resources have adopted different approaches to tackle this issue. Sign notation systems are used in some dictionaries, but generally these are complementary to an image of the sign. Most printed lexical resources use line drawings or photographs to represent a sign as shown in Figure 2.4, taken from Radutzky dictionary [2]. Sign representation on the paper requires extra space and it is not easy to provide the detailed information. Most paper lexical resources of SL existing to date contain limited entries as compared to the spoken languages.

2.4.2 Business Use Cases LIS Dictionary

Use cases represent the behavior of a system, scenarios that the system goes through in response to stimuli from an actor. During business analysis of a system, one Use Case model for the system can be developed and can be further extended with packages to represent the various business domains of the system. Each package can be decomposed with a Use Case diagram that describes the process in details along actor interactions. Each scenario shows a different sequence of interactions between actors and the system. All following UML use case diagrams describe LIS dictionary functionality from the deaf user's perspective and some management tasks which are supposed to be performed by the administrator. Details about sign creation and implementation details will be discussed in subsequent sections.

2.4.2.1 Use Case 1: Searching Word From LIS Dictionary

The user searches a word from the dictionary and corresponding meanings in the MWN and LIS translation are displayed. As a precondition, the user must have started the application. The system is displaying the home page. When the user starts the application, the list of words/lemmas existing in the MWN are loaded. When the user starts typing, the list starts filtering the word, the user can select the word from the filtered drop down list or type the required word. The results for the requested word are displayed to the user. Alternatively, if the user searches a word which doesn't exist in the MWN database, an exception is displayed. As a post condition, the results are displayed correctly and accurately according to the user query. Figure 2.6 shows the use case diagram of searching lemmas in LIS Dictionary.

2.4.2.2 Use Case 2: Display videos of the Signs

The user can see the video of the Sign animation by clicking the play button in case of web application, while in mobile application, it is played automatically. The user can stop the video, play and repeat the video animation of the sign. Users can also set the speed of the video. If the word has corresponding meanings in the

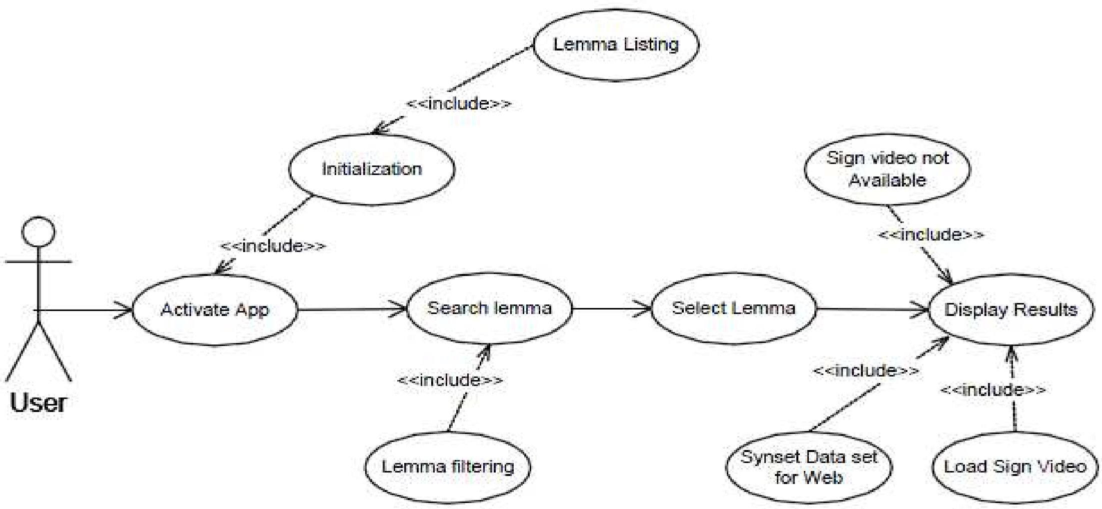


Figure 2.6. Use case: Searching words in LIS dictionary

sign language then the video is displayed. If any of the related words or synonyms exist in the LIS dictionary that word will be highlighted. Users can explore the hyperlinked word in the sign language. Alternatively, if the video animation of the corresponding word is not found in signary, an exception is displayed. User can see the video of the synonyms highlighted to get an idea of the word. Figure 2.7 shows the use case diagram about displaying videos for selected signs.

2.4.2.3 Use Case 3: Purpose Changes in LIS Dictionary

The users can propose some modifications or additions to the system. The user can suggest changes to the word or its meanings (Synset) which are displayed on the user screen. If the users suggests the addition of a new word then he also adds the corresponding meanings. The user can also propose new animation of a sign or change in the existing signary. The system validates the user credentials in order to allow suggestions for the optimization. If the user is not registered or validated then access to the system will be denied. The users propose modifications or additions which are managed by the administrator. The administrator logs into account to see the changes in the system. The administrator can reject or accept the advised changes and delete or add new users of the system. The administrator sees the modification in the meanings of the word (Lemma synset), checks its validity and originality, then accepts change or removes it. If the users requested to change or reset his password, administrator handles the request. Figure 2.8 shows the use case diagram about changing user credentials or proposing changes to the LIS Dictionary.

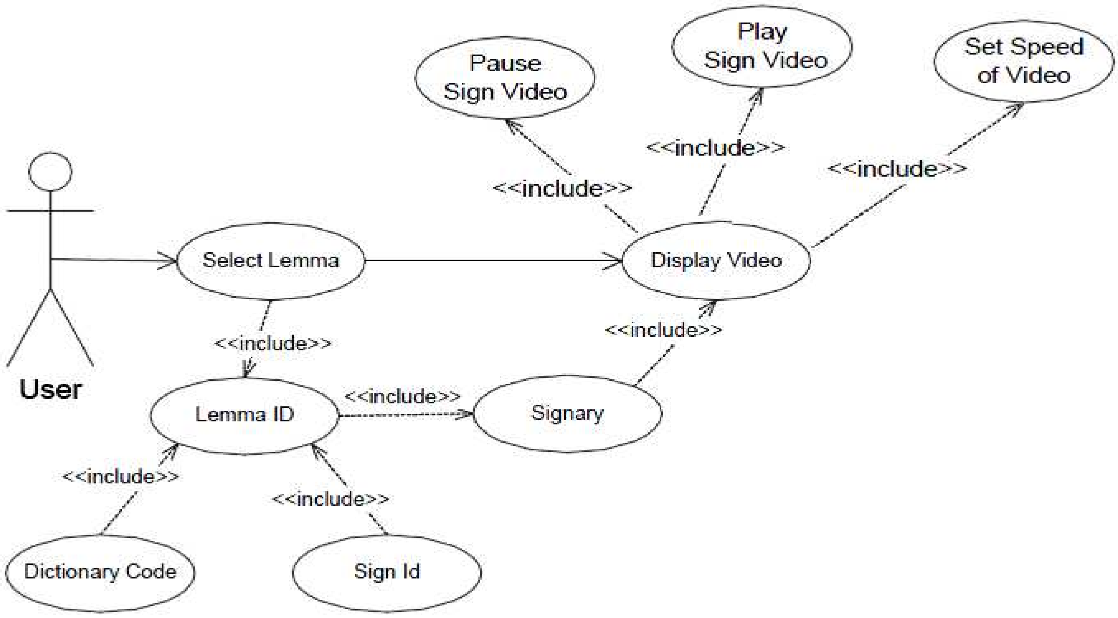


Figure 2.7. Use case: Displaying videos of the selected signs

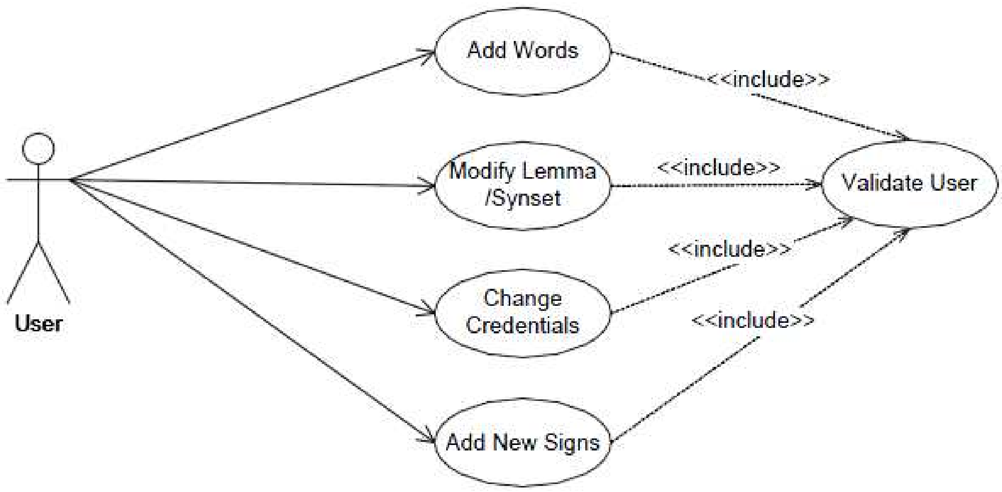


Figure 2.8. Use case: User purposes changes in LIS dictionary

2.4.2.4 Use Case 4: Administration of LIS Dictionary

This use case describes the Administrator interaction with the system. The Administrator mainly controls and manages the complete system. He handles all the queries regarding system maintenance, optimization and updates it. All the changes suggested by authorized users must be approved by the administrator of the system.

When the user starts the application, it is loaded and the login screen is displayed. After the administrator logs in he has the option either to search the word or manage modifications. If the administrator selects the modification management option then the user requests are shown. These requests could be either for the password recovery or suggestions for additions/modifications to optimize the system. If user selects the search word option then the home page is shown. Where he can either select the word from the drop down menu, or type in the box to search for the meaning. Use case described in figure 2.9 highlights the functionalities the system administrator.

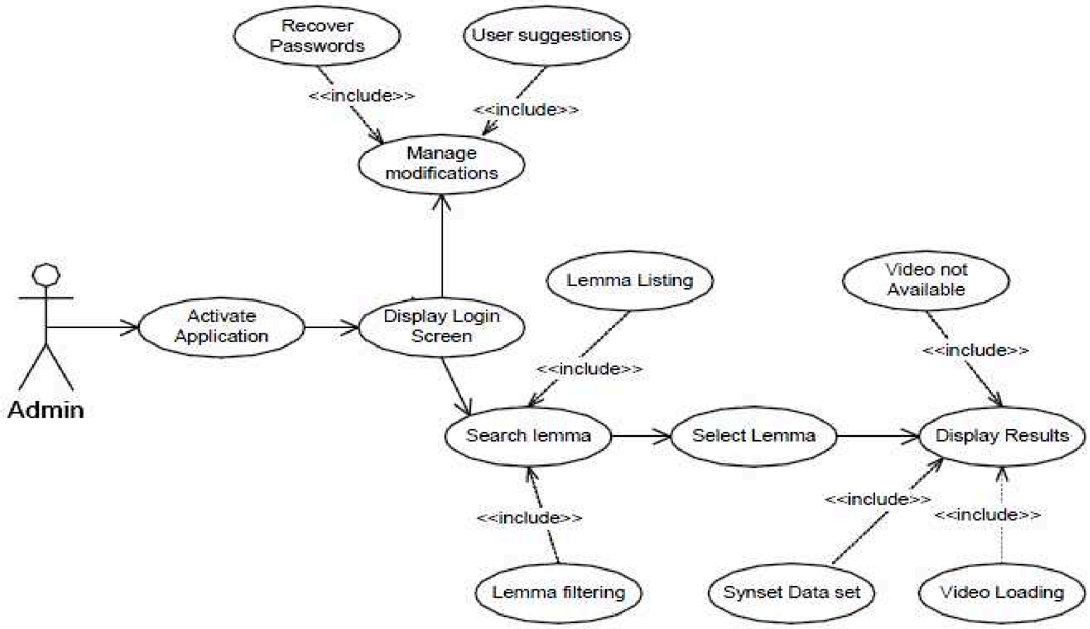


Figure 2.9. Use case: Administration of LIS dictionary

2.5 Lexicon Creation Process

The lexical entries were generated on the basis of data extracted from the Radutzky paper dictionary. Lexical analysis has been performed on these signs to identify the signs not available in our database, or those that are not in use and are required to change accordingly. These newly created or modified signs were termed as new signs. Figure 2.10 shows the whole lexicon creation process that is divided into different phases.

About twenty seven hundred video animations were created using standard Radutzky dictionary. As the ATLAS project especially targeted the weather forecast domain, the lexicon was composed of two sets of signs: general-purpose and weather forecast related. The data is manually extracted from Radutzky dictionary by means of a user interface (Figure 2.11) and stored in ATLAS Multimedia Archive(AMMA). The interface allows to store lexical information along with phonological data. The information stored includes:

- Lemma
- Radutzky ID
- Radutzky Notation

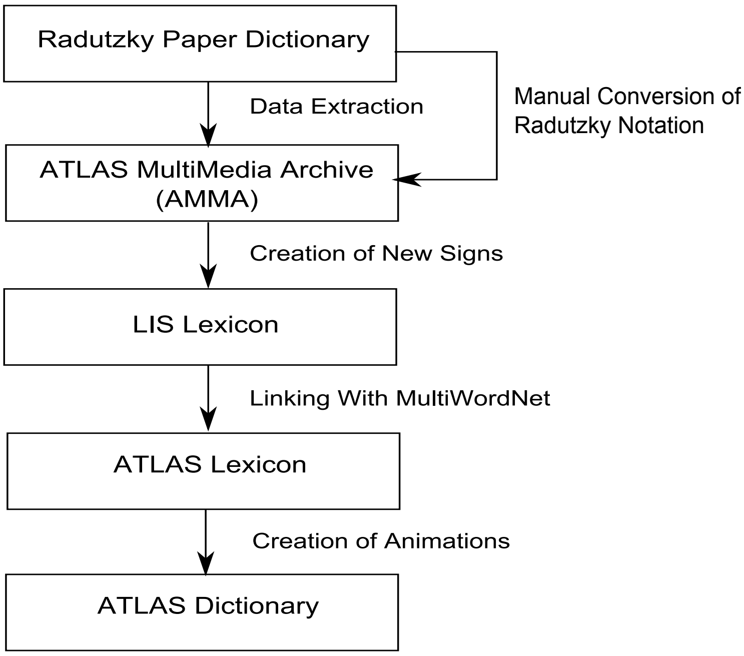


Figure 2.10. The lexicon creation process

The information about the phonology is manually extracted and signs were classified in following categories.

1. Hands involved: whether the specific sign is performed using one or two hands.
2. Single or compound: A compound sign is formed by two or more lexical units. The same as in spoken English where a compound word is made by two distinct words to express one meaning (i.e. "Blackbird").
3. Static or dynamic: A sign is static if the movement is not present in the RN, otherwise is dynamic.

4. Symmetric: A sign is symmetric if it is symmetric with respect to configuration, orientation and movement.

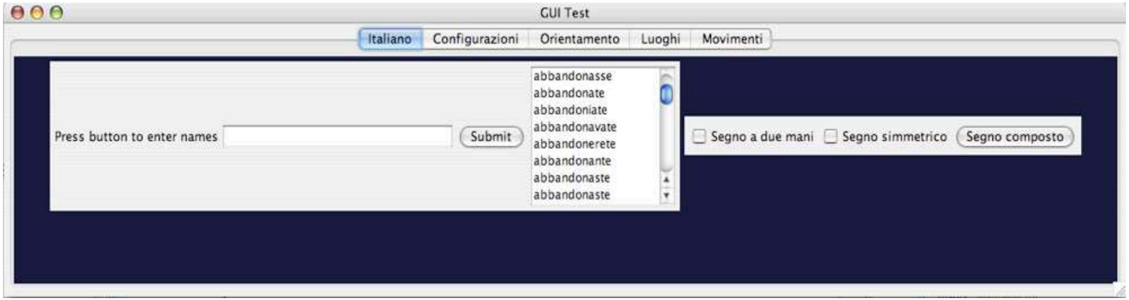


Figure 2.11. Interface for the lemma and sign category data entry

2.5.1 Creation of New Signs

The creation of the new signs was started from the video data analysis of the original weather forecast news bulletin broadcasted by RAI (Radio Televisione Italiana). The set of 55 bulletins, containing about 15K Italian words corresponding to about 1.5 hours of Italian audio/video were chosen randomly, uniformly distributed along the whole year. The need of creating new signs is felt by the consistent lack of signs related to weather forecast domain in the common LIS lexicon. For example the sign for "temperature minima" (minimum temperature) was newly created as it did not exist in LIS. A group of annotators including people from the deaf community agreed upon these newly created signs.

The new signs can also be detected during annotation and analysis of source text using lexical frequency analysis. Table 2.1 shows some statistics of the signs of Italian words (a subset) and their frequency of occurrence in weather forecast bulletin selected within the ATLAS project. These are divided into two categories, the dictionary lexemes (also available in the Radutzky dictionary) and non-dictionary lexemes (not available in the Radutzky dictionary). For each word, its frequency and English translation are shown. In the bottom line the total number of words taken for the lexical frequency analysis are given. An initial study of lexical frequency in the collected corpus provided information about much needed data for researchers and language teachers. This data can be used to design experiments about LIS processing and to teach languages. It has also helped to further enhance our understanding of the distribution of grammatical sign categories (and unique sign types) within LIS, across signed and spoken languages.

A work team consisting of three deaf LIS signers, LIS experts and an interpreter was set up in order to create new or modify existing signs. The procedure for the

Table 2.1. Lexical frequency of dictionary and non dictionary words [1]

Dictionary words			Non Dictionary words		
Lemma	Translation	Freq	Lemma	Translation	Freq
Che	That/what	219	Essere	To be	187
Regione	Region	182	Settentrionale	Northern	88
Giornata	Day	146	Precipitazione	Precipitation	82
Ancora	Again	123	Muovere	To move	79
Vedere	To see	120	Nuvola	Cloud	76
Nord	North	113	Nuvolosità	Cloudiness	72
Più	More	111	Zona	Zone	72
Tutto	All/everything	108	Pre	Pre- (prefix)	69
Domani	Tomorrow	104	Meridionale	Southern	67
Qualche	Some	104	Occidentale	Western	59
Questo	This	100	Ci	Us/ourselves	54
Nostro	Our	95	Area	Area	53
Poco	A little	79	Adriatico	Adriatic	47
Nuvoloso	Foggy	77	Settore	Sector	46
Temperatura	Temperature	77	Aumento	Increasing	44
		Total:4040			Total:4236

creation of the signs was followed by a qualitative and quantitative analysis. This was performed to verify the comprehension of these signs by deaf individuals. The new signs were verified by setting up a sign-image pairing test. These were also compared with the human performed signs and signs reproduced by the virtual avatar. The following procedure was followed in order to create the new signs:

1. A transcript of the video forecast news in Italian spoken language is created by means of a speech recognizer.

2. The produced data is checked for errors and an edited version is produced.

3. The edited versions are analyzed to identify the set of signs that are not present in the LIS lexicon. For new signs we mean:

   (a) Completely new signs, typically related to meteorological phenomena (113 signs).

   (b) Signs widely used by the deaf community, but not present in our reference Italian-LIS dictionary and needed for the pure LIS translation (214 signs).

   (c) New signs, present in the dictionary but considered obsolete by the work team (73 signs).

4. For each new sign we identify an appropriate lemma and define a brief description if needed.

5. We define the complete list of new signs and we fill the Lexicon database. The Lexicon is filled by:

   (a) Signs present in the standard Italian-LIS dictionary

   (b) New signs defined as described in the steps from 1 to 4.

6. For each of them:

   (a) We give a translation in written LIS with the following criteria:

      i. We can use just the signs present in the Lexicon (step 5)
      ii. We try to give a simple and linear translation
      iii. We try to use the "standard" signs (namely, from the dictionary) as much as possible
      iv. In case of synonymy we use the one that is most widely used according to the work team's knowledge.
      v. If we translate a concept using a sequence of signs we must guarantee that it conveys complete sense of that concept.

The ATLAS LIS dictionary has more than three thousand signs as set of Avatar animations in which 400 signs are newly created. Each sign is linked to corresponding Italian lemma. The new signs along with their meanings and animation videos are stored in the Atlas MultiMedia Archive (AMMA). Figure 2.12 summarizes the steps of new sign creation process.

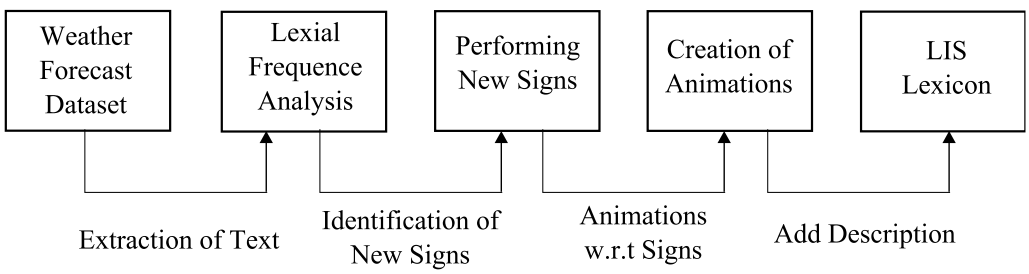


Figure 2.12. Creation of new signs

2.5.2 The Annotation Process to Support the Lexicon Creation

Several researchers described that annotated corpora constitute a very useful tool for testing language hypotheses [89, 90]. Different issues were faced during the annotation task due to the fact that unlike spoken languages annotation, SLs annotation involves a meta-linguistics task in order to grasp the multidimensional aspects of SLs [91]. In order to produce an accurate annotation, the relevant information must be chosen. In particular, it is required to detect which manual and non-manual features should be considered for the implementation of an automatic Italian text to SL translation system and which information is not essential.
The process of annotation includes the use of computer tools to load the video dataset of the signing, and helps the human annotator with several facilities, making the annotation process easier. A custom tool called ATLAS Editor for Annotation (ALEA) has been developed for this purpose. The Sentence Annotation Module (SAM) of ALEA was used to load the video information [92] and the information was annotated by using a time-line. Users can perform the following operations with ALEA.

1. Sentence modification (lemma additions and deletions, order changes).

2. Lemma tagging (association with attributes proper of the AEWLIS specification and format).

3. Linking to lexicon resources (linking each synset to a sign in the LIS dictionary).

The annotation is performed by modifying the sentence and tagging the lemmas. As a consequence the annotation is performed sentence-wise, defined as a concatenation of expressions of the signs. The input text is analyzed using a morphological analyzer [93] and the relevant lemmas are properly segmented. The useful morphological information (i.e. number and syntactic role) is automatically detected and annotated from lemmas using the morphological analyzer. ALEA allows the annotator to manually modify the automatically generated information in order to match with the specific LIS feature.
Lexical signs were required to be uniquely identified using glosses, in line with [23]. In our LIS lexicon, we refer to the gloss uniquely identifying a sign using an id (sign id in the following). These ids are assigned as a numeric value that is used to label a sign regardless of the alphabetic order or its semantics. Unique identification of each sign was an obvious motivation for the assignment of ids to the glosses in the database. For example we have more than one sign for the word “casa” which means house. Different ids were assigned to different signs of the same Italian lemma

"casa". The id is neither a translation nor a necessary indicative of the grammatical class of the sign (e.g. noun, verb, adverb, etc.). The ALEA tool is used in order to integrate signs and lemmas. This allows linking each lemma with the corresponding sign movie in the lexical database and also provides the preview of each sign [92].
The other factor considered is sign spatial location, which specifies whether the sign is performed in another place within the signing space compared to the citation form. In this case, each lemma is signed in the neutral space. Locations have two equivalent alternatives for the expression: the actual space location in terms of the placement with respect to the so called neutral space and the spatial location occupied by another sign. A given LIS sign form can usually be used with or without modifications in more than one type of syntactic slot (and hence grammatical class). One has to look at the assigned grammatical class of the sign in a particular context to check its grammatical category because the sign id does not provide the information about the grammatical class. Another important factor to be considered is the temporal unit as reference of our annotation. We have chosen to use the lemma as the temporal unit. Therefore it is not possible to further segment the lemma in order to annotate, for instance, two different locations. The actual location of the sign must be uniquely expressed as a single piece of information within the temporal slot. Even though this approach could seem risky and even less flexible than other approaches, which could segment the annotation using a more granular temporal segmentation. It was verified that this provides enough information to enable an intelligible signing by the virtual character.

2.5.3 The Sign Visualization System

The module used for the sign generation is a system that generates real-time animations, displays them in a player and exports them as picture frames. It takes as input the symbolic form of LIS called AEWLIS, a sequence of lemmata and a description of the meaning, syntactic number and other semantic, pragmatic and articulatory information. The signing space in front of the interpreter is structured with locations where the items mentioned in the sentence are positioned and kept for further references in the sentence. Signs are described by a default animation (stored in a repository, the "signary"), which has to be parametrized with respect to the context given by the specific sentence. When the player receives a sentence in AEWLIS form, it plays the signs in sequence adding manual and non-manual components with well defined priority strategies [37].
The process for the generation of avatar based signs in the LIS Dictionary is defined as follows (See figure 2.13):

1. A LIS interpreter is recorded signing the lemma of each LIS sign.

2. A raw animation file is produced through the key frame animation technique.

3. The raw animation file is played within the ATLAS player in order to produce virtual character based motion frames.

4. The frames are integrated together to form movies by means of an automatic procedure. Figure 2.13, shows the process of signing a lemma.

Italian words given as input to the system are linked to the corresponding sign movie in a database. The signs are rendered through the ATLAS Player developed at Virtual Reality and Multi Media Park which is able to perform several functionalities such as the blending of signs in one sentence and the application of procedural transformations on the sign to manage relocation [69].

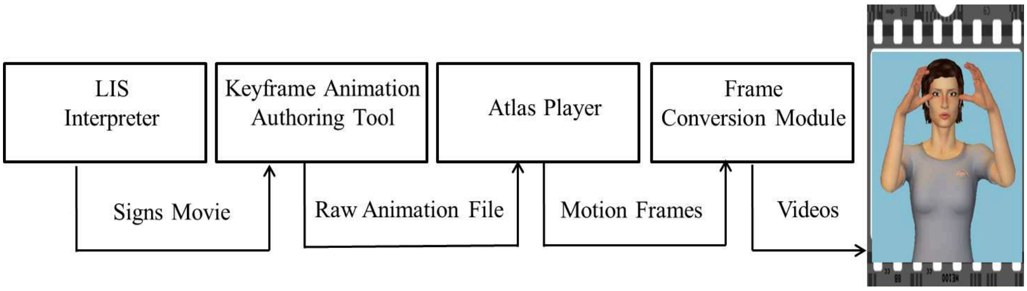


Figure 2.13. The process for the generation of avatar based signs

2.6 Abstract Architecture of LIS Dictionary

The LIS dictionary provides the integration with the MWN database (The integration details are discussed in Chapter 3). The integration with MWN allows a rich extension with the meanings and senses of the words existing in MultiWordNet. The dictionary allows users to acquire information about lemmas, synonyms and synsets in the Sign Language (SL). The LIS Dictionary is implemented in such way that its functionality can be enhanced according to the availability of resources. For offline users lexicon data set along video signs is stored locally while for online users the dictionary is mapped to MultiWordNet synsets in client server fashion. Figure 2.14 shows an abstract architecture of the Platform Independent LIS dictionary [37].

The LIS video animation is displayed against the selected lemma and if the video of the desired lemma doesn't exist, the user can see any word from the synset existing in dictionary to understand the meaning. Availability of signary (the LIS database) locally maximizes the performance as no streaming is required to display results. Browsing through synonyms helps to enhance the understanding of the sign

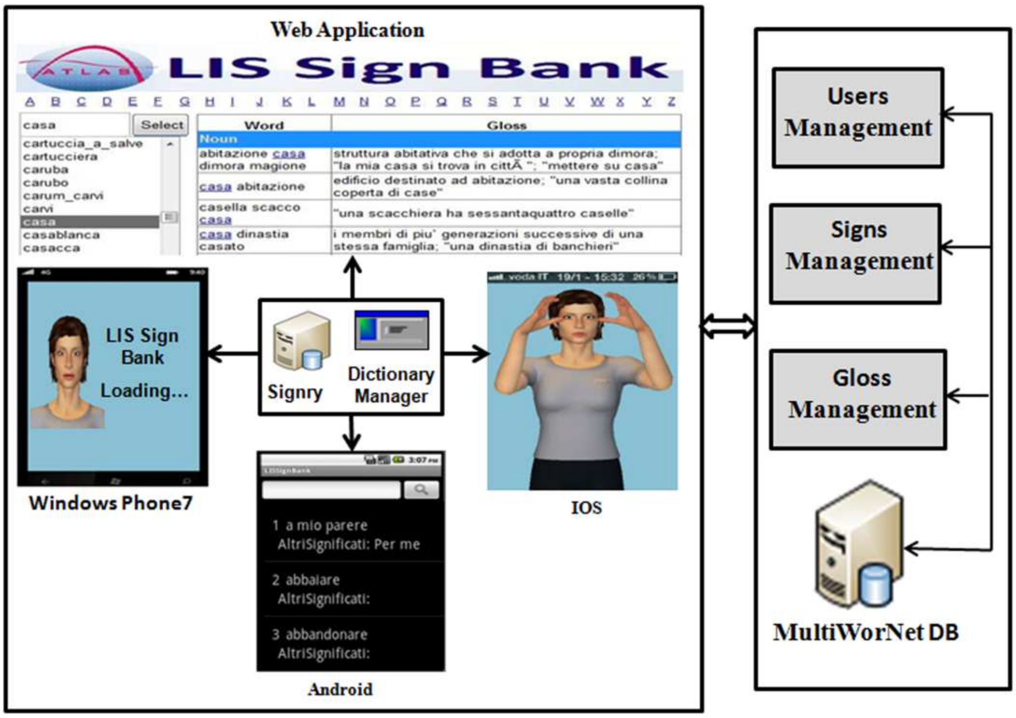


Figure 2.14. Abstract architecture of platform independent Italian Sign Language dictionary

with different lemmas.
The system developed can be updated, extended and optimized. To continuously extend the system with new signs and animations, sign Management is performed by the system administrator on server side. End users can suggest new signs by submitting lexemes and corresponding videos or can propose changes or modifications to an existing sign. The system administrator will check the new lexeme suggested and will verify its existence in the database. After verification of the originality and duplication, the system administrator will add gloss and will generate animations against the suggested lexeme and will make them part of the system.

2.6.1 Information Flow in LIS Dictionary

The users will have the facility to use a web application or a mobile version for the LIS Dictionary. The functionality of the LIS dictionary is divided into three major components: the keyword manager is responsible for handling requested lemmas and displays filtered matching words in a drop down list. The Query Processor is responsible for handling incoming queries and checks whether word filtering or sign display is requested. The video animations are matched with the lemmas by a lemma movie matcher and are assigned a unique id. The Id manager keeps track of the stored video signs along with the Id. User requests are taken by the query interface and results are shown on user interface. The information flow in LIS dictionary is shown in the Figure 2.15 [39].

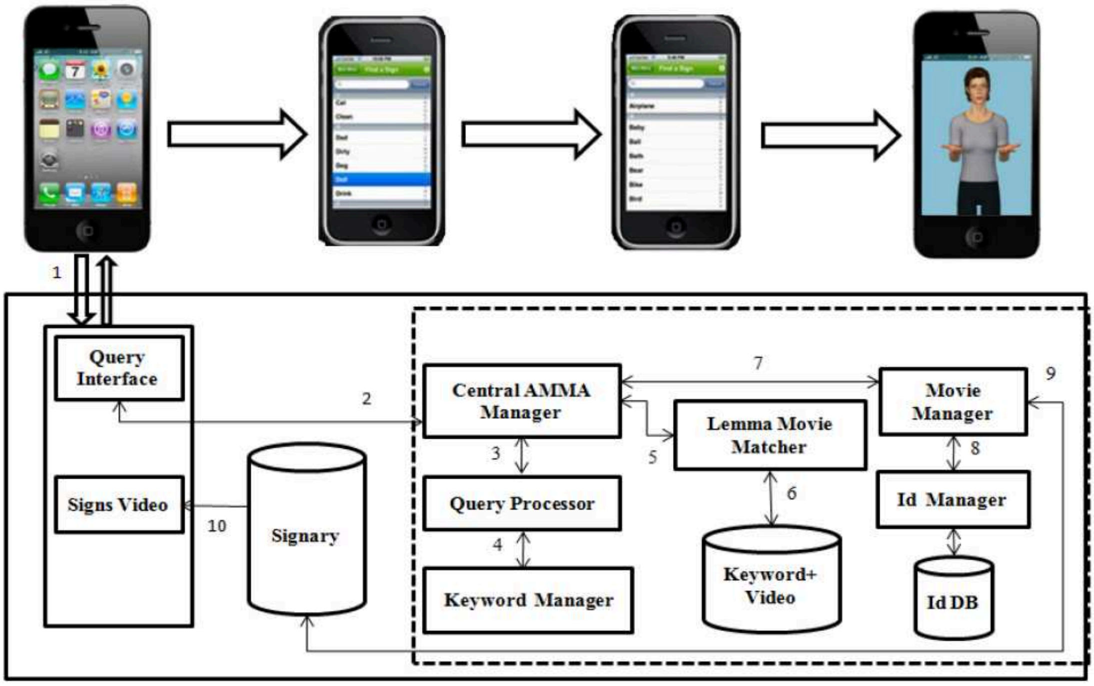


Figure 2.15. Interactive information flow in LIS dictionary

The processing sequence is shown as a sequence of steps. The request received from the interface contains the parameters to request for specific information along with the authentication credentials of the user. Steps 1, 2 and 3 show that the interface is linked with the AMMA database and the Query Processor. AMMA includes the Signary, the Keyword + Video DB and the Id DB. The Query Processor in the central manager checks whether user has started entering the word or it's the request to find the meaning the sign language. 4) If a letter is entered control is passed to the Keyword Manager, which shows the filtered result in the drop down menu. 5) When the user requests/selects the word, the Query Processor sends it to the Lemma Movie Matcher. 6) The Movie Matcher searches for the corresponding lexical video from the database containing the couple keyword-video and, if it is found, sends back its Id to the AMMA Manager. 7) Then the request is sent to the Movie Manager to fetch corresponding file path against the intended Id. 8) The Movie Manager takes the path of the corresponding video from the Ids DB along

with the Id. 9) The Movie Manager forwards the path containing the animated video to the signary. 10) The video stream of the sign is finally displayed on the user interface. The graphical user interface facilitates users with the icons to play, pause, resume, and replay the video. The user can watch again or pause at a specific point to better understand the sign.
Platform independent LIS Dictionary helps in learning new words and is capable to run on several devices. The LIS dictionary contains a large video data set and its structure is flexible enough to include further signs in the future. The next chapter will provide details about integration of the LIS Dictionary with the MultiWordNet knowledgebase.

Chapter 3

LIS MultiWordNet

A novel Italian Sign Language MultiWordNet (LMWN), which integrates the MultiWordNet (MWN) lexical database with the Italian Sign Language (LIS), is presented in this chapter. The approach relies on LIS lexical resources which support and help to search for Italian lemmas in the database and display corresponding LIS signs. MWN is a lexical database that handles the existence of relation among words defining the synset: it models the sense of a single headword. Thus, words sharing the same synset can be defined as synonyms. Including the MWN platform as part of the translation process requires to store the sysnet data to the lexicon, too. The integration with MWN allows a rich extension with the meanings and senses of the words existing in MWN.

The chapter also provides the description of the approach which links the Italian lemmas and LIS signs to extract and display bilingual information from the collected lexicon and the semantic relationships of LIS Signs with MWN. The users can view the meanings of almost one fourth of the lemmas of MWN in LIS.

The chapter is organized into four sections; Section 3.1 discusses the motivation behind creation of LMWN and describes the WordNet principles. Section 3.2 introduces MWN and discusses semantic relationships used to create MWN. In Section 3.3, the linking procedure of MWN with the LIS information and integration flow is described. Section 3.4 presents the results related to LMWN coverage and compares the syntactical relationships supported by MWN and LMWN.

3.1 Motivation for the creation of LIS MultiWordNet

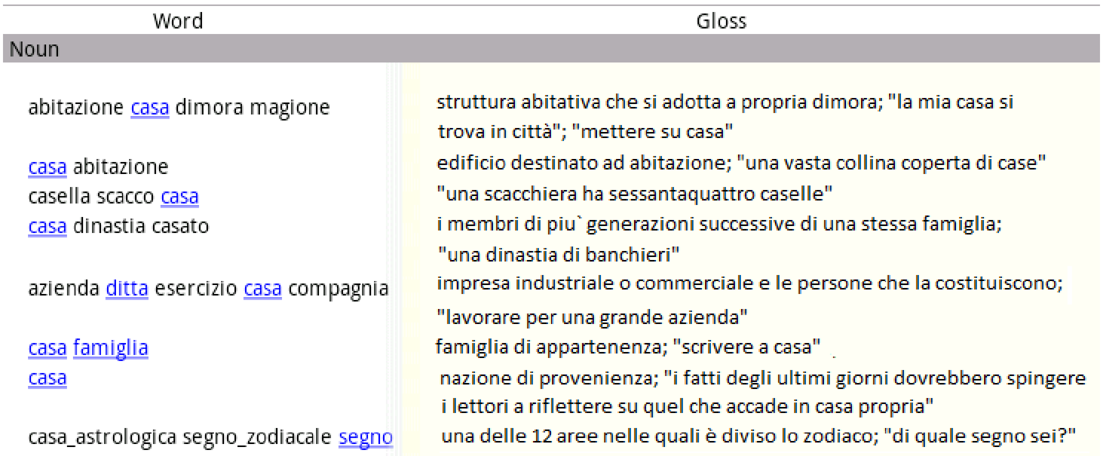
The motivation for the creation of LIS WordNet is to provide a comprehensive lexical resource that can be used as a tool for enhancing the learning of people having a hearing disability. Through MultiWordNet it is possible to resolve for each lemma

a better semantic representation while the association with the dictionary provides access to their basic sign representation, henceforth referred to as citation form. Each citation form is grouped in MultiWordNet synsets to manage synonyms. This grouping helps to check existence of the Italian source text in the LIS Dictionary.
WordNet is used for many natural language processing and computational linguistic tasks such as Word Sense Disambiguation, Word Similarity, Information Retrieval and Extraction and Machine Translation, etc. Initially WordNet was developed for English in Princeton University, and is currently the largest WordNet available called Princeton WordNet (PWN) [94] . Mapping and linking of knowledge networks has always been a problem of great interest. The mapping of the PWN to other knowledge networks has also received attention in the lexical knowledge network community. The coordinators of the Princeton WordNet and EuroWordNet started the Global WordNet in 2000 [95]. A multi-lingual linked structure of WordNets was the goal of the Global WordNet effort.
Multilingual SL knowledge bases address the development of language resources by capturing the relationship between words and their corresponding concepts in SL and hence making the semantic connection between lemmas of different languages. This sort of lexical information can be useful in different communicative situations and provide support for web pages, mobile applications and also television broadcasting [96, 97]. There are WordNets in more than 50 different spoken languages, but to our knowledge, there is hardly any that provides information about the corresponding SL.

3.1.1 The Lexical Matrix

The information in WordNet is usually maintained in a lexical matrix format, the row headings represent word meanings and the column headings stand for word forms. An entry in a cell of the matrix implies that the column can be used to describe the meaning of the row. The relationship in words and its forms is many to many [98]. The WordNet groups synonymous word senses into synsets, and shows them in structures of grammatical type hierarchies. The words in the definitions (or "glosses"), included in each synset are helpful in assessing word disambiguation.
For example, someone who understands Italian, can easily recognize "casa" as home (building used for living) and will be able to pick out the intended sense. In addition, the set of synonyms (abitazione, dimora, magione) will serve as unambiguous designators of these three meanings of "casa". These synonym sets (synsets) do not explain the meaning of the word, they merely group the similar words in an appropriate context. The gloss is used to describe the context in which the words represent similar sense. In another context "casa" is also used to represent the sign (the one of the 12 areas in which the zodiac is divided). For disambiguation between

home and sign, it is necessary to differentiate the sense of "casa" with the help of gloss. Figure 3.1 shows an example from our LIS Dictionary in which all the senses of "casa" are organized into lexical matrix.



Word	Gloss
Noun	
abitazione casa dimora magione	struttura abitativa che si adotta a propria dimora; "la mia casa si trova in città"; "mettere su casa"
casa abitazione	edificio destinato ad abitazione; "una vasta collina coperta di case"
casella scacco casa	"una scacchiera ha sessantaquattro caselle"
casa dinastia casato	i membri di piu` generazioni successive di una stessa famiglia; "una dinastia di banchieri"
azienda ditta esercizio casa compagnia	impresa industriale o commerciale e le persone che la costituiscono; "lavorare per una grande azienda"
casa famiglia	famiglia di appartenenza; "scrivere a casa"
casa	nazione di provenienza; "i fatti degli ultimi giorni dovrebbero spingere i lettori a riflettere su quel che accade in casa propria"
casa_astrologica segno_zodiacale segno	una delle 12 aree nelle quali è diviso lo zodiaco; "di quale segno sei?"

Figure 3.1. The word "casa" represented in form of lexical matrix in LIS dictionary.

3.1.2 WordNet Principles

WordNet defines different senses associated with or related to the meanings of a word and other well-defined lexical relations such as synonyms, antonyms, hypernyms, hyponyms, meronyms and holonyms. WordNets are composed of synsets which are created on principles of minimality, coverage and replaceability:

- Minimality: Only those lemmas will be part of sysnset which uniquely identify the same concept, e.g. Synsets of intelligent in Italian are intelligente, valente, abile, capace.

- Coverage: The synset should contain all the words representing a similar concept e.g. All possible synsets of smell in Italian language are afrore, ammorbamento, bestino, effluvio, fetidume, fetore, leppo, lezzo, miasma, olezzo, pestilenza, puzza, puzzo, puzzore, sito, tanfo, zaffata.

- Replaceability: In a specific context the two synonyms should be interchangeable and should not affect the overall meaning of the sentence if substituted with one or other. Table 3.1 shows some examples of replaceability where gloss is explaining the concept in which lemmas are being replaced with respect to synonymy.

Table 3.1. Examples of replaceability: lemmas are replaced with synonym words.

Lemma	Gloss
Casa [House]	La mia casa si trova in città [My house is located in the city]
Magione [mansion]	La mia magione si trova in città [My Mansion is located in the city]
Animo [Mood]	Tutto procede secondo il mio animo [Everything is going according to my mood]
Intento [Intent]	Tutto procede secondo il mio intento [Everything is going according to my intent]
Scopo [Purpose]	Tutto procede secondo il mio scopo [Everything is going according to my purpose]
Bagno [Bathroom]	Chiudere la porta del bagno, [Please close the bathroom door]
Toilette [Tiolet]	Chiudere la porta della toilette [Please close the toilet door]

3.2 MultiWordNet

The MultiWordNet(MWN) is a popular multilingual lexical database in which the words are connected through lexico-semantic relationships and organized into sets of synonymous meanings called "synsets", each representing one underlying concept. These words and concepts are linked through various lexical and semantic relationships [99]. MultiWordNet can be seen as a lexical matrix, having two dimensions: the lexical relations, between language specific words, and the conceptual relations among senses. Each synset expresses a distinct concept. In other words, synsets are linked by conceptual semantic and lexical relations.

There are different approaches for building a multilingual WordNet. The goal of the EuroWordNet project [100] is to build a multilingual lexical database of WordNets for several European languages i.e. Dutch, Italian, Spanish, German, French, Czech, English and Estonian. These WordNets are linked to an interlingual index. A set of common concepts has been defined to ensure compatibility between WordNets [100, 101]. In the framework of the BalkaNet project the WordNets for the Balkan Languages (Bulgarian, Czech, Greek, Romanian, Serbian and Turkish) were developed [102]. Afterwards, WordNets have been developed for numerous other languages.

The second approach is to build language specific WordNets aligned with the Princeton WordNet (PWN)[103] keeping the same semantic relations and principles adopted within PWN. This is achieved by importing semantic relations from the corresponding PWN synsets or whenever possible by establishing new synsets in correspondence with the PWN synsets. The MWN is structured in the same way as PWN around the notion of synsets, but many other semantic relations between the synsets were

identified and are extensively or partially encoded. Nouns, verbs, adjectives, and adverbs represent one underlying lexical concept and are interlinked with a variety of relations [104].
Among these relations between words referring to similar concepts and belonging to the same semantic order, each synset has an associated definition called gloss. A gloss is a short explanation of the meaning of the concept represented by the synset. The advantage of MWN over EuroWordNet is that it is easy to define relationship between PWN and newly derived WordNets due to similar semantic relationship [99]. The multilingual hierarchy implemented can represent true lexical idiosyncrasies between languages, such as lexical gaps and denotation differences. We are currently working with the MWN in order to extend it with the LIS lexicon and eventually produce the LMWN.

3.2.1 Semantic Relationships in MultiWordNet

Some semantic relationships used to create MWN are discussed as follows:

1. Synonym: The most important relationship in every WordNet is the synonym (similarity in meaning). Two expressions are synonymous, if one is changed with the other, but this does not affect the truth value of the sentence. Such expressions in linguistics are very rare. In weekend definition two expressions can replace each other in specific context [98]. An example is discussed in Table 3.1.
   For substitution of synonyms, the grammatical categorization of lemmas into nouns, verb, adjectives and adverbs is important. The nouns can serve as the subject or object of a verb while the verbs serves as the predicate of a sentence. The adjectives class qualifies nouns while the adverbs qualifies verbs or clauses. The lemmas in different grammatical categories can not substitute each other.

2. Antonym: Another important relationship is the antonym. The antonym of one word is opposite in meaning to that word but this is not always true. A person who is not fat not implies that he is slim, he could be of average weight. Antonym is relationship in word form not in word meanings. For example death is antonym of life, but demise is not considered antonym of life. Such facts enhance the need to distinguish between word meanings and word forms. Antonyms are semantic relationships between word forms, so it is better to discuss antonyms in their specific context which is communicated through gloss in WordNets.

3. Hypernymy and Hyponymy describe semantic relations between a more general

term and specific instances of it.
Hyponymy:
X is a hyponym of Y iff f(X) entails f(Y) but f(Y) does not entail f(X):

Adam has an iphone $\Longrightarrow$ Adam has a phone
Adam has a phone $\not\Longrightarrow$ Adam has an iphone

Hypernymy: Y is a hypernym of X if X is a hyponym of Y.
Properties of hypernymy/hyponymy applies only to lexical items of the same word class and applies at the sense level. For example: "Africa" is hypernym of "continente" (continent) similarly "agricoltura"(agriculture) is hypernym of "coltura" (crop) and "arcobaleno"(rainbow) is hypernym of "arco" (arch).

4. Meronymy and Holonymy express the part-of relationship and its inverse.
   Meronym:

   "X" is a meronym of "Y" if Xs are parts of Y(s), or
   "X" is a meronym of "Y" if Xs are members of Y(s).

   Holonym: Holonymy defines the relationship between a term denoting the whole and a term denoting a part of, or a member of, the whole. Holonym is a transitive relationship. For example: "terra"(land) is meronym of "provincia" (province) similarly "uomo"(man) is meronym of "popolo" (people) and "Italia"(Italy) is meronym of "Italiano" (Italian).

5. Entailment is a semantic relationship between two words. Any word A entails B, if the meaning of B follows logically and is strictly included in the meaning of A. This relation is unidirectional, i.e. componendo (dialing) entails "chiamando" (calling), but calling does not entail dialing.
   Identifying new synsets and creating appropriate links among synsets is an ongoing task during building of new WordNets. In a mature corpus, the lemmas have greater richness as they are linked to more synsets.

3.3 Integrating MultiWordNet with LIS Lexical Resources

The LIS lexicon was integrated with the MWN database. The MWN provides rich lexical and semantic relationship between Italian words. It is linked with the basic sign representation of spoken language words (lemmas). The MWN database and the LIS signs dataset are interlinked to obtain lexical, semantic and translational relations. The MWN synsets representing different concepts are linked with the corresponding LIS signs. The two databases are integrated in a way that if a particular lemma among synonymies of synset also has the corresponding in the output in our lexicon, then the lemma is hyperlinked. Users can play and view the videos of the lemma signs which are hyperlinked by clicking on the link. All the syntactical relations in our database are inherited from the MWN syntactical relations.

3.3.1 Linking Procedure

For each sense of an Italian lemma I, we look for a MWN synsets S(I) including at least one LIS of L if exists in our database. A link between L and S is established. When the procedure has been applied to all Italian word senses, we can build the equivalence class of all sets of Italian words W which have been linked to the same LIS signs. Each set in the equivalence class is the Italian synset synonymous with some LIS signs.

$$f(I,L,S(I)) = \begin{cases} L \in S(I) & \text{if at-least one link is established,} \\ \phi & \text{otherwise.} \end{cases}$$

$$\forall \text{ I}, \mu(\text{I}) \iff \text{L , W} = \text{Sign}(\mu(\text{I}))$$

Where,
μ = A function to determine links between synset and LIS sign
L = Links between synset and LIS sign
S(I) = Synset linked to corresponding LIS sign
ϕ = empty set
Sign(.) = Sign related to corresponding link
W = Set of hyperlinked words related to the signs

The integration of LIS dictionary with MWN is useful for semantic dictionaries and semantic resource applications such as information retrieval and natural language processing. Since MultiWordNet has multilingual lexicons, such as Spanish, Romanian and Portuguese, it could be helpful for multi-language learning.

3.3.2 The MultiWordNet Integration Flow

By linking MultiWordNet lemmas to each video animation, it is possible to display the words which have the same meaning, although some of them don't have a corresponding translation in LIS, but deaf people can understand them with the help of synonyms. The LIS Dictionary performs a set of operations to translate an Italian word into a LIS sign. There are three different possible scenarios when a deaf users try to find the sign of a specific word:

1. The input word is not found.

2. The input word has only one correspondence.

3. The input word has multiple correspondences.

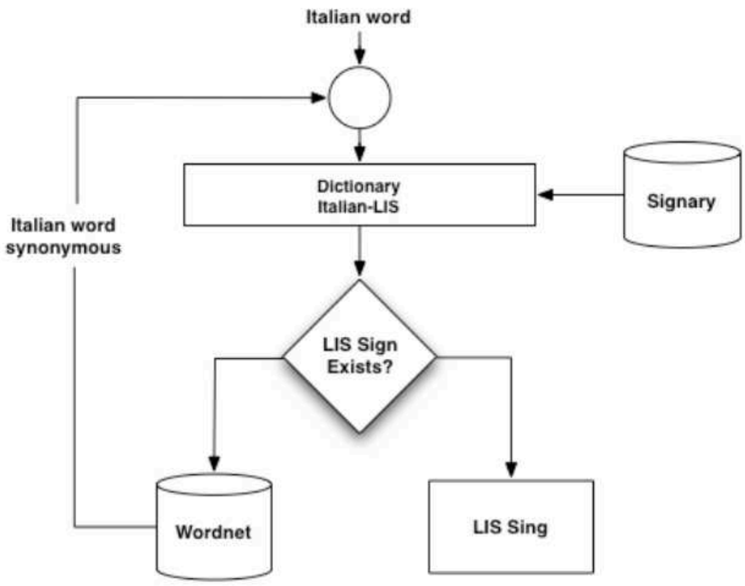


Figure 3.2. Searching lemmas in LIS dictionary integrated with MultiWordNet.

In the first scenario, the synset information associated to the word became the entry for a Wordnet search. If some of the word synonyms is already in the lexicon cases 2 and 3 can rise.
The second case gives an immediate translation but can also be supervised to check if more than one LIS sign is linked to the original word. In this case, the user is able to manually select the preferred LIS sing.
The third case happens when some disambiguation issue is related to the original word, since some words of the Italian language have multiple meanings (e.g., the Italian word "pesca" may refer to both the "peach" and the to fish verb). In this case, using the synset data associated to the retrieved results, the user is able to select the right LIS. Figure 3.2 summarizes the MWN integration flow.
The number of synonymous lemmas in a synset present in the MWN is shown in Figure 3.3 [38]. From the graph it can be seen that 3105 synsets have one synonymous member while 1608 synsets contain two synonymous lemmas, whereas 913 synsets have three corresponding synonymous words in the MWN database. Significantly there are two synsets for which there are twenty-nine lemmas which have similar meanings. Due to the limited LIS lexicon, covering a large vocabulary and providing corresponding LIS signs was a challenging task. To meet this challenge, the signs have been linked with the synsets. The sign for the requested lemma, synonyms and the MWN gloss are shown as output. Table 3.2 shows some synsets which have the maximum number of synonymous lemmas and their corresponding gloss in the MWN database. The synset relationship with lemmas helps to explore

more lemmas which are either synonyms of the synsets or used for similar meanings [40].

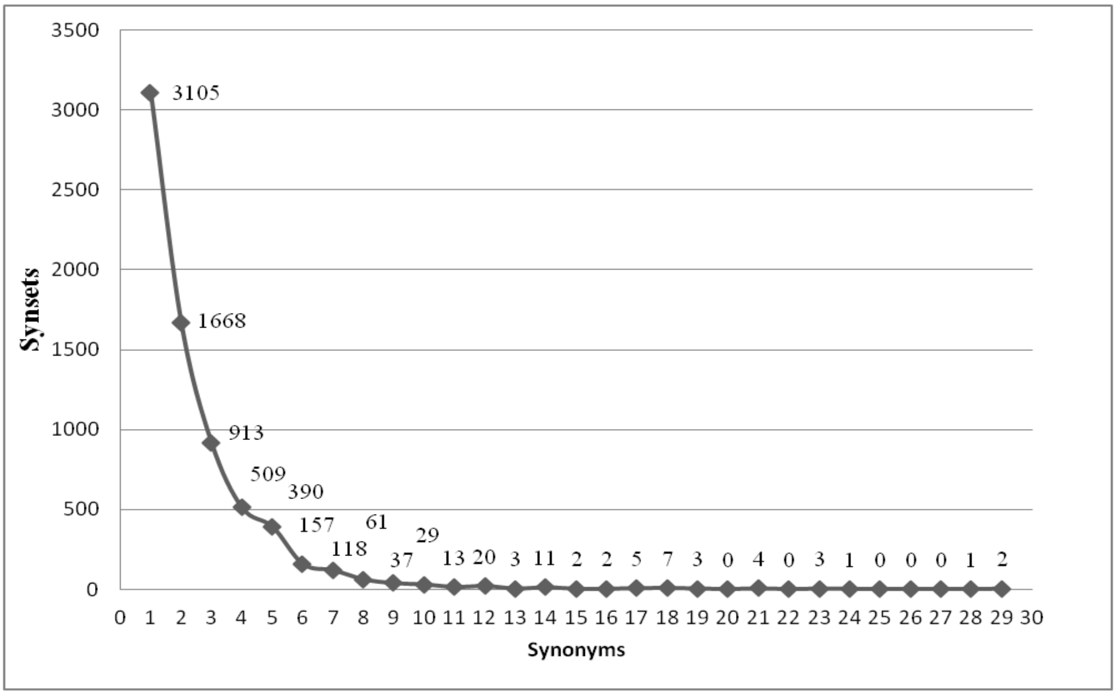


Figure 3.3. Distribution shows number of synonym lemmas in synsets

3.4 Formation of LIS MultiWordNet (LMWN)

The LIS Dictionary is integrated with the MWN in order to extend it with LIS information and hence form a LIS MultiWordNet (LMWN). LMWN inherits the semantic relationships from the MWN. It has the same semantic relationship model and exploits the possibility available in MWN through linking the LIS terms to the corresponding concepts in the MWN lexical database. A concept mapping procedure was adopted to form meaningful relationships between lexical words and the Italian sign language. The concept mapping process promotes and assists meaningful learning by encouraging learners to identify key words, establish relationships with other words and relate new concepts to the previous concepts [105]. In LMWN concept mapping is achieved by interlinking nouns, verbs, adjectives, and adverbs through semantic relations of synonymy.

Table 3.2. MultiWordNet synsets with maximum synonymous lemmas

Synsets	Translation	No. of Synonyms	Synonymous Lemmas	Gloss
Superbia	Pride	29	albagia, alterigia, altezzosità, altura, arroganza, boria, boriosità, burbanza, despotismo, dispotismo, grandigia, Iattanza, insolenza, muffosità, oltracotanza, orgogliosità, prepotenza, presupponenza, protervia, sdegnosità, sfrontatezza, sicumera, sostenutezza, spocchia, strafottenza, superbia sussiego, tracotanza, tronfiezza	Overbearing pride evidenced by a superior manner toward inferiors
Imbroglione	Cheat	28	abbindolatore, aggiratore, arcadore, arcatore, baro, bidonista, blackleg, cagliostro, camorrista, frodatore, gabbacompagno, gabbacristiani, gabbamondo, gabbatore, giuntatore, imbroglione, impostore, lestofante, magliaro, mistificatore, pataccaro, pelagatti, pellaccia, raggiratore, truccone, truffaldino, truffatore, turlupinatore	A person who swindles you by means of deception or fraud
Affaticato	Fatigued	23	affaticato, debilitato, debole, distrutto, esaurito, esausto, estenuato, fiaccato, indebolito, logoro, macero, prostrato, sfiancato, sfibrato, sfinito, slombato, snervato, spossato, stracco, stremato, stroncato, svigorito, trafelato	Completely emptied of resources or properties

3.4.1 MultiWordNet Coverage in LIS MultiWordNet

Preliminary comparisons of LMWN with the MWN data reveal some interesting similarities and differences. The comparison between the syntactical relationships of MWN is shown in Figure 3.4 at left side while for LMWN it is shown in Figure 3.4 at right side. In MWN hypernymy and hyponymy type of relations occur the most (55%). The meronymy and holonymy relations are 24%. From LMWN relationships it is clear that hyponymy and hypernymy are the dominant categories of relationships and cover most of LMWN with a significantly higher percentage (75%). Meronymy and holonymy make up only 14% in the LMWN. If we compare the relationship types of MWN and LMWN, it is evident that LMWN mainly inherited the hypernymy and hyponymy relationships.

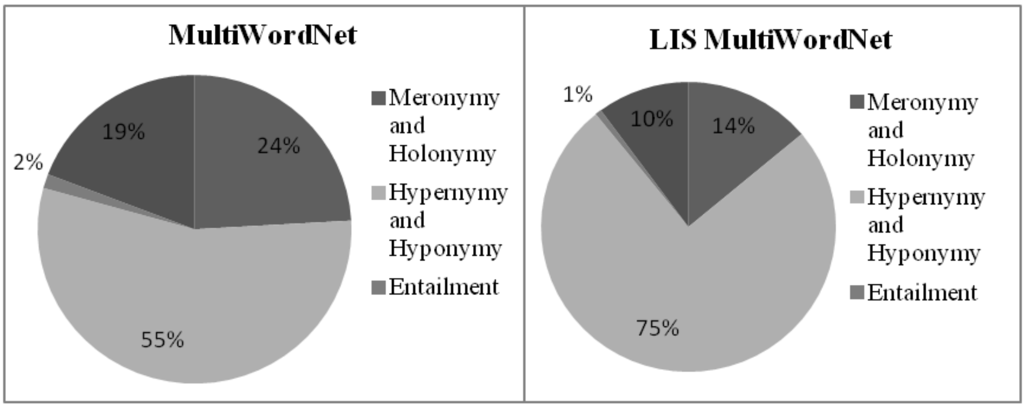


Figure 3.4. Distribution of semantic relationships supported by MWN and LIS-MWN

The relationship of LIS signs with the MWN synsets enables us to cover and help users to explore larger datasets using LMWN. Figure 3.5 shows the relationship between synsets and LIS signs. If we look at the breakdown of LIS signs corresponding to synsets, there are 792 such relations which have one sign linked with a single synset. The signs associated to two synsets are slightly less than these having only one synset. There are 504 relations where a sign is associated with two synsets. Interestingly, we have two common signs linked with twenty three different synsets [40].

The newly created lexical resources cover 30% of the MWN lemmas existing within the synset. The comprehensiveness of our lexical video dataset is an important aspect. Table 3.3 compares lemmas and synsets of MWN and LMWN according to the grammatical category distributions and provides the percentage of each MWN covered in LMWN. The left hand side of the table provides statistics of grammatical categories and LMWN coverage of lemmas. Verbs have the highest relative coverage

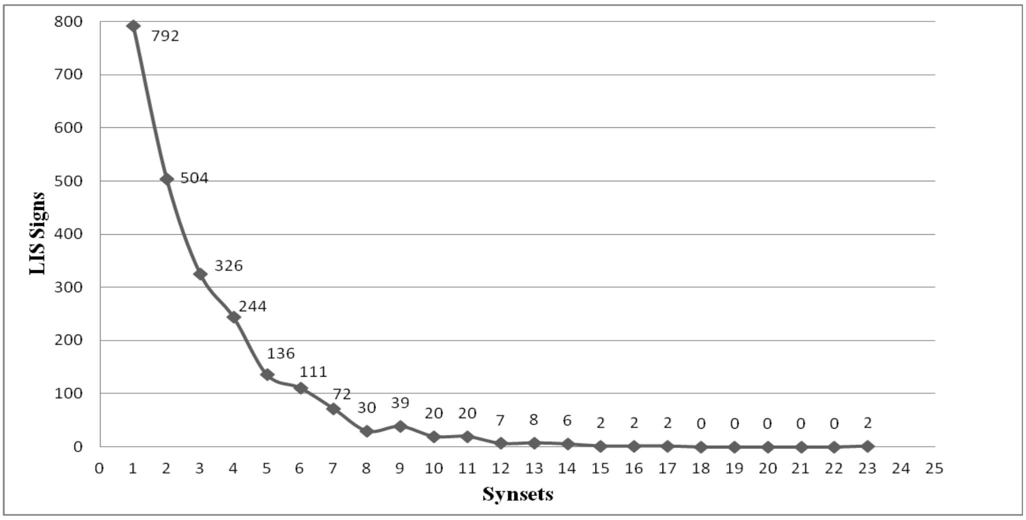


Figure 3.5. Relationships of LIS signs with synsets

(4534 verbs) covering almost half of the MWN verbs. 9345 nouns are linked with LIS lemmas. Their coverage percentage is the lowest of all categories, about one fifth of the total MWN noun lemmas covered. There are 2417 adjectives which represent about one third of the total. Overall, LMWN covers 17214 MWN lemmas. This is quite significant and is about one fourth of the total.
The right hand side of Table 3.3 provides the synset characteristics of MWN and LMWN. The right-most column in Table 3.3 describes the cumulative percentage of LMWN synsets. LMWN provides coverage for 18.3% of the MWN synsets. The results indicate that signs for the 13.5% nouns (total 3910), signs for the verbs 36.5% (1814 in total), signs for the 25.6% adjective (total 998) and 27.6% adverb synsets (342) associated with a sign are covered in LMWN [40].
LMWN allows the users to see the signs of the synonyms of the any specific word if it exists in order to better understand the concept. Sometimes more than one word represents the same sense which can be understood from existing signs. The meanings of the requested words are categorized into their morphological behavior and lexical category.

3.4.2 Conclusion and Discussion

This chapter illustrated the process for the creation of a LIS lexical resource and reported on the analysis of a small fragment of the lexicon. The MWN sysnsets having a corresponding LIS sign in the database are hyperlinked and can be explored. The results are grouped by synonymous word senses, and show them in the structures of grammatical type hierarchies. The grammatical distribution (i.e. noun, verb etc)

Table 3.3. Lemmas supported in LIS MultiWordNet according to grammatical category

	Italian Lemmas			Italian Synsets		
Grammatical Category	MWN	LMWN	Coverage (%)	MWN	LMWN	Coverage (%)
Nouns	49,003	9321	19.0	28,429	3910	13.8
Verbs	9861	4534	46.0	4975	1814	36.5
Adjectives	6443	2417	37.5	3898	998	25.6
Adverbs	2243	942	42.0	1239	342	27.6
Total	67,550	17, 214	25.5	38,541	7064	18.3

helps the user to better understand the lemma and its senses. The users will not get only the meaning of words in text language but also synonyms, related words and the synset along with the LIS sign. Even if the searched lemma does not have a sign in the database, users can see the signs of the related synonyms to better understand the concept. We implemented the MultiWordNet infrastructure to support both the limitation of annotated words and the disambiguation of word meanings. This level of semantic connections cannot be attained by a conventional paper dictionary.
In this work the LMWN building process is initiated with a limited number of LIS signs to derive a large set of meanings i.e. possible word meanings associated to a specific sign and different forms of sign existing in LIS. LIS Lexicon dataset integrated with multilingual lexical MWN database which makes LMWN was discussed. More specifically, we have focused on the lexicon by reporting the steps of the LMWN creation process in detail. We have more than three thousand animated videos in the lexicon dataset. More than seventeen thousand lemmas of the MultiWordNet can be translated using LIS Dictionary, which is about one fourth of the total words existing in the MWN. It connects senses by semantic relationships similar to subclass relation and hence induces a hierarchical organization as well as partial/whole relation.
Overall, this chapter concludes that the relatively small size of the LIS lexicon can help to cover a significant portion of the MWN. The words which have maximum frequency should be given more attention in the creation process. This initial study of lexical frequency in LIS provides much needed evidence about frequency which can be used to design experiments about LIS processing. It has also further helped our understanding of distribution of grammatical sign categories (and unique sign types) within LIS across signed and spoken languages. Such comparisons both within and across language modalities are crucial for a better understanding of language processing, acquisition, variation and change in language in general, and also for theory

building.

Chapter 4

Servification of LIS Translation Processes

Services provide an abstraction mechanism and usually a service is defined by its functionality, input and output. For example a text to sign language service can be defined in terms of effects it has on deaf users, and the resources used to produce the output will be encapsulated in it. Process models help us in understanding complex problems and its potential solution through abstraction, naturally software systems greatly benefit by it. The process models are expressed using concepts that are much less bound to the underlying implementation technology and are much closer to the problem domain [106]. By raising abstraction levels, Model Driven Development (MDD) aims to simplify many complex programming tasks. Translation is one of the complex programming tasks of computational linguistics, since its very beginning.

There are two fundamental machine translation approaches: Statistical and Rule-based translation. Statistical machine translation (STM) uses a statistical model, based on the analysis of a corpus, to generate the translated text while Rule-based machine translation uses linguistic rules to analyze the input contents in its source language to generate text in the target language. Rule-based translation is much more complex than word to word translation and different sets of rules are applied in three different phases: parsing, semantic interpretation and generation. For successful translation Rule-based translation requires syntax analysis, semantic analysis, syntax generation and semantic generation. The details of both translation methodologies will be discussed in section 4.2. The fundamental question is whether there is any way to conquer this complexity that minimizes the technical gap so that programmers can work close to the level of human thought? The eXtreme Model-Driven Development (XMDD) paradigm is a very rigorous approach for tackling the complexity challenge of service-oriented design of complex applications. XMDD is extreme as its involves the customer/application expert continuously throughout the

whole system life cycle and it inherits the advanced model-driven methodology of the 'One-Thing Approach' in which user-level models are successively refined from the user perspective until a sufficient level of detail is reached, where elementary services can be implemented solving application-level tasks [34, 107]. For quality assessment of the produced translation, the involvement of user is compulsory as they can decide which of the virtual interpreters (either Rule-based or Statistical) versions lead to better comprehension of the translation. During the interaction with signing deaf users it is discovered that the Rule-based translation can be refined by utilizing communicative strategies commonly used by signing deaf individuals, for example: 'coast'(in Italian: 'costa') could be better comprehend by using compound form of 'sea' and 'coast' (in Italian: 'mare+costa'). The usage of compound form is a viable solution for geographical places and proper names (of geographical places). Some signs are in fact non-lexical classifications like a word usually used in singing 'la la la'. In this case the lexical semantic might be not relevant in translation process for identifying the sign because at the phrase level it is possible to identify the sign referent.
The chapter is organized into four sections. Section 4.1 introduces the Extreme Model-Driven Development (XMDD) methodology and discusses the Java Application Building Center (jABC) which is first framework for XMDD. Section 4.2 provides detail about Statistical and Rule-based translation methods used in automatic translation of Italian text to Sign Language. Section 4.3 represents the Rule-based translation processes into jABC workflows and provides detail about each Service Independent Block (SIB). Finally Section 4.4 concludes the chapter by highlighting the advantages of transformation of Rule-based translation processes into service oriented one.

4.1 Extreme Model-Driven Development (XMDD)

Classical software development methodologies (for details see section 5.3) are no longer adequate for the number of application programming especially for dealing with the continuously changing requirements in heterogeneous environments [34]. XMDD works on the model level, and thus has models and their architecture as the primary artifact. XMDD adds a lot of value to existing ways of writing software, since using rich GUI with architectural views of the system is much easier than old ways of writing software. XMDD is very flexible, since the level of modeling and its granularity can be varied according to requirements during the design and development process [35]. The automatic translation into sign languages involves various components including linguistic translation and virtual character animation (see section 2.3.2). These components and interfaces typically evolved independently

requires a lot of effort in interoperability and deployment due to heterogeneous resources and sometimes necessary requirement to upgrade some components, which no longer behave as expected. The Enthusiasm framework used in virtual character animation component deals with interoperability among OGRE3D(3D video rendering), OpenAL(spatialized audio rendering) Tarta4D(handle scenes, objects, paths and animations), MESH editor(flavor declination of the Enthusiasm GUI adding an higher abstraction level in the scene management) and Java Apps. Integration testing and reliable interoperability are major risks in overall system implementation and deployment. At the larger scale of system development, XMDD methodology can better integrate heterogeneous components and strictly separates compatibility, migration, and optimization issues from model/functionality composition. In this way XMDD has the potential to drastically reduce the long-term costs due to version incompatibility, system migration and upgrading, and lower risk factors like vendor and technology dependency [107].
XMDD combines the decisive features of eXtreme programming(by providing immediate feedback through requirements and design by model tracing, simulation, and early testing), service orientation(existing or external features, applications or services can be easily integrated into a model), aspect orientation(increasing modularity by allowing the separation of cross-cutting concerns) and model-driven design(controlling the development at modeling level) [34]. XMDD focuses on bringing software development closer to conceptual design, so that architecture of a software solution could be understood by someone unfamiliar with programming concepts. Early validation is supported in XMDD, since the project can be compiled and tested any time. Paperwork and documentation can not be eliminated fully in XMDD, but the architecture act as a map to navigate through the system and helps in making the system more understandable with less text. XMDD, as a way of automating software design approach top-down, where the developer pays more attention to the architecture since code generation is automated. Changing the architecture in this case will not mean going all the way through the code structure, because of that such changes can be applied easily. Since such changes will happen often if product specifications change, XMDD is much more suited for working with prototypes than other means of development. The global system model is constructed in a hierarchical fashion with the help of a library of models, which reduces the model's complexity through modularization, enables a collaborative creation of the model by means of iterative refinement and hierarchical modeling also allows the reuse of entire models [108].

4.1.1 Java Application Building Center (jABC) Framework for Workflow Design

The Java Application Building Center (jABC) is a multi-purpose, domain-independent modeling framework based on Lightweight Process Coordination (LPC) [109]. In particular, it has been applied as a comprehensive service engineering environment following the XMDD paradigm [110]. The jABC can be used by business and application experts to graphically orchestrate complex business processes into running applications on the basis of a service library [35]. Users can easily develop services and applications in jABC by composing reusable building blocks (called Service Independent Building Blocks, SIBs) into graph structures (known as Service Logic Graphs, SLGs). This development process is enriched on extensible set of plugins which enhances the functionality by providing support e.g. in rapid prototyping, formal verification, debugging, code generation, and evolution.
The basic idea of LPC is to add an additional coordination layer (composed of application and services layer) into the well known three-tier architecture based on presentation tier, logic tier and data tier. This coordination layer is purely model driven development layer, created and managed within jABC. In this program the user builds his coordination models by arranging predefined building blocks (called SIBs) simply by drag and drop. LPC as a service-oriented, model-driven development approach, offers a number of advantages when integrating distinct functionalities including simplicity (easy to explain for non programmers), agility (changes are welcome), customizability, consistency(evolution of same model throughout the process), verification (using techniques like model checking and local checks), service-orientation (easy integration of existing features or applications) and universality (platform independence) [111].
The graphical editor of jABC facilitates composing and manipulating SLGs by placing SIBs (representing concrete services) on a canvas and connecting them with labeled edges (called branches) according to the flow of control. Further functionality can be added by means of plugins. Figure 4.1 shows the user interface of jABC which is displaying three major areas:

1. Project and SIB browser: At the left upper corner of Figure 4.1, the project browser displays the list of all available projects. The SIB browser tab displays all available SIB classes in a tree view. The SIB browser supports a hierarchical as well as flat display. The corresponding SIBs are organized by means of a taxonomy that is a tree representation of a directed acyclic graph.

2. Inspectorpane: At the left lower corner of Figure 4.1, the inspector-pane provides detailed information on the currently displayed SLG and shows properties of SIBs.

3. The canvas is the central interface of the jABC. In the Figure 4.1, the canvas is showing the SLG which presents the process of MultiWordNet integration with Italian Sign Language dictionary.

The collaboration work on jABC is divided in several roles with different responsibilities.

- Application expert: The application expert or business expert models his process flow in jABC. The application expert can be a person with no programming skills, or not a technical expert of XMDD. The application expert will construct his model either from existing SIBs or he can define new SIBs and its properties like name, appropriate parameters and branches.

- Domain expert: The domain expert has technical knowledge of the application domain. The domain expert customizes the jABC, selects suitable services and plugins on the basis of domain specific characteristics to facilitate the application expert. The domain expert uses the Genesys framework to select the required code generators, in order to create the code.

- SIB expert: The IT expert or SIB expert is usually developer with detailed knowledge about the development of SIBs and appropriate plugin interfaces. He also create more concrete models in lower hierarchy.

4.1.2 The jABC as an XMDD Environment

The jABC can be regarded as a first framework for XMDD. SIBs are the elementary building blocks of jABC and refers towards abstract representation which is independent from its context or usage and can be reused. The granularity of the services represented by SIBs covers from simple functions like adding two values to complex systems such as complete translation system [107, 108]. jABC contains a library of SIBs known as Common SIBs which are very generic and can be used in any domain. The existing and newly created SIBs are properly organized into different categories like ftp SIBs contains functionality related to download or upload contents at FTP server and openligadb SIBs contains sports results of different matches.
The central model structure of the jABC are Service Logic Graphs (SLGs) which are directed graphs that represent the flow of actions in an application. SLGs model the process flow with the help of SIBs. In an SLG a SIB may represent a single concrete service or also stand for a whole subgraph (another SLG), thus serving as a macro which hides more detailed processes [112]. In Figure 4.1 *SearchinLISDictionaryforSigns* is a macro SIB which contains another SLG which is expanded in Figure 4.2 and shows a workflow model of searching a word in Italian Sign Language dictionary.

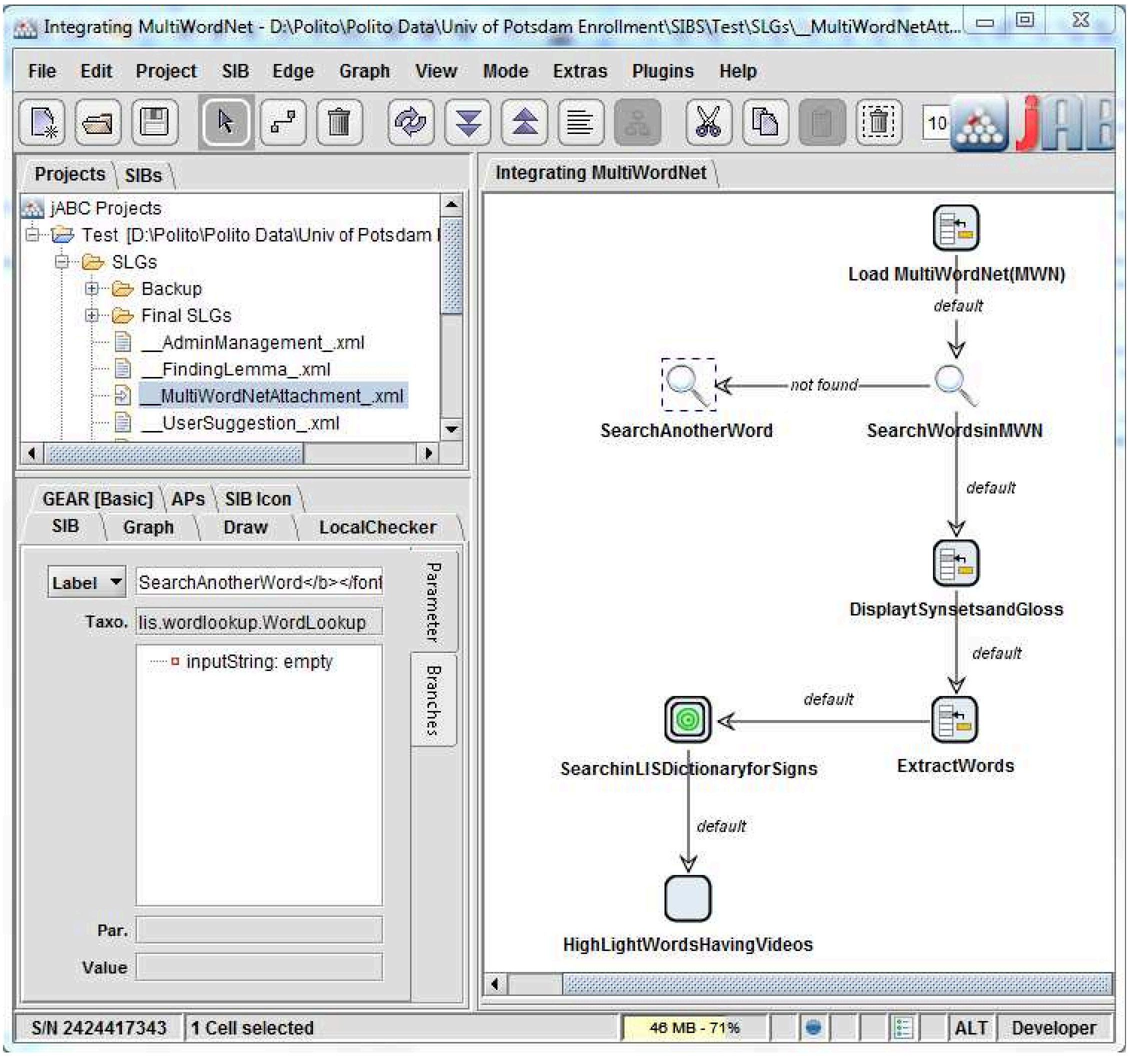


Figure 4.1. The Java Application Building Center (jABC) user interface

Additional functionality is brought to the jABC framework by means of different plug-ins and provides interpretation of the model [113]. The basic jABC framework can be extended by two principal kinds of plugins:

1. Semantics plug-ins provide functionality for specific interpretations of the graphical model or its components. Examples of such plug-ins are the Tracer (model interpreter), Genesys (the code generator framework) and PROPHETS (enables constraint-driven workflow design according to the loose programming paradigm) [114].

2. Feature plug-ins extend the range of general, domain independent functionality

that is available in the jABC. Examples of such plug-ins are the TaxonomyEditor, the GEAR model checker, the SIBCreator and the Layouter.

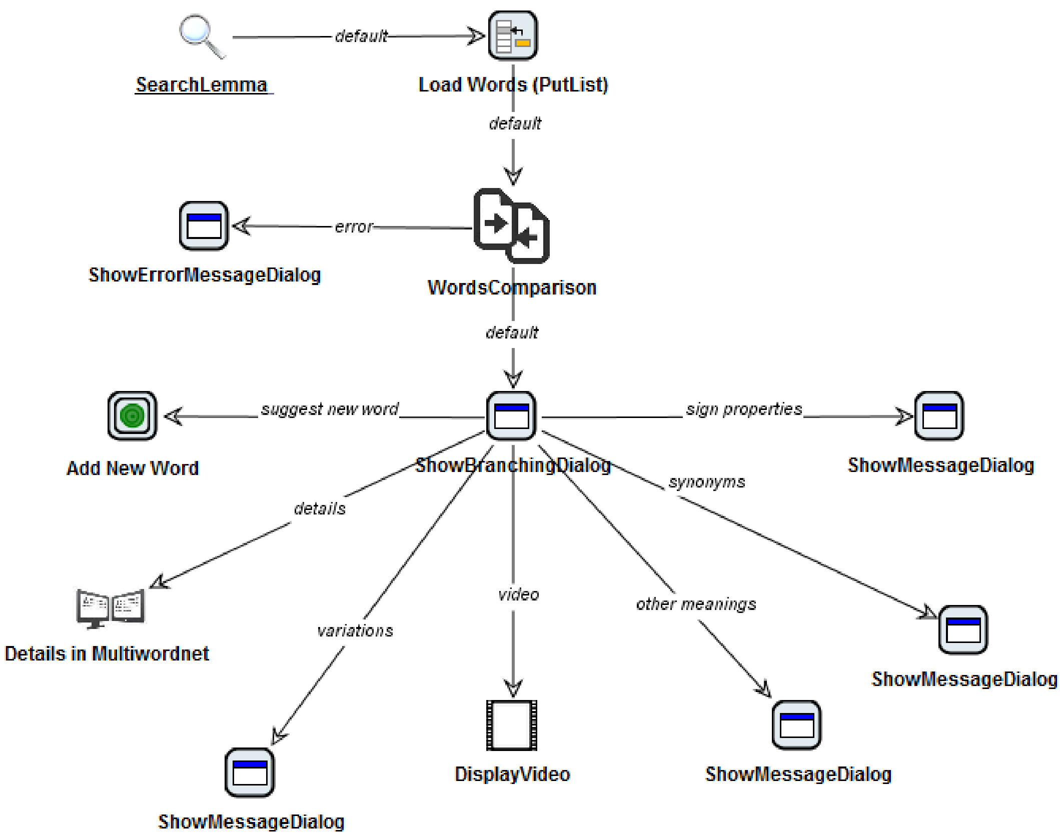


Figure 4.2. The Service Logic Graph of searching a word in Italian Sign Language Dictionary

jABC is mature framework and has been used in several application domains: complex supply chain management with IKEA [115], telecommunication services especially intelligent network services [116], modeling and execution of bio-informatics workflows [110, 117, 118, 119], online conference system [120], compiler optimizations [121], the semantic web services challenge [122], remote configuration management of any service [123], robot control programs [124], system-wide test cases [125], graph-based modeling and generation of formulae [126] and graphical construction of game strategies [127]. In following sections, this dissertation presents Rule-based translation of Italian text to Sign Language process modeling in jABC.

4.2 Italian Text to Sign Language Translation

Machine Translation (MT) investigates the use of computer programs to translate text from one source language to another target language. MT becomes a difficult task due to the complex structure of natural languages. Many words have multiple meanings, every meaning should be disambiguated from others, sentences may have multiple interpretations, and certain grammatical rules in one language might not exist in another language.
MT systems can be categorized into two fundamental approaches: the Rule-based approach and the Corpus-based approach as described in Figure 4.3. Hybrid MT approach uses multiple machine translation approaches in parallel and final output is generated by combining the output of all the sub-systems. Hybrid systems most commonly use combination of Statistical and Rule-based translation approaches to improve the accuracy of the translation. In Sections 4.2.1 and 4.2.2 the Statistical and the Rule-based translation approaches are discussed in detail.

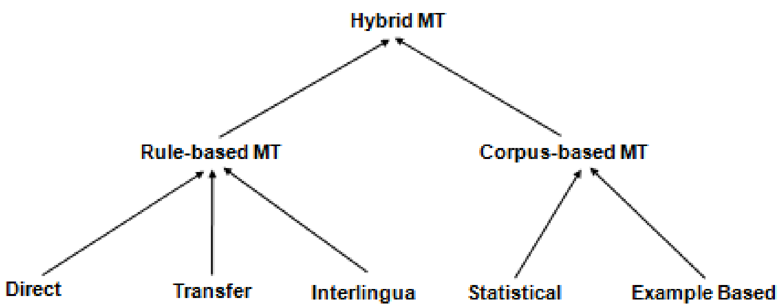


Figure 4.3. Taxonomy of various Machine Translation(MT) approaches

Example-based translation and Statistical translation are two approaches which are used in Corpus-based paradigm. Example-based machine translation (EBMT) was suggested by Makoto Nagao in 1984 and used for translation between Japanese and English [128]. The basic idea of EBMT is to reuse existing examples as the basis for new translations. The EBMT uses the idea of analogy, which advocates that usually people do not translate sentences after deep linguistic analysis rather they decompose sentences into certain phrases, then they translate those phrases and finally compose them to form a sentence. The EBMT systems are trained from bilingual parallel corpora, which contain sentence pairs. Table 4.1 shows an example of bilingual corpus with minimum pairs where both sentences vary in one element.

The EMBT system learns three units from this translation and different combinations of these units will be used for translation of similar sentences in future:

Table 4.1. Example of bilingual corpus.

English	Italian
Do you know how to use *a dictionary*?	Sai come usare *un dizionario*?
Do you know how to use *this machine*?	Sai come usare *questa macchina*?

1. Do you know how to use X ? corresponds to Sai come usare X?.

2. *a dictionary* corresponds to *un dizionario*.

3. *this machine* corresponds to *questa macchina*.

4.2.1 Statistical Translation

The first SMT idea was introduced by Warren Weaver in 1949 while in 1993 the idea of SMT is re-introduced by IBM [129]. Statistical Machine Translation(SMT) needs a suitable corpus of SL data for training as it treats translation as a machine learning problem in which a learning algorithm is applied to a parallel corpus. SMT models consider every sentence in the target language as a translation of the source language sentence with some probability. The best translation will be considered which one has the highest probability.
MOSES is a SMT system that allows to automatically train translation models for any language pair. All you need is a collection of translated texts (parallel corpus). Parallel corpus is a collection of sentences in two different languages, which is sentence-aligned, in that each sentence in one language is matched with its corresponding sentences in the other language. Once the model is trained, an efficient search algorithm quickly finds the highest probability translation among the exponential number of choices [130]. Our text to Sign Language translation system is based on MOSES.
There are different ways adopted for translation in SMT including word-based translation and phrase-based translation where the later is more dominant in research nowadays . In word-based translation, the fundamental unit is a word and its mapping with the target language is practically not one to one due to compound words, morphology and idioms. The fertility is the ratio which tells how many foreign words are produced against native word. The phrase-based translation systems outperform word-based translation systems in dealing with high fertility [131]. The phrase-based translation model is based on the noisy channel model and used in the MOSES toolkit. A Bayes rule is used to reformulate the translation probability for

translating an Italian sentence s into English e as:

$$\text{argmax}_e \ p(e|s) = \text{argmax}_e \ p(s|e) \ p(e)$$

where p(e) is representing a language model and p(s|e) exhibits a separate translation model p(s|e).

A uniform probability distribution over all possible segmentation is assumed during decoding, the foreign input sentence s is segmented into a sequence of I phrases $\underline{s}_1{}^I$. Each foreign phrase $\underline{s}_i$ in $\underline{s}_1^I$ is translated into an English phrase $\underline{e}_i$. After translating each phrase into English, the phrases may be reordered while phrase translation is modeled by a probability distribution $\phi(\underline{s}_i|\underline{e}_i)$. A simple distortion model is used in MOSES as:

$$d(\text{start}_i,\text{end}_{i-1}) = \beta^{|\text{start}_i\text{-end}_{i-1}\text{-}1|}, \quad \text{where } \beta \text{ has an appropriate value.}$$

According to MOSES model, if a sentence s is provided as input then the best English output sentence e_{best} will be as following:

$$e_{best} = \text{argmax\_e} \ p(e|s) = \text{argmax}_e \ p(s|e) \ p_{LM}(e) \ \omega^{\text{length(e)}}$$

where p(s|e) is decomposed into $p(\underline{s}_1{}^I|(\underline{e}_1 I) = \phi_i{=}1^I \phi(\underline{s}_i|\underline{e}_i) \ d(\text{start}_i,\text{end}_{i-1})$.

ω is a word cast factor for each generated English word in addition to trigram language model p_{LM}.

4.2.2 Rule-Based Translation

The Rule-based Machine translation (RBMT) systems are summarized in the Figure 4.4. The RBMT systems can be roughly organized into three approaches: direct, transfer or interlingua [132]. The three approaches represent different depth of analysis provided by the system.

- Direct systems process word-by-word translation by matching source words with targets with the help of some dictionary. This approach is least sophisticated and can be used in translating some product catalogs or could help in human-assisted translation.
- The Transfer system is further divided into two parts one is syntactic transfer and the other is semantic transfer. The Transfer systems analyze the input text to some syntactic or semantic level, and then a set of transfer rules produces a corresponding syntactic or semantic structure in the target language. A generation component is used to convert this linguistic structure into a target-language surface form [24]. The Syntactic transfer approach is also called superficial transfer and it is suitable for translation in languages belonging to the same family. Syntactic analysis deals with the word order and the

way the sentences are put together. The Semantic transfer approach which is also called deep transfer is more suitable for distant languages. The semantic analysis deals with word meanings.

- The basic idea of the Interlingua translation is that the source language text is transformed into an interlingua, i.e., an abstract language-independent representation. In the second phase the target language is generated from the interlingua. The Interlingua approach is more suitable if the source text is transformed into several target languages. The system can handle more divergences if the source text is analyzed more by going up in the pyramid but it also enhances the amount of domain specific development work.

Both Transfer-based and Interlingua-based machine translation systems have the same idea except transfer translation approach does not transform the source text into a language-independent representation.

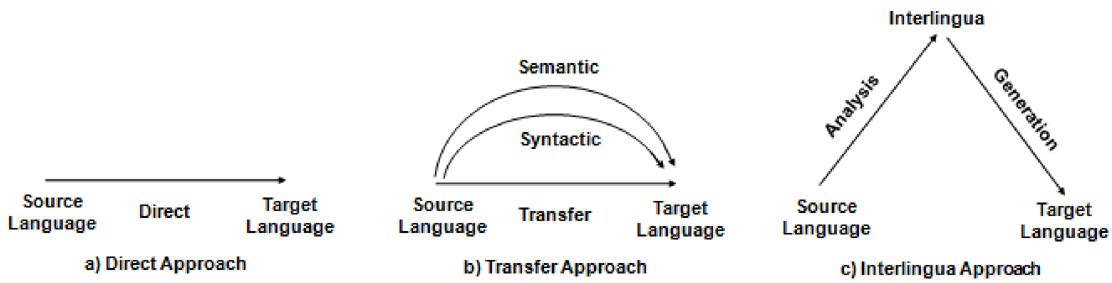


Figure 4.4. Types of Rule-based Machine Translation(MT) systems

4.2.2.1 Rule-based Machine Translation in ATLAS

The Rule-based translation in ATLAS is performed according to interlingua approach which is currently most widely used method in machine translation in specific domains due to its high accuracy. The goal of the Rule-based translator is to extract pieces of information from Italian written text and encode them in a form (AEWLIS, see details in section 2.3.3) which can be provided as input for the phase of Signed Language generation. Due to the complexity in natural languages it is difficult to implement systems that achieve full understanding in open domains, the Rule-based approach can be better implemented in limited domains like in our case the translation is aimed at weather forecast domain. Figure 4.5 describes the overall organization of the Rule-based translator developed by Interaction Models Group at University of Torino in which highly modular approach is adopted [3].

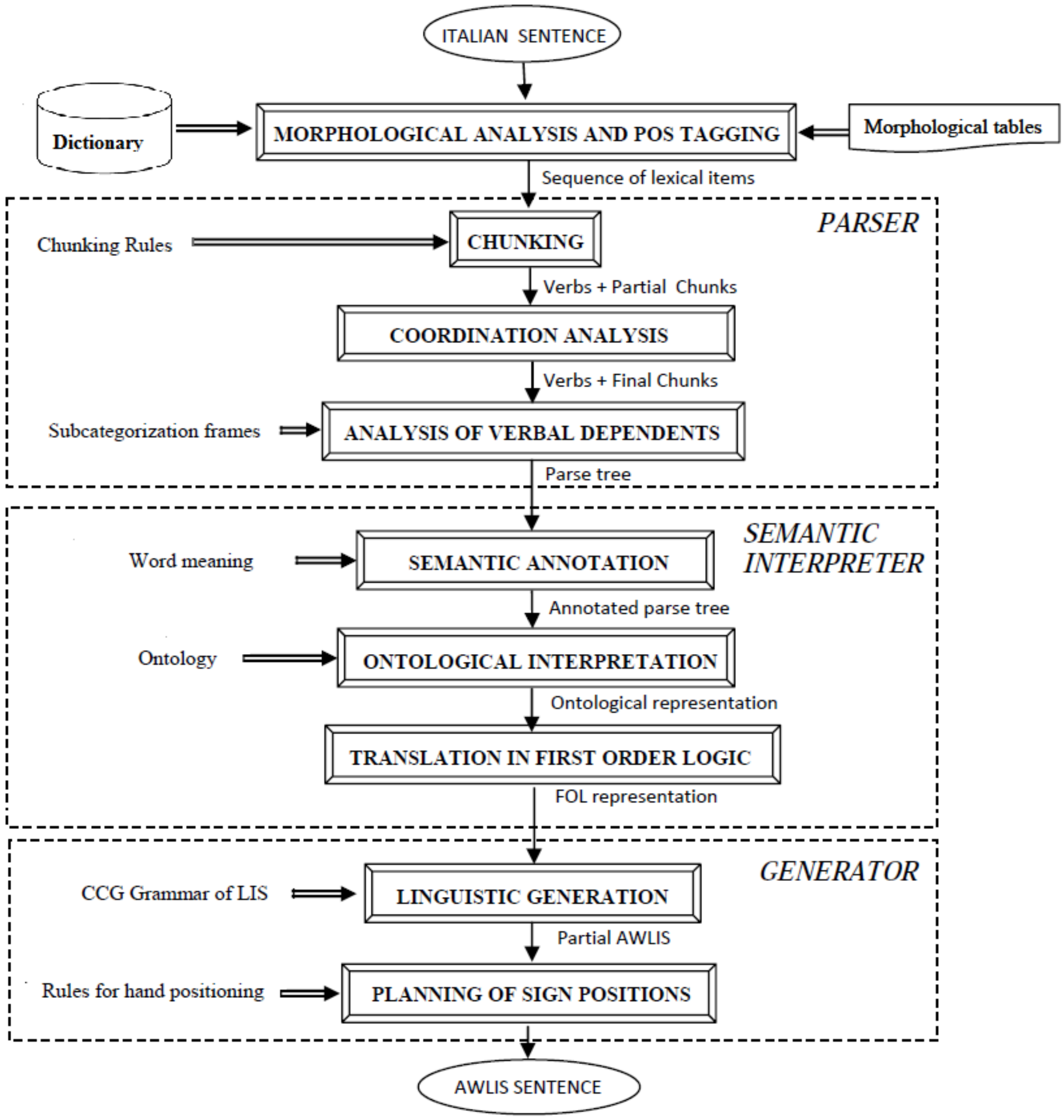


Figure 4.5. The modules of Rule-based translation approach [3]

Tokenization is the first step, which classifies various types of entries such as general words, proper names, abbreviations, dates, numbers, and so on. The Italian text is analyzed and POS tagged (part-of-speech tagging) by means of a morphological analyzer which accepts tokens as input and produces a sequence of words (each of which is possibly ambiguous). The morphological analyzer uses various knowledge sources including a dictionary and a suffix table. The morphological analysis provides grammatical information of a word given its suffix.

The output of this morphological analysis is parsed and syntactically analyzed by using the Treebank TUT approach [133]. The parsing process tags each word with syntactic categories and subcategories appearing in the treebank. A parse tree is produced as a result of the parsing process according to dependency grammar rules [3]. The Italian Sign Language (LIS) has a lexical restriction as its lexical resources are very limited and the Radutzky dictionary which is officially used for Italian Sign Language contains less than three thousand lemmas so the translation of most of the Italian words in LIS is not possible. It is therefore necessary to pass through a semantic representation of the sentence and find such words whose counterparts in LIS are also available. This particular task is performed by the semantic interpreter where semantic annotation of the sentence is performed resorting to the word meaning. The output of the semantic annotation is an annotated parse tree. An ontological representation is derived by means of the weather (meteo) ontology which is a formal, but partial, representation of the weather forecast domain and the outcome is represented using First Order Predicate Logic.
The ontological representation is mapped into frames representation which contains all information extracted from the Italian sentence, that can be useful for the generation of the corresponding AEWLIS interpretation. The outcome is then translated through Combinatory Categorial Grammar (CCG) and converted into a subset of the AEWLIS formalism. The CCG is efficiently parsable, yet linguistically expressive grammar formalism. AEWLIS is the output of the Rule-based translation process which is finally delivered to the animation system.

4.2.3 Use Case Mapping

The automatic translation into Sign Language process aims at translating texts into the Italian Sign Language (LIS) and visualizing it through animated virtual characters on different client media. The complete translation flow consists of three main modules: the external ingestion system, the core and delivery system. The external ingestion system manages the input in order to generate the structured text from a generic source as plain text, teletext page, subtitles, web page, audio stream, and TV Program. The core is responsible to translate structured text into virtual actor commands. The core component is based on Assisted editor, Rule-based translator, Statistical translator and Translation Merger modules. The delivery system uses the stored results produced by core component and delivers them on web and mobile platforms. The main components involved in the complete translation process are summarized in Figure 4.6.

The Use Cases define the interactions between external actors and the system processes to accomplish a specific Service. The following Use Cases describes the users' interaction with the system and discuss different scenarios from selecting

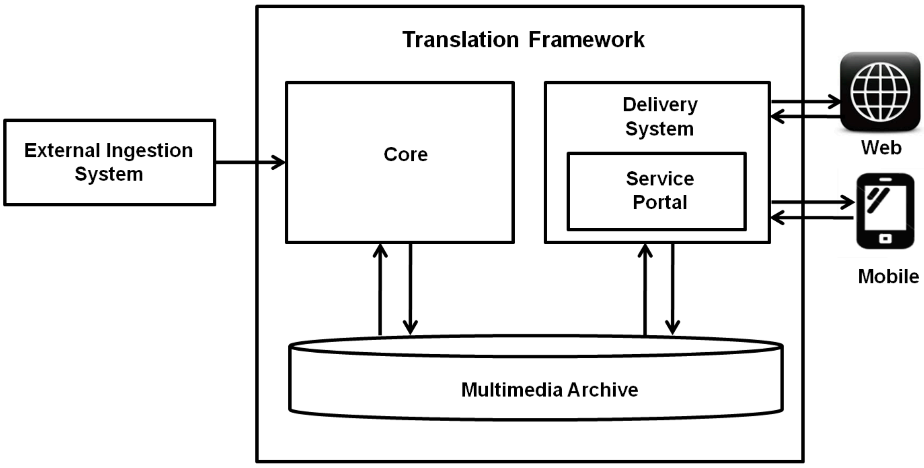


Figure 4.6. Automatic translation into Sign Language framework

ingestion source to creating virtual commands.

4.2.3.1 Teletext Pages

The automatic translation into Sign Language process starts from the selection of ingestion source. If some users have no difficulties in reading and understanding teletext pages in Italian but they prefer their translation in Sign Language, the teletext pages can be selected as input for the translation system. In this way they can share this information with their deaf friends that do not understand written Italian language very well. Teletext is a television information retrieval service that provides information about news, weather and TV schedules. Figure 4.7 shows the teletext ingestion process and its transformation into structured text.

The user selects a teletext page from external archive and request its translation into LIS. The structure analysis of teletext page (title, topic, content) is performed during the parsing process and XML schema is applied after removing teletext control characters. The output is stored in multimedia archive. The linear text is further converted into structured text according to the platform specifications.

4.2.3.2 Tv Programs

In an alternative scenario, the user can select a television program for translation in LIS or can also access the archive of television contents that are translated into LIS by the animated Virtual Actor. The transformation of a TV Program transcription speech into a structured text is shown in Figure 4.8. The process starts with user's

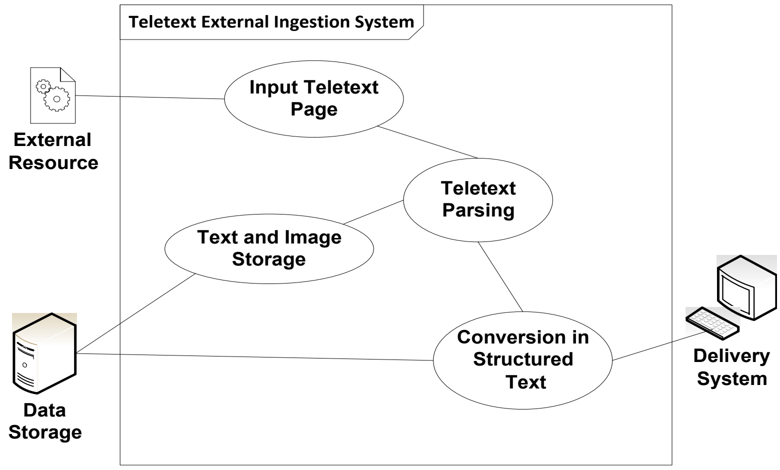


Figure 4.7. External ingestion system, teletext pages as input

selection of favorite Tv program. The audio track of the TV program is extracted and converted into synchronized text. The translation system accepts the input text in structured format so the text extracted from audio stream is further converted into structured text. In the automatic translation into Sign Language project, the users also have the options to consult the catalog of films and documentaries translated into LIS or can translate text of any website

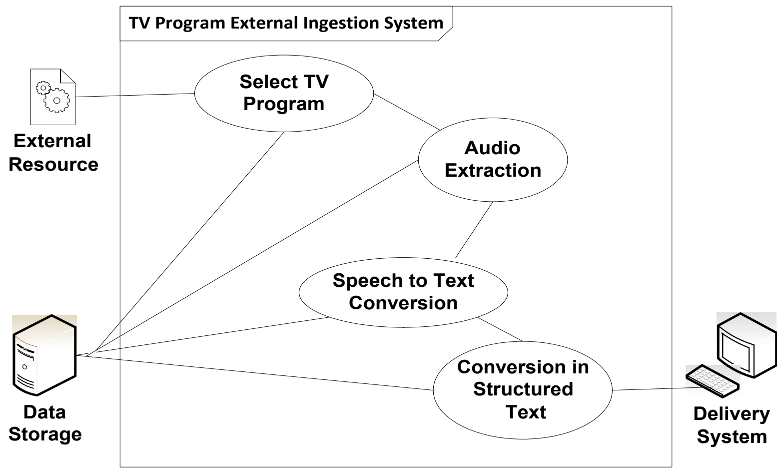


Figure 4.8. External ingestion system, TV program as input

The translation of Italian text into Sign language is performed in three different ways. An editor for assisted translation called ALEA is used to translate the contents manually. In manual translation the Italian text is imported and text segmentation is performed. In text segmentation, the sentences are divided into meaningful units such as words or topics. For the translation process the ALEA takes information from the Italian dictionary, morphological and semantic analyzers. The editor facilitates manual shifting of words inside sentences to produce the order of sentence according to the target language. If a lemma has unique sense then it is automatically translated into LIS; otherwise word disambiguation is performed with the help of MultiWordNet and a correct synonym is chosen from database [38, 134].

4.2.3.3 Statistical Translation: Use Case

In Statistical translation the AEWLIS sentences are generated by using an automatic method based on mathematical and statistical algorithms. The translation system is trained with a large number of manually translated phrases (corpora). By using an existing knowledge-base the system automatically generates the translation for new input sentences with a sufficient precision. The Figure 4.9 shows the use case diagram of Statistical translation system.

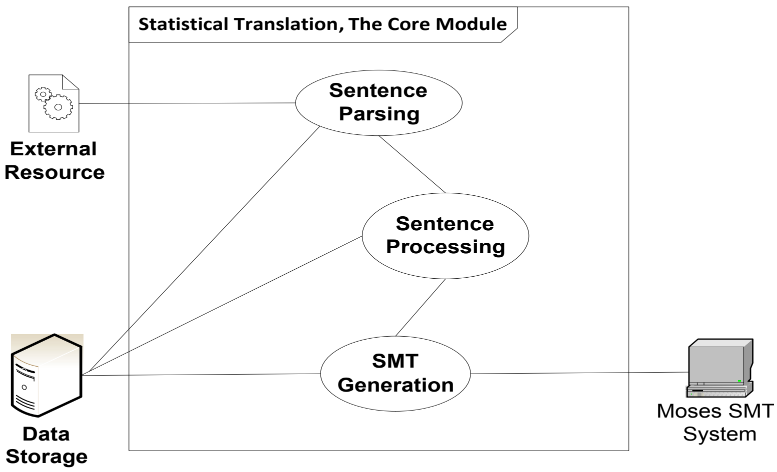


Figure 4.9. Use case: Statistical translation system from Italian text to Italian Sign Language

The statistical translation process is further divided in the following steps.

- Parsing: Performs syntactic and semantic shallow parsing on input sentences.
- Processing: The sentence is normalized to generate the syntactic structure of sentence more similar to the LIS sentence.
- SMT Generation: Produces the best possible translation according to the language model, translation model and a reordering model.

4.2.3.4 Rule-based Translation: Use Case

In Rule-based translation the grammatical analysis is performed in order to translate text in LIS. The Figure 4.10 shows the use case diagram of Rule-based translation process.

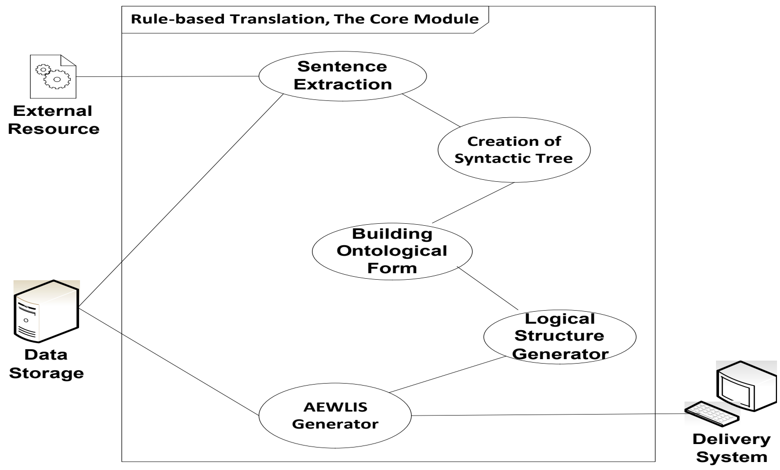


Figure 4.10. Use case: Rule-based translation system from Italian text to Italian Sign Language

In the first step, the input text is parsed into sentences. In the second step, the Syntactic Tree of an Italian Sentence is produced according to syntactic rules. In the next step the semantic interpretation is performed to generate the Ontological Structure which is further transformed into logical structure. Finally AEWLIS formalism is generated by using Combinatory Categorial Grammar.

4.2.3.5 Generating Virtual Actor Commands

The delivery system which is the third module of translation process is responsible to generate the LIS signs with the help of 3D virtual characters and provide end user services on web and mobile platforms. The resource allocation planner provides additional information for AEWLIS Communication Channels. Figure 4.11 describes the use case diagram of Virtual Actor commands generation process. The animation expert extracts AEWLIS sentences and inserts the timing information, sets the speed of Signs and produces the Virtual Actor commands. The Sequencer parses the time-line information to issue the right commands at the right moment.

The LIS interpretation is a process in which the Signs are produced by the ATLAS player. The ATLAS player receives Virtual Actor Commands, character model

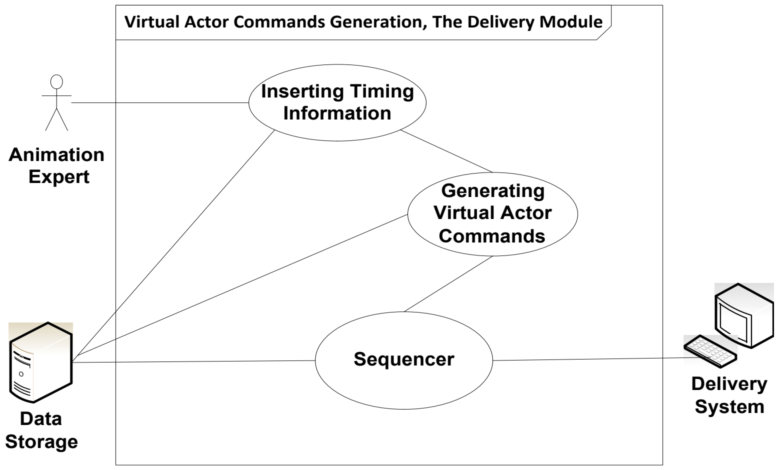


Figure 4.11. Use case: Generation of Virtual Actor commands from AEWLIS

and Signs as input while it provides Virtual Actor frame sequence and animation files as output. The frames are converted into videos by using Libavcodec which is audio, video codec library of FFmpeg. FFmpeg is the multimedia framework, able to decode, encode, transcode, mux, demux, stream, filter and play audio video files. The videos are stored in ATLAS Multimedia Archive [135]. The users can search any LIS Sign from database as described in Chapter 2. All business use cases regarding end users' communication with system are already described in Section 2.4.2.

4.3 Translation Workflows in jABC

Figure 4.12 shows the complete workflow for the automatic translation into Sign Language processes in jABC. This chapter concentrates on the Rule-based translation approach, which is represented by a macro SIB in the workflow model. The rules based translation SIB contains another SLG which is expanded in Figure 4.15. The Rule-based translation workflow model presented in Figure 4.15 contains further domain specific seven SIBS. Every SIB is an independent building block which can perform a specific functionality. The outcome of the Rule-based translation is described in the AEWLIS formalism which is provided as input to the LIS interpreter for signing the lemma of each LIS sign. A raw animation file is produced through the key frame animation technique which is played within the ATLAS player in order to produce virtual character-based motion frames. The frames are integrated together to form movies.

4.3.1 Description of the Entire SIB Library

The Rule-based translation workflows use some domain independent common SIBs from jABC library while some other SIBs are self implemented. Following subsections provide the details of SIBs already available and domain specific SIBs which are especially created for servification of Rule-based translation process.

4.3.1.1 Common SIBs Library

The jABC comes with a built in library having large collection of SIBs for common usage. Many frequently used SIBs which are domain-independent are useful for representing our own workflows. These common SIBs are grouped into different categories some of them are listed here:

- Basic SIBs used to deal with the execution context. `PutExpression`, `PutFile` and `RemoveElement` are some examples of Basic SIBs used to put and remove objects on the execution context.
- Collection SIBs deal with arrays and maps. `PutMap`, `PopElement` and `IterateElements` are some examples which facilitate storing information in arrays and maps along iteration in elements.
- Graph Model SIBs provide services to traverse SIB graph models. `AddModelBranch`, `ConnectSIBs` and `GetSIBBranches` are some examples of Graph Model SIBs.
- GUI SIBs are used for handling dialogues for users' interaction. `ShowMessageDialog`, `ShowEmailDialog` and `ShowBranchingDialog` are some examples of common dialogues available in the GUI SIBs library.
- IO SIBs are used to perform input/ output related tasks. `CopyFile`, `GetCurrentDirectory` and `UnzipFiles` are some examples used to copy files, explore directories and zip or unzip files.

The Rule-based translation workflows presented in Figure 4.12 and Figure 4.15 utilize following SIBs from jABC library either available under common SIBs category or already implemented for some other projects as domain independent SIBs:

  – `ShowBranchingDialog` is used to select the ingestion source for the input text. The execution will proceed with the branch for which the corresponding button on the dialog has been selected by the user.
  – `ForkSIB` is used to start either a Statistical translation or a Rule-based translation thread against one outgoing branch. Each thread runs in its own context.

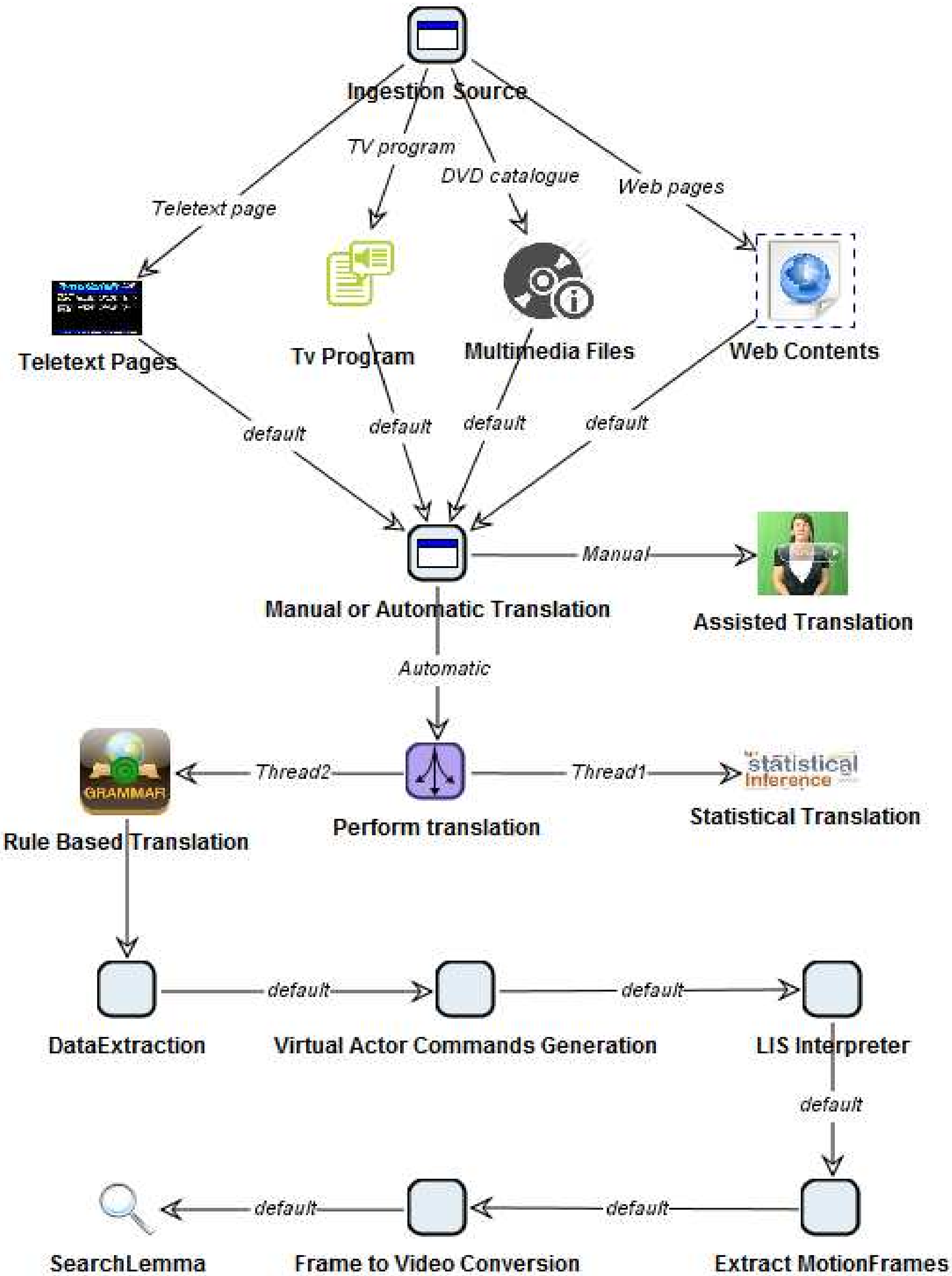


Figure 4.12. Complete workflow for the automatic translation into Sign Language processes

– `MacroSIB` contains another SLG which hides more detailed process. As described earlier, the Rule-based translation SIB is represented as MacroSIB in Figure 4.12 while the detailed process is described in Figure 4.15.

– `FtpUpload` is used to upload a file on the FTP Server. The results of Statistical and Rule-based translations are stored on a FTP server.

– `FtpDownload` SIB is used to extract AEWLIS sentences from the FTP server.

– `IterateElements` is used to iterate the elements of a collection array. The user can provide multiple sentences together for translation and IterateElements SIB is used to handle the input sentences one by one.

– `ShowMessageDialog` is a common GUI SIB which is used a number of times to show information. For example in the LIS dictionary the SIB is used to display the Sign properties, synonyms, other meanings and variations.

4.3.1.2 Domain Specific SIBs for Machine Translation of Languages

The SIBs found in the jABC library were helpful but not enough to represent complete translation workflows, so some domain specific SIBs are implemented which are described as following:

– `SearchSignbyId` searches a LIS Sign by using Sign ID.

– `SearchSignbyLemma` searches a LIS Sign against Italian Lemma.

– `RuleBasedTranslationHandler` extracts the sentences from input text.

– `RBTParser` extracts the relations and produces a syntactic tree.

– `RBTSemanticInterpreter` converts the input syntactic tree into ontological form called First Order Logic (FOL).

– `RBTGenerator` performs segmentation, lexicalization, simplification and produces Attribute-Value Matrix (AVM) form.

– `RBTSpatialAllocationPlanner` produces the extended AEWLIS syntactic tree.

– `Recovery` produces word by word translation if some error is occurred in Rule-based translation process.

– `ProduceAEWLISXML` creates a local XML file and stores the information received from `RBTSpatialAllocationPlanner` or `Recovery SIB` in it. It also provides the FTP server directory path to upload the AEWLIS sentences.

- `LISInterpreter` send Virtual Actor commands to Atlas player and generates motion frames.
- `FramestoVideoConversion` converts motion frames into LIS videos.
- `TeletextPages` searches text of Teletext Pages.
- `TvPrograms` searches all multimedia files of Tv program contents.
- `MultimediaFiles` searches all multimedia files of translated contents.
- `WebContents` searches all Web contents available at Server.

Table 4.2 lists all the SIBs implemented in jABC for translating Italian text into LIS. The SIBs are further categorized into four domains: SIBs concretely providing Rule-base translation, SIBs interacting with LIS dictionary, SIBs producing Signs from Virtual Actor commands and SIBs for selection of different multimedia contents. There are two major kinds of services:

- API-based Services: Most of these SIBs are programmed in java and jABC provides well defined interface to interact with them.
- SOAP Services: Provides a basic messaging framework upon which web service clients are built to access different required functions.

Table 4.2. Overview of the Italian text to Sign Language domain specific SIBs library.

Domain	API Based	SOAP Services [WSDL-based]

Rule-based Translation	`RuleBasedTranslationHandler`
	`RBTParser`
	`RBTSemanticInterpreter`
	`RBTGenerator`
	`RBTSpatialAllocationPlanner`
	`ProduceAEWLISXML`	

Dictionary-based	`Recovery`	`SearchSignbyId` `SearchSignbyLemma`
Signs Generation	`LISInterpreter` `FramestoVideoConversion`	

Remote Services		`TeletextPages`
		`TvPrograms`
		`MultimediaFiles`
		`WebContents`

4.3.1.3 Taxonomic classification

In the jABC framework the SIBs are organized by means of a taxonomy. The TaxonomyEditor plug-in is mainly used by the domain expert to classify the SIB libraries according to specific domain properties or characteristics. The TaxonomyEditor provides a user interface which initially shows available SIBs according to physical Java package structure and the domain expert renames and rearranges the SIBs as required [108, 111, 113]. The taxonomy is a tree representation of services in which intermediate nodes represents different groups or categories. Taxonomies are considered as simple forms of ontologies having subset relationship with descendent nodes. Figure 4.13 displays the taxonomy of automatic translation from Italian text to Sign Language where SIBs are organized into a simple tree structure.

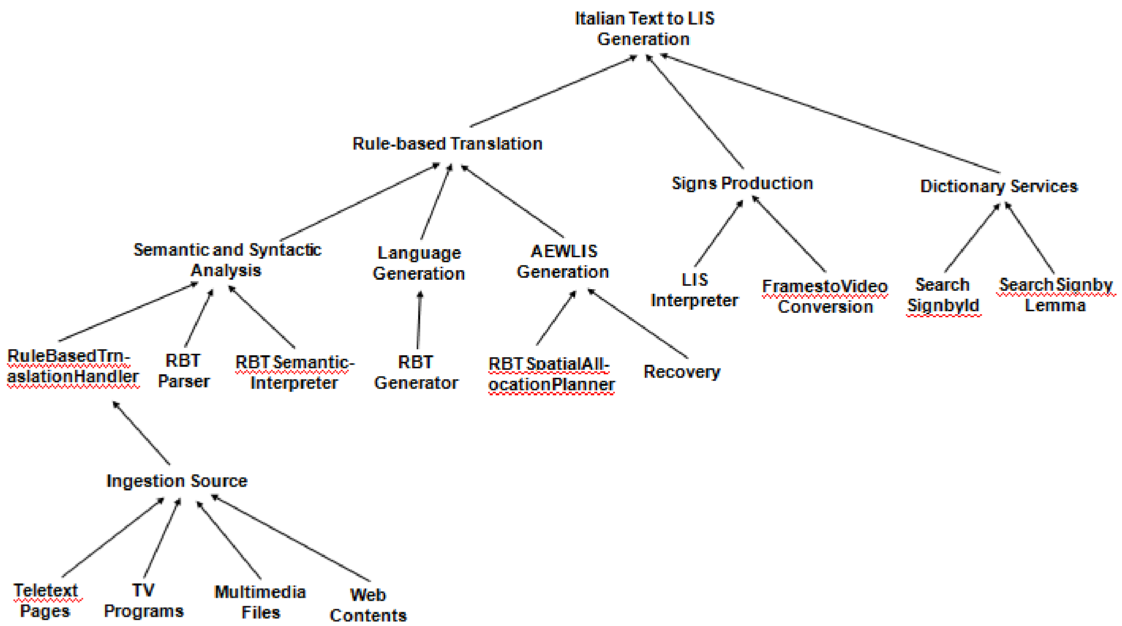


Figure 4.13. Service taxonomy of automatic translation from Italian text to Sign Language model

The domain specific services accept different kinds of input and also provide variety of output. The Semantic characterization of these input and output types facilitates error checking during workflows execution. The Figure 4.14 shows type taxonomy of automatic translation from Italian text to Sign Language workflow. The *Thing* is the most generic type. Teletext pages, Tv programs, multimedia files and web contents are semantically characterized as files. The sentences are further categorized into Italian sentences and AEWLIS. These type classifications helps to

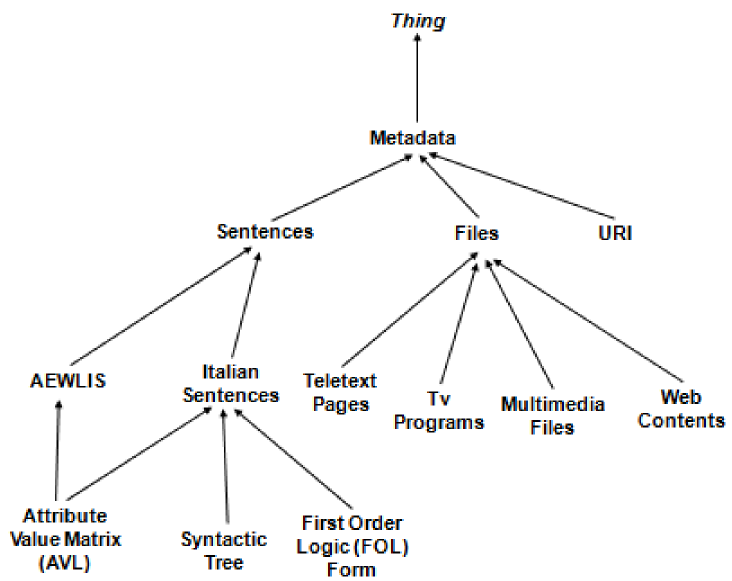


Figure 4.14. Type taxonomy of automatic translation from Italian text to Sign Language model

assess the compatibility/incompatibility of services, given that they are properly annotated in terms of the domain-specific type vocabulary [110].
Table 4.3 lists the domain specific services developed for automatic translation from Italian text to Sign Language along their semantic types. The set of input types shows the mandatory inputs required for successful running of the service and set of output types exhibits all possible outcomes. If an error occurs during execution of any service, an error message can be displayed by the ShowMessageDialog SIB or can be handled by `Recovery` SIB. In service oriented implementation of Rule-based translation process all the SIBs have well defined interface to interact with them which enhances their re-usability.

4.3.2 Rule-based Translation Workflows

The Rule-based Translation involves grammatical processing and it is performed by number of lightweight processes. The workflow starts from user input and after translation, it stores the results at users' local system and also uploads the translation on FTP server. The realization of Rule-based translation in the form of SLG is given in Figure 4.15. Each SIB in SLG performs a specific task which is described in the following subsections.

Table 4.3. Type Taxonomy in Rule-based translation from Italian text to Sign Language model.

Services	Input Types	Output Types
RuleBasedTranslation-Handler	Sentences	Sentences
RBTParser	Sentences	Syntactic tree
RBTSemnaticInterpreter	Syntactic tree	First Order Logic (FOL) form
RBTGenerator	FOL Form	String Attribute-Value Matrix (AVM)
RBTSpatialAllocation-Planner	Attribute-Value Matrix	AEWLIS
Recovery	Sentences	AEWLIS
ProduceAEWLISXML	AEWLIS, Data	URI
TeletextPages		File
TvPrograms		File
MultimediaFiles		File
WebContents		File
SearchSignbyId	Metadata	Metadata
SearchSignbyLemma	Metadata	Metadata
LISInterpreter	URI	Metadata
FramestoVideo-Conversion	URI	Metadata

4.3.2.1 Rule-Based Translation Handling

The `RuleBasedTranslationHandler` is the first SIB, which extracts the Text ID, Sentence ID, start time, duration time and sentences from user input XML string. User can input multiple sentences in a single request. Following is a standard XML request which is accepted by `RuleBasedTranslationHandler` SIB for parsing.

```
<?xml version="1.0"?>
  <text id="17">
    <sentence id="5" start_time="8.45" duration_time="1.425">
      Domani avremo pioggia al sud
    </sentence>
    <sentence id="6" start_time="9.875" duration_time="2.359">
        Sono previsti temporali sulla Liguria
    </sentence>
```

```
</text>
```

Three different java classes are implemented to handle the input. The XMLUtil class validates the input XML format, extracts different XML tags and their values. The SentenceVO class stores the sentenceID, content, start_time and duration_time. The class TextVO provides the public methods to extract the senetences and other information. The Vector built-in Java class implements a growable array of objects, which is used to iterate the elements of SentenceVO class. The IterateElements SIB receives collection of array sentences as input from RuleBasedTranslationHandler SIB and passes one by one sentence to RBTParser for further processing.

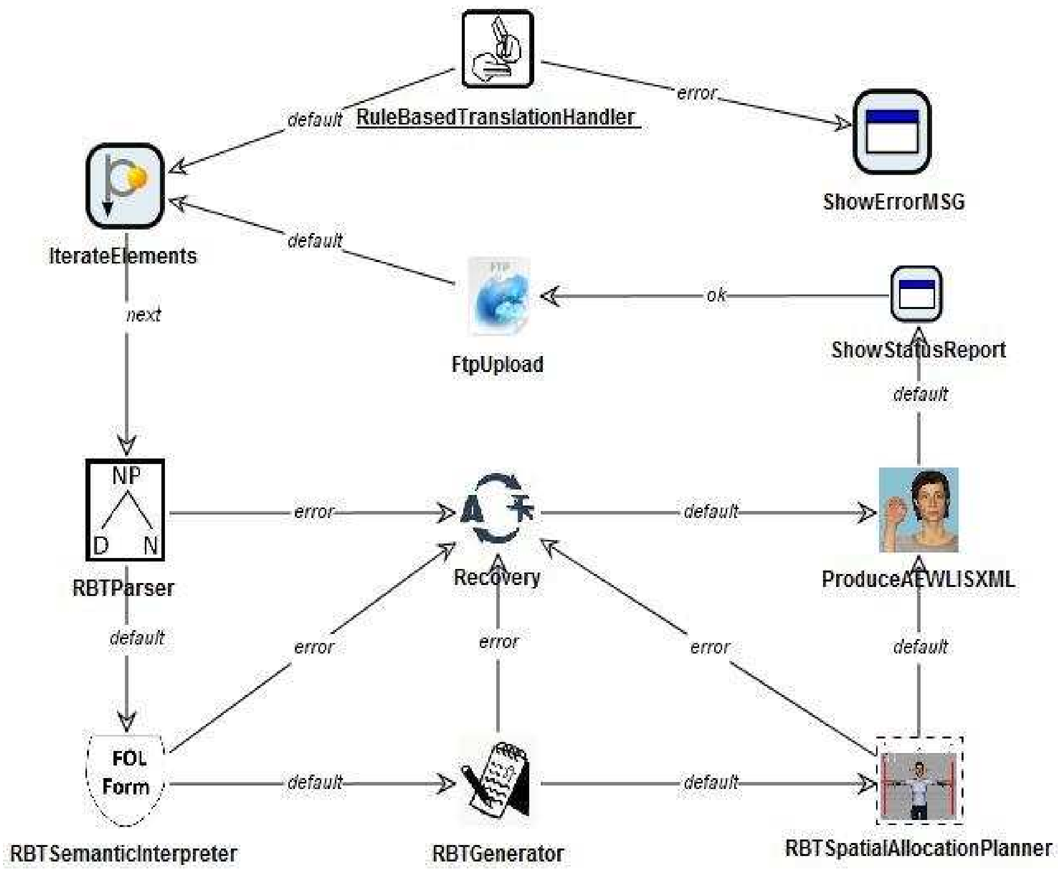


Figure 4.15. Rule-based translation realization as Service Logic Graph (SLG)

4.3.2.2 Syntactic Analysis

The parsing process is performed by the `RBTParser` SIB which produces a syntactic tree. The `RBTParser` extracts the input and performs the tokenization and morphological analysis. The tokenizer recognizes the words, possible proper names or identifiers and punctuation marks. The output of the tokenizer is a sequence of “token hypotheses” that belongs to a “tokenizer category”. The main tokenizer categories are defined as:

- GW: any standard word, excluding proper names
- NOMEP: proper names
- SIGLA: abbreviations, codes, initials, etc.
- SEGNOINTER : punctuation marks
- SPECIAL: other symbols
- NUMBER : numbers in digits
- DATE : dates (in a standard forms, such as 16/02/2014)
- PARAGRAPH-NUMB: chapter and paragraph numbers such as 3.1.3.
- YEAR: just for forms as ’14 (standing for 2014)

After tokenization the morphological analysis is performed by splitting standard words in pairs <root, suffix>, where all suffix possibilities are taken into account and those possibilities are matched with the dictionary. There are many common suffixes in every language, for example in English the suffix ‘ance’ exhibits the state or quality of something as used in ‘maintenance’. In case an input form is ambiguous, as ‘pesca’ is an Italian word, which has two senses, either it is used in sense of ‘fishing’ or it refer towards peach ‘fruit’, a Part Of Speech tagger chooses the most appropriate interpretation. In next step the `RBTParser` SIB extracts the relations among words and looks for chunks, i.e. subsequences of words usually related to nouns. To explain the concept, consider following sentence:

-In Italian: “Le immagini mostrano che pioverà in Roma”... (i)
-In English: “the images show that it will rain in Rome)”

In case of this example, we get “Le immagini” and “in Roma” chunks. After finding the chunks, verbal subcategorization is used for attaching them to the verbs. The weather forecast sentences are often structured in a way that the subject is absent. To represent the sentence in tree form, an empty "t" block is inserted in tree. After dealing with the unattached fragments the final syntatic tree is generated which is showed in Figure 4.16. In the syntactic tree the word "immagini" is also expanded further to show the results produced by tokenization process, morphological analysis and POS tagging. The SIB implementation is composed of two parts: the

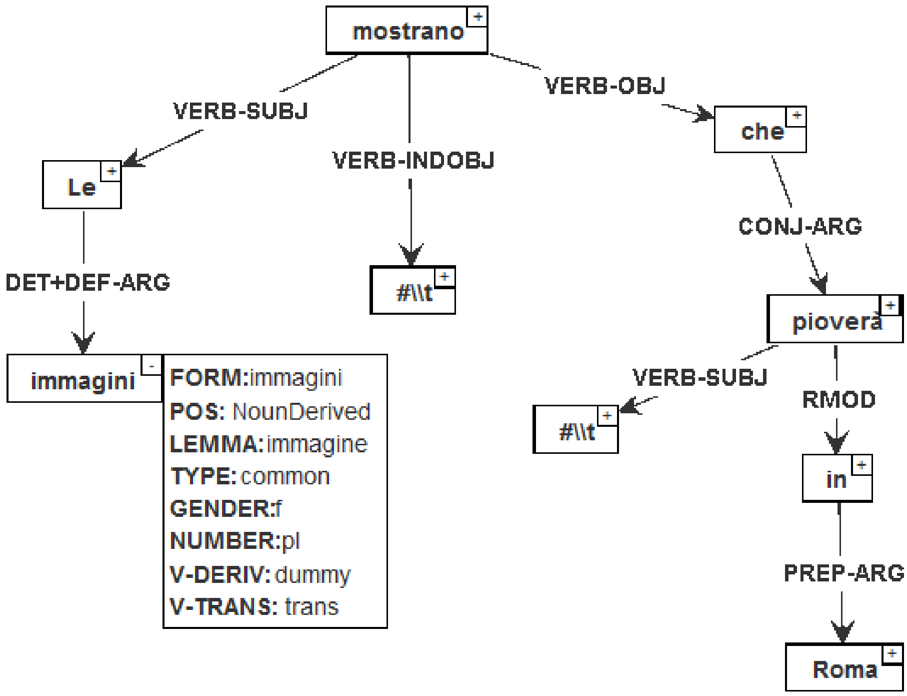


Figure 4.16. The syntactic structure of Italian sentence "Le immagini mostrano che pioverà in Roma"

SIB itself and its service adapters. A service adapter realizes the SIB's behavior for target platform and number of service adapters can be attached to a SIB, programmed in supported languages of target platform. The SIBS remain completely platform independent as platform specific implementations are separated from the SIBs description [108]. Typically, in Rule-based translation, the service adapter calls number of CLISP programming files containing thousands lines of codes that realize the actual service represented by the SIB and also perform necessary data conversion. CLISP includes an interpreter, a bytecode compiler, socket interface and strong internationalization support. The `RBTParser` sends the sentence to the CLISP local parsing server running in Linux virtual machine by using a bidirectional socket-based communication on the internal TCP port 3001. Following is the code snippet taken from `callServer()` method which opens the IP port and sends the input sentence for parsing to CLISP local server.

```
try{
    echoSocket = new Socket(ip, intPort);
    } catch (Exception e) {
        // If server for the parser does not exist.
        System.err.println("Don't know localhost. IP=" + ip + "
            intPort=" +intPort );
          }

try{
    out = new PrintWriter(echoSocket.getOutputStream(), true);
    in = new BufferedReader(new InputStreamReader(echoSocket.
        getInputStream()));
    } catch (IOException e) {
        System.err.println("Couldn't print Outstream to localhost");
        Thread.sleep(10);
        throw e;
        }
try{
     inputString = inputString.replaceAll("\n", "");
     out.println(inputString);
     buffer = in.readLine();
     outString += buffer;
     while (buffer != null) {
       // System.out.println(buffer);
           buffer = in.readLine();
                               if (buffer != null) {
                                   outString += buffer;
                                             }
         }
      } catch (IOException e) {
     System.err.println("Couldn't read Inputstream to localhost");
     Thread.sleep(10);
     throw e;
      }
```

The code also provides debugging information and displays proper exception messages. As described earlier the `RBTParser` SIB provides syntactic tree as output. If following sentence is provided as input to `RBTParser` SIB:

-In Italian: "Sono previsti temporali sulla Liguria" ... (ii)

The `RBTParser` produces following syntactic tree as output. ... (iii)

```
(((HEAD ((FORM |previsti|) (POSITION 2) (SYN ((LEMMA PREVEDERE) (CAT
   VERB) (TYPE MAIN) (MOOD PARTICIPLE) (TENSE PAST) (TRANS TRANS) (
   NUMBER PL) (GENDER M))) (LINK TOP-VERB) (SEM NIL))) (DEPENDENTS
   (((HEAD ((FORM |Sono|) (POSITION 1) (SYN ((LEMMA ESSERE) (CAT
   VERB) (TYPE AUX) (MOOD IND) (TENSE PRES) (TRANS INTRANS) (PERSON
   3) (NUMBER PL)))
```

The one fourth part of syntactic tree is shown in above paragraph (iii) to provide an idea that how Syntactic tree information is composed into String format.

4.3.2.3 Semantic Interpretation

The `RBTSemanticInterpreterr` SIB accepts the the syntactic tree as input and builds the ontological form by accessing the weather domain ontology. The ATLAS Meteo ontology is a formal, but partial, representation of the weather forecast domain in which some conventions have been adopted for ontology names:

- Instances are defined by £prefix.
- Concepts are defined by using ££as prefix.
- Relations and relation instances have a & prefix.
- Three special individual cases have a §prefix (the system: §myself, the user: §speaker and an indefinite individual: §indef-ref).
- Pseudo-concepts have a - prefix, which usually plays a bridging role in syntax.
- Datatypes have a $ prefix ($string and $sound).

The ontological forms are generated in two steps. The first one concerns the annotation of the syntactic tree with information extracted from a semantic dictionary, while the second step further converts it into First Order Logic (FOL) form. The actual annotation maps the word of the input sentence to the nodes in the ontology. some extra nodes are also added on the top of the tree for adding pragmatic information concerning the role of the sentence in a dialogue. Many steps in ontological form are related to internal organization of the ontology, but they do not affect the actual semantic contents. Before generating the FOL representation, the simplified logical form is achieved by removing irrelevant information. This representation includes all data required to 'understand' the meaning of the sentence, i.e. it talks about a

'showing' event whose agents are some 'images'; what is showed is a 'rain' whose location is an Italian middle region named 'Rome'. The final FOL representation is obtained by associating each concept with a unary predicate having the same name, and each relation with a binary predicate where all variables are quantified.
Following is the code snippet taken from service adapter implementation of `RBTSemanticInterpreter` SIB in which the outcome of the `RBTParser` SIB is provided as parameter and `mainRBTSemanticInterprete()` function of RBT class is called to forward the Syntactic tree to CLISP local server for further processing.

```
try{
 System.out.println("\n LightWeightServiceAdapter for
    rBTSemanticInterpreter Started:\n");
 RBT rbtobject=new RBT();

  // Calls main Semantic Interpreter function of Rule-based
     translation class.
  String S= RBT.mainRBTSemanticInterpreter((environment.evaluate(
     inputsentences).toString()));

  // Puts the response object on execution context.
  environment.getLocalContext().put(responseObjectInterpreter,S);
  System.out.println("\n Ended Response:\n");
 if(S!=null && !(S.isEmpty()))
       { System.out.println("\n Ended Response:\n"+ S);
              return Branches.DEFAULT;
       }
 else
              return Branches.ERROR;
} catch (Exception e) {
       // put the exception into the context
       e.printStackTrace();
       environment.getLocalContext().put("exception", e);
       // return error branch
       return Branches.ERROR;
       }
```

The `mainRBTSemanticInterpreter()` function calls `callServer()` method which sends the sentences to CLISP server on the internal TCP port 3002. The RBTSemanticInterpreter SIB provides following FOL form as output in case of sentence (ii) is provided as input. ... (iv)

```
((EXISTS (X1 X2 X3) ((Â£Â£DIALOGUE X1) (&HAS-DIAL-TOPIC X1 X2) (
   Â£Â£TO-FORESEE-1 X2) (&FORESEEN X2 X3) (Â£Â£STORM X3) (&FORESEER
   X2 Â§GENERIC-AG) (&HAS-SITUATION-LOCATION X2 Â£LIGURIA))))
```

4.3.2.4 Language Generation

The `RBTGenerator` SIB extracts and organizes the various pieces of information available in the FOL with the help of a "microplanner" called `SentenceDesigner`. The `SentenceDesigner` class splits the logical form into sets of predicates ('messages') that is the foundation for the lexicalization process. The `SentenceDesigner` class performs a number of operations described in following list.

List A:

- Segmentation: splits the FOL representation into a number of single messages
- Lexicalization: introduces pre-lexical nodes for each message and establishes the relations among them
- Simplification: removes the redundant information from each lexicalized message, e.g. repetitions and non-necessary pre-lexical nodes.

The example sentence is very simple, so it produces a single message. The SentenceDesigner produces a set of 'abstract' syntactic trees (one for each message), for the generation of LIS output. Some ontological nodes can be replaced by more "abstract" counterparts. It helps us for producing an output where the verbal form "pioverà" (it will rain) is replaced by the nominal form "pioggia" (rain). A first step towards the production of the output is the attachment of LEX nodes (lexicalization step), that is intermediate data, linking the concept to the actual LIS lemma. In second phase (called Realization) the syntactic trees are encoded in the standard format used by OpenCCG (hybrid logics) to derive the surface sentence. The final output of the linguistic generator is expressed in the form of an Attribute-Value Matrix (AVM). In AVM each tree is characterized by a head and a set of dependents where the special form (#\#) represents the empty tree.
The `RBTGenerator` SIB receives the semantic interpretation results and sends those set of FOL formulas to the CLISP server on the internal TCP port 3003. The AVM, which is the output of the linguistic generator against input sentence (i), is shown as following: ... (v)

```
((head ((form |mostrare|)
        (position 2)
        (idSign 03087)
```

```
            (syn ((lemma mostrare) (cat verb) (number sing) (arg-ref 0) (hand 2)))
            (link TOP-VERB)
            (sem (semtype VnA))
            (phon ((dir -)))))
 (dependents (((head ((form immagine_mostrare)
                   (position 1)
                    (idSign 02845)
                    (syn ((lemma immagine_mostrare) (cat noun) (number sing)-
                        (arg-ref 0) (hand 2)))
                    (link VERB-SUBJ)
                    (sem (semtype VnA))
                    (phon ((dir -)))))
                (dependents ((#\#))))
              (#\#)
              ((head ((form pioggia)
                        (position 4)
                        (idSign 1767)
                        (syn ((lemma pioggia) (cat noun) (number sing)-
                            (arg-ref 0) (hand 2)))
                        (link verb-OBJ)
                        (sem (semtype VnA))
                        (phon ((dir -)))))
             (dependents (((head ((form Roma)
                            (position 3)
                            (idSign 2810)
                            (syn ((lemma Roma) (cat noun) (number sing)-
                              (arg-ref 0) (hand 1)))
                            (link NOUN-RMOD)
                            (sem (semtype VnA))
                            (phon ((dir -)))))
                        (dependents ((#\#)))))
                       (#\#))))))
```

4.3.2.5 Creation of AEWLIS Formalism

The `RBTGenerator` SIB produces a kind of “linguistic” representation of the LIS sentence, in other words that is a form of “written” LIS which is different from commonly used Sign Language notation systems described in section 2.2. This written representation provides no information about how to move the hands in order to produce the "animation" of the virtual character. The "written" LIS is provided to the `RBTSpatialAllocationPlanner` SIB which produces the "augmented written" LIS. The process roughly corresponds to annotating with phonologic information, a step towards "reading" it. The position of the hands is encoded in terms of coordinates inside a tri-dimensional "box" placed in front of the signing character. The algorithm assigns the sentence to the whole signing space and then, starting from the root of the AEWLIS parse tree, it assigns portions of the space to the various subtrees, according to the syntactic/semantic relations between the root and the roots of the subtrees [136]. Finally `RBTSpatialAllocationPlanner` SIB produces the extended AEWLIS syntactic tree by sending the AVL form to the CLISP spatial allocation planner on the internal TCP port 3004. The output of the `RBTSpatialAllocationPlanner` SIB is given here when the AVM form of sentence (ii) is provided as input. ... (vi)

```
("(
<?xml version=\"1.0\"?>
<ALEAOutput>
  <newSentence text="???" italianText="???" writtenLISSentence="???"
      lemmaNumber="???">
    <newLemma lemma=" LIGURIA" idAtlasSign=" 2810">
       <handsNumber> 1 </handsNumber>
          <signSpatialLocation> 0.45 0.0 0.0 </signSpatialLocation>
    </newLemma>
    <newLemma lemma=" PREVISIONE" idAtlasSign=" 2678">
       <handsNumber> 1 </handsNumber>
          <signSpatialLocation> 0.0 0.0 0.0 </signSpatialLocation>
    </newLemma>
    <newLemma lemma=" TEMPORALE" idAtlasSign=" 2697">
       <handsNumber> 2 </handsNumber>
          <signSpatialLocation> -0.6 0.0 0.0 </signSpatialLocation>
    </newLemma>
  </newSentence>
</ALEAOutput>
)")
```

4.3.2.6 Recovery

The `RBTParser` SIB, `RBTSemanticInterpreter` SIB, `RBTRBTGenerator` SIB and `RBTSpatialAllocationPlanner` works in a pipeline. The output of the each SIB is used as an the input of next SIB. If any of these SIB could not produce output and gets some kind of error then the error branch will lead towards `RBTRecovery` SIB as

shown in figure 4.15. The `RBTRecovery` SIB performs word by word translation by the help of LIS dictionary. The `RBTRecovery` SIB produces the extended AEWLIS syntactic tree in same format as produced by `RBTSpatialAllocationPlanner`. The `RBTRecovery` SIB takes input sentence directly from execution context and dispatches the request to CLISP Recovery server on TCP port 3005.
The `ProduceAEWLISXML` SIB receives the output either from `RBTSpatialAllocation-Planner` SIB, if the Rule-based translation is performed without any error otherwise from `RBTSpatialAllocationPlanner` SIB and creates an XML file at local system to save the output translation. The `ProduceAEWLISXML` SIB also provides the path of FTP server directory to upload the results on FTP server. The `ShowMessageDialog` SIB a common GUI SIB which displays the path of local XML file stored in the system. The *FTPUpload* SIB uploads the final AEWLIS XML files on ATLAS server which are further used to produce virtual character based animations.

4.3.3 Hierarchical Modeling in jABC

Hierarchical modeling facilitates in the reduction of complexity in the global model. A `Macro` SIB encapsulates another SLG which can be assumed as parent child relationship. A `Macro` SIB also contains parameters and branches but these parameters and branches belong to entire model which is associated with it [108]. The model presented in Figure 4.15 can be simplified by introducing the `RBTGrammarProcessing` macro SIB. The `RBTGrammarProcessing` SIB is realized with another model which contains the SIB instances that are concretely performing grammatical analysis. Figure 4.17 behaves as parent model and contains the SIBs related to input, output and recovery functions. Figure 4.18 contains `RBTParser`, `RBTSemanticInterpreter`, `RBTGenerator` and `RBTSpatialAllocationPlanner` SIBs it behaves as sub-model and performs all grammatical translation.

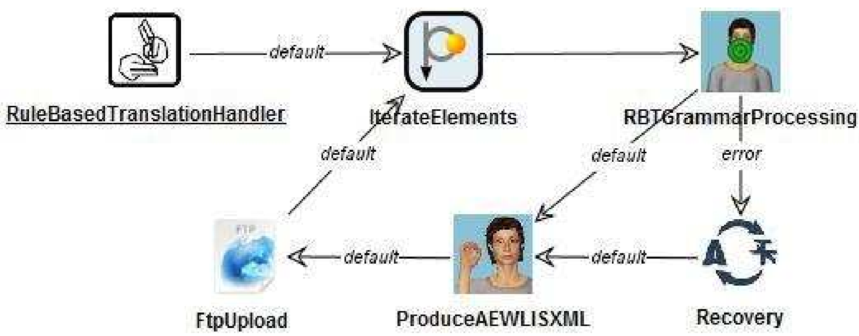


Figure 4.17. Rule-based translation hierarchical modeling in jABC

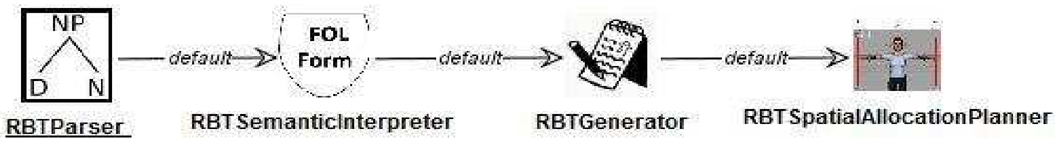


Figure 4.18. Rule-based translation grammar processing SIBs

In Figure 4.18 the `RBTParser` SIB is underlined which shows that when submodel will be called the execution will be started from here. In jABC, if multiple SIBs have some common behavior like in case of error the error-handling routine will be called then a SIB can declare its outgoing branch as model branch and all subsequent SIBs are able to map that model branch. The `RBTParser` SIB declares error branch as model branch and all other three SIBs outgoing error branches are mapped with model branch. The `RBTSpatialAllocationPlanner` SIB exports its default branch as model branch. In Figure 4.17, default and error branches are added in the `RBTGrammarProcessing` macro SIB according to the flow of submodel. In case of error in the submodel, the `RBTGrammarProcessing` SIB leads towards `Recovery` SIB. The models presented in Figure 4.17 and 4.18 are collaboratively providing functionality exactly equal to the model presented in Figure 4.15.

4.3.4 Workflows for Transfer-based Machine Translation

Our translation approach can help to model direct and Transfer-based machine translation. The SIBs enhances the re-usability of the system by providing option to organize SIBs in a different way to produce a different model. The direct and Transfer-based machine translation will be discussed here in more depth to analyze that which SIBs can be re-utilized, what are alternative workflows and what is missing if we want to use interlingua system especially for Transfer-based machine translation systems. In direct translation there are two approaches, one is simple approach while other is advanced that is more closer to Transfer-based syntactic approach. There are some challenges involved in direct translation. We need a dictionary which covers all cross lingual phenomena. As we discussed in Chapter 3 there are almost 67 thousand Italian lemmas exist in MultiWordNet while in our LIS Dictionary there are almost 3 thousand signs. Finding language pairs and establishing relationships among them is a challenging task. In simple translation, only low level processing regarding tokenization is performed. In advanced approach morphological processing is performed along identification and handling of syntactic ambiguity.
The syntactic Transfer-based approach requires a large tree-bank to transfer syntactic structures between the source and target languages. The syntactic Transfer-based

approach is realized in figure 4.19.

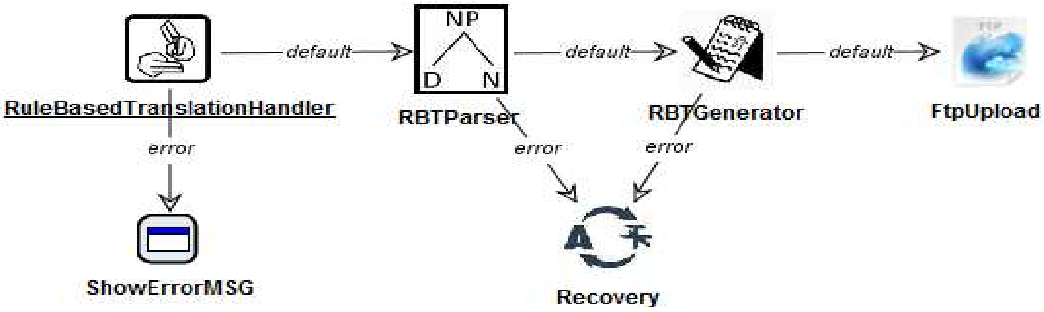


Figure 4.19. Syntactic Transfer-based Machine Translation(MT) system

The syntactic Transfer-based Machine translation workflow will call the `RBTParser` SIB, which performs following tasks defined in list:

List B:

- Tokenization
- Morphological analysis
- Part Of Speech tagging
- Extract chunks
- Verbal subcategorization
- Producing syntactic tree

The details of each task is already described in Section 4.3.2. The output of the `RBTParser` SIB is a Syntactic tree in String format which can not be passed directly to `RBTGenerator` SIB. The workflow presented in Figure 4.19 is incomplete and `RBTGenerator` SIB will produce the error due to inappropriate input String and the system will go in recovery mode where it will perform word by word translation. For successful implementation of the Transfer-based system the RBTGenerator CLISP module requires some changes to enable it to accept syntactic trees directly as input, furthermore extensive research is required to identify that how output can be generated to support other Sign writing notation systems e.g HamNoSys, Sutton Sign Writing and Stokoe Notation Systems. It is already discussed in Chapter 2 that Italian text to Sign Language system transfers the source language into an intermediate language which is entirely different from Sign Languages writing notation systems discussed in Section 2.2.
For syntactic Transfer-based machine translation systems the functions performed

on the source language are reversely implemented in target language. Target language (TL) parsers, TL taggers and TL morphological generators are applied to produce the translation. The translation from English to ASL by machine(TEAM) project used the syntactic Transfer-based machine translation approach [137]. In English to ASL translation system the input text was analyzed with a TAG parser. The English lexical entries were searched in bilingual lexicon to identify the corresponding ASL signs. The tag generator for ASL assembled the ASL tag trees into a complete sentence. Due to emphasis on syntactic analysis the translation system can handle many syntactic divergences found in both languages [24].
In existing system, Syntactic Transfer-based approach can be adapted with some variations. The translation can be performed either word by word after syntactic analysis or the `RBTGenerator` could proceed towards normal flow and can generate the AEWLIS sentences which are already supported by ATLAS player. If the word by word translation is performed, it will offer comparatively richer results than the simple direct approach due to the process performed in syntactic analysis like word disambiguation, noun relations identification and separation of root and suffix in input text.

4.3.4.1 Alternative Workflow for Transfer-based Machine Translation

In an ideal case, the `RBTParser` SIB can be decomposed into number of SIBs, each one performing a specific functionality independently from each other. Currently the system has strong dependency and the whole functionality described in List B is realized as single SIB. Similarly the `RBTGenerator` can be decomposed into `RBTSentenceDesigner` SIB which will perform the tasks described in List A and `RBTRealizer` will encode the syntactic trees in OpenCCG format. Figure 4.20 shows alternative workflow for Syntactic Transfer-based Machine Translation(MT) systems.

The workflow described in Figure 4.20 contains two rectangles which indicates that the main workflow can also be realized as hierarchical model with parent child relationship. The SIBs defined in rectangular box A and rectangular box B can be realized as another SLGs by replacing both boxes with Macro SIBs and assigning submodals to them on the same pattern described in Figure 4.17. The outgoing error branch of `tokenization` is declared as model branch and other error branches are mapped with this model branch. The alternative workflow for Syntactic Transfer-based Machine Translation(MT) systems has two assumptions to work properly, which provides necessary information that how the approach can be enhanced in future to support other automatic Machine Translation(MT) systems.

- `RBTParser` and `RBTGenerators` SIBs functionality is properly decomposed into number of lightweight processes.

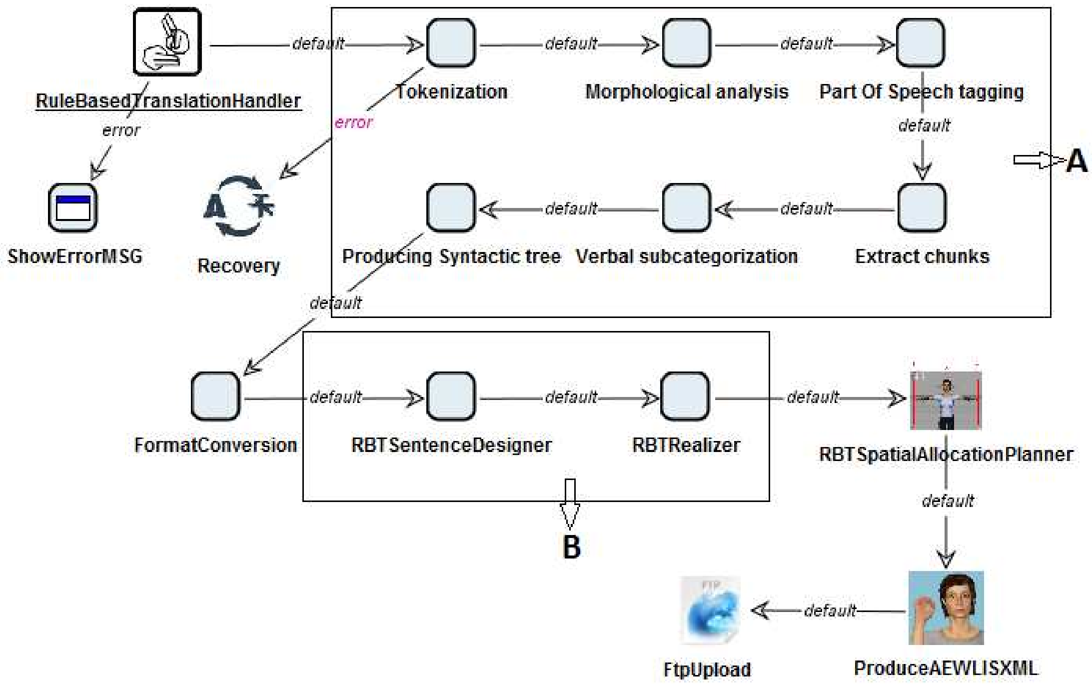


Figure 4.20. Alternative workflow for Syntactic Transfer-based Machine Translation(MT) systems

- `FormatConversion` SIB is implemented in such a way that its output is accepted by `RBTSentenceDesigner` SIB.

4.3.5 Extending Service Oriented Translation System

To extend this system for Semantic Transfer-based approach or for some other Sign languages, here some necessary components are described which play important role in Rule-based translation systems.

- Morphological analyzer and generator for source and target language.
- Parser for source and target language which analyzes source sentences and compose target language sentences.
- A translator and bilingual dictionary to translate words in target language.
- Source Grammer and Lexicon which can generate syntactic tree by considering all available lexicon.
- Source and target mapping rules to map the grammatical functions on domain concepts in interlingua and vice versa in target language.

- A domain ontology which in our case is weather forecast domain.

Now there are some design decisions in which various analysis and transformation methods can be used to obtain translation results. If the target language is other than Italian Sign Language, then at-least two module require necessary changes, 1st the language generation module which maps the rules of the source language into target language and 2nd bilingual dictionary because after grammatical analysis the dictionary will be used to lookup the word pairs. One scenario we have already discussed in details in Figure 4.20, another alternative workflow can be easily drawn for semantic Transfer-based machine translation. If the source language is not changed then `RuleBasedTranslationHandler`, `RBTParser` and `RBTSemanticInterpreter` SIBs can be easily substituted in new system to perform semantic and syntactic analysis on source language. The integration of Rule-based translation assits the users to design their own workfows according to their requirements and maximizes the option to use existing building blocks to fasten the process of development. The re-usable building blocks of XMDD approach can be realized as step forward towards solving the complexity of Natural Language Processing(NLP) problems.

4.4 Advantages of Servification

In automatic translation of Italian text to Sign Language, the Rule-based translation process before converting it into Service oriented process was acting like a black box which was getting input from ingestion source (i.e takes from the Orchestrator database) an Italian sentence and was producing corresponding AEWLIS sentence. Although Figure 4.5 provides the impression that a modular approach is adopted, in reality, all the modules were tightly coupled which severely limits the re-usability of the system.
As we described earlier, in Rule-based translation approach the semantic and syntactic analysis is performed, now theoretically the parser and semantic interpreter may be used for translating text into a different target language i.e., Swiss-Italian Sign Language, but practically the modules of the existing system were highly dependent on each other and had not well defined interface to interact with them. XMDD as advocated by T. Margaria and B. Steffen, is an approach which encourages the development of loosely coupled service independent building blocks [41].

"Service-oriented computing radically alters the way business processes are modeled, realized, and maintained. Domain-specific services virtualize complex functions of the underlying business applications so that they can be loosely coupled to form transorganizational processes. This level of abstraction fosters agility and lessens traditional provider dependence".

The Rule-based translation process is transformed into service oriented one and the SIBs presented in section 4.3.1 are implemented as platform independent components. Each SIB is realized as a service which has well defined interface. This kind of implementation offers the users the flexibility to re-use the SIBs for other languages and every SIB can be upgraded without effecting the functionality of other SIBs.

4.4.1 Performance and Efficiency

An important aspect of translation is the performance of the system, which is related to response time, throughput, or timeliness. Performance metrics are the best way to evaluate the effectiveness and efficiency of the system. Usability measures the quality of the end-users experience which is directly related to the time consumed in task completion. Time-on-task becomes particularly important where the tasks are performed repeatedly by the users.
To evaluate the performance of the preexisting translation system and Service oriented one, the following five Italian sentences related to weather forecast domain were executed 10 times on both systems to get the average execution time of each SIB involved in the translation process:

- S1. “Sono previsti temporali sulla Liguria” (5 words).
- S2. “Pioggia al Sud e neve al Nord” (7 words).
- S3. “Nel corso della giornata le nubi aumenteranno al Centro” (9 words).
- S4. “Annuvolamenti sparsi sulle regioni Settentrionali ed alta Toscana con possibili piogge; prevalentemente soleggiato sulle restanti regioni” (16 words).
- S5. “Molto gradevoli su tutte le regioni, con minime in diminuzione” (10 words).

The sentences having different word lengths are randomly selected from Teletext pages. The Table 4.4 shows the average time (in seconds) consumed by individual SIBs and total Mean time consumed by both systems in translation process for each sentence. The first row of the table shows the total number of words in each sentence. The first column of the table shows the five core grammatical processing SIBs and then total run-time of each SIB of the existing system(E) and Service oriented(S.O) one. The time consumed by the `Recovery` SIB is discussed separately at the end of the table. The alternative rows are highlighted to evidence the time difference consumed in both approaches.

Table 4.4. Performance of existing and Service oriented Rule-based translation systems.

Sentences		S1	S2	S3	S4	S5
No of Words		5	7	9	16	10
RBTParser	E	5.33	5.29	6.94	5.58	5.74
	S.O	5.3	5.3	6.79	5.62	5.94
RBTSemantic-Interpreter	E	5.15	5.2	5.21	5.21	5.11
	S.O	5.16	5.18	5.14	5.14	5.27
RBTGenerator	E	6.27	5.12	5.15	5.01	5.04
	S.O	5.65	5.12	5.27	5.05	5.03
RBTSpatialAllocationPlanner	E	5.06	5.04	5.02	5.01	5.02
	S.O	5.28	5.06	5.1	5.12	5.07
ProduceAEWLIS-XML	E	0.03	0.02	0.01	0.01	0.05
	S.O	0.01	0.01	0.01	0.01	0.02
Total Time	**E**	**21.84**	**20.67**	**22.33**	**20.82**	**20.96**
Total Time	**S.O**	**21.4**	**20.67**	**22.31**	**20.94**	**21.33**
Recovery	E	5.31	5.2	6.39	5.48	5.8
	S.O	5.27	5.25	6.36	5.51	5.86

The Table 4.4 shows a slight difference in the performance of existing and Service oriented translation systems. Some Service oriented technologies in fact cost a lot of overhead, but in our case, either the Service oriented system performed equally or even better in case of shorter sentences having less than 10 words.
The Table 4.5 compares the efficiency of both approaches in terms of time consumed by core grammatical SIBs and shows the percentage of total execution time consumed by each SIB during the translation process. The table clearly indicates that the Service oriented rule based translation system always performed better in `RBTGenerator` SIB and `ProduceAEWLISXML` SIB. Basically, `RBTGenerator` SIB has two parts: one is `SentenceDesigner` and other is `Realizer`, whose details were already discussed in Section 4.3.2.4. The `Realizer` part is completely implemented in Java code, similarly `ProduceAEWLISXML` SIB functionality is also implemented in java while the remaining SIBs call CLISP modules from Linux server to perform the functionality. The Service oriented Rule-based translation system comparatively performed better where the SIBs are based on Java code.

The hypothesis that Service oriented Rule-based translation performs better if the the SIB is only calling some Java code is further developed by running `RuleBasedTranslationHandler` SIB ten times in both systems. The Figure 4.21 shows the time performance SLG of `RuleBasedTranslationHandler` SIB in which

Table 4.5. Efficiency of Service oriented Rule-based translation system.

	Existing System	Service Oriented System
RBTParser	27.05%	27.1%
RBTSemanticInterpreter	24.3%	24.29%
RBTGenerator	24.93%	24.50%
RBTSpatialAllocationPlanner	23.61%	24.05%
ProduceAEWLIS-XML	0.11%	0.06%

`StartTimeMeasurement` and `StopTimeMeasurement` (two common SIBs) are used to calculate the consumed task. The SLGs described in Figure 4.21 consumed 70 milliseconds in average in 10 executions while the same code when called from Eclipse IDE consumed 166 milliseconds in average. Although a set of data in which different five sentences are executed ten times in both translation systems is very small, it is evident from the performance tables that at least the Service oriented implementation has not decreased the efficiency of the system, rather enhanced it in some cases.

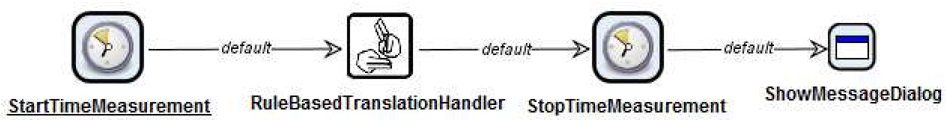


Figure 4.21. Time performance SLG for `RuleBasedTranslationHandler` SIB in jABC

4.4.2 Homogeneous Interface for Heterogeneous World

The automatic translation into Sign Language involves the integration of heterogeneous resources. Different partners in ATLAS project worked on different modules and some of them evolved independently of each other. There are many issues related to information transfer mechanisms. The systems usually communicate through application programming interface(API) but in many cases the API's are not provided or may not guarantee the compatibility of two sets of systems for communication. The other way of system communication is using a Middleware which is computer software that enables two separate software components, systems, processes or applications to exchange information. It is difficult to address multiple required properties such as platform independence, better performance, adequate level of security in a single Middleware. If any Middleware addresses these properties at some level then it may has issues regarding tightly coupled interfaces.

Incompatible data formats also become problem when heterogeneous tools are involved in the development, especially when some of them are legacy systems. In jABC all components will have a uniform appearance (as SIBs) and with XMDD, even complex heterogeneous workflows can be easily changed or extended at the graphical level [34]. In XMDD, the abstraction from technical details of services through virtualization concepts enables the integration of heterogeneous resources into a homogeneous environment [110].

4.4.3 Better Maintainability

Continuously evolving systems require a delicate balance of maintaining essential parts of the product, yet at the same time modifying erroneous or no longer adequate features and adding totally new functionality [35]. Consequently, updates of single components may be problematic, as it could affect the correct functionality of the overall system. The deployment of a complex system like in case of automatic translation into Sign Language, on a heterogeneous distributed platform is typically very complex, where maintenance and upgrading also becomes time consuming and expensive. Integration testing and interoperability are major cost factors and risks in a system implementation and deployment.
The jABC provides automatic deployment and maintenance support, system-level testing, regression testing and provide 'intelligent' feedback in case of late problems discovery or errors [34]. In 'one thing approach', the hierarchical model is automatically maintained whenever changes arise. The horizontal consistency is guaranteed by maintaining the global perspective throughout the refinement process down to the code level. The jABC comprises a high-level type checker, two model checkers, a model synthesizer, a compiler for SLGs, an interpreter, and a view generator. The model synthesizer, the model checkers and the type checker take care of the consistency and compatibility of the system [107].
The process of translation from Italian written text to LIS is tranformed into service oriented paradigm. This paradigm addresses the haterogenity challenges of linguistic translation modules and integrate them into translation workflows. These workflows improve the flexibility to entertain languages other than Italian. In addition offering translation modules into loosely coupled components will improve the re-usability of the system.

Chapter 5

Usability and Learnability of Virtual Character Based Interfaces

In this chapter a virtual character based interactive interface is introduced which concentrates on localization of interfaces by providing verbal help in the regional language through life-like characters. A comparative study of virtual character based interactive interface and traditional (textual) interface has been conducted. The results show that semi literate users found former interface more useful, motivating and encouraging for e-participation. The purpose of evaluating different interfaces is to enhance the e-participation of low literate users in the developing countries. No doubt, ICT has opened significant vast areas and tools for learning and development, but literacy is a big obstacle for accessing these opportunities. Heavy usage of text on everything from menus to documents means that low literate users are not able to access functions and services implemented on most computer software. Most computer applications pose an accessibility barrier to those who cannot read properly and fluently.

The chapter is organized into five sections. Section 5.1 discusses the role of ICT in bridging the digital divide, provide global statistics on languages used in web-sites and defines the literacy. Section 5.2 provides literature review on user interfaces for functionally illiterate users. Section 5.3 analyzes different software development methodologies and compare them with User Centered Design. In Section 5.4, first case study about evaluating Virtual Character-based interfaces is discussed while second case study about evaluating four different interfaces is presented in Chapter 6. Section 5.5 discusses the results of first case study in terms of effectiveness, efficiency and satisfaction related to the usability of textual and Virtual Character-based interfaces.

5.1 Bridging the Digital Divide

Digital divide refers towards the gap between individuals or societies in accessing ICT facilities. At the end of 2013 almost 2.7 billion (39%) people of the world population are using internet in which the mobile broadband subscription is near to 2 billion, while still 4.4 billion (61%) are not connected with internet. There are 6.8 billion mobile subscriptions till end of 2013, which is almost equal to the overall population of the world, but till the end of 2012 only 50% of mobile users have 3G network coverage. Figure 5.1 shows the number of internet users worldwide in 2013. Although the lowest numbers of users are in Africa, the internet users penetration has doubled over the past four years in Africa [4]. There are 39 countries (2.4 billion inhabitants) which are marked as least connected countries and kept under the category of developing countries. In least connected countries the internet access is limited, hardly ever high-speed, very expensive, and used by only a small percentage of the population.

A weak road network, non-availability of electrical power or a lack of fixed tele-

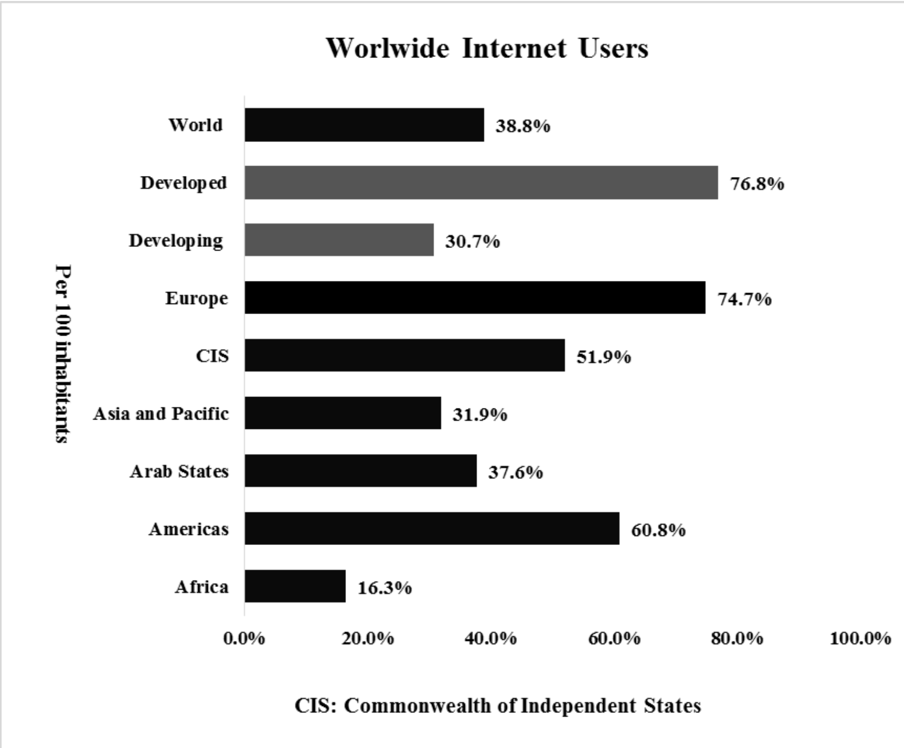


Figure 5.1. Individuals using the internet, by region and level of development, 2013 [4].

phone lines discriminate against rural areas in reaping the full potential of ICT. Powerful uses of technology in people require building a vision, developing human

resources, integrating the community, establishing innovating learning environments, and building good follow-up and evaluation procedures. Economic, organizational, multilingual, socio-cultural factors, development of corresponding multimedia interfaces, ensuring re-usability and ease of porting to other languages are some barriers in bridging ICT. In order to make use of information technology, other resources must be available, e.g. infrastructure, job opportunities or access to credit or health services.
There are three critical factors which cause digital divide in developing countries [138]:

1. Connectivity: Whether the infrastructure is deployed which enables the users to access ICT services?

2. Affordability: Whether the services are affordable by all users?

3. Capability: Whether the user have technical abilities, language and literacy skills to be truly benefited from ICT service?

The objective behind the evaluation of different interfaces is to enhance the capability of low literacy users by concentrating on the third factor. With over 774 million people classified as illiterate or semi-literate (see section 1.1), there is a need to better understand how to design user interfaces to allow this group of individuals access to online services. Extensive research is required to explore the reason behind the success or failure of ICT based learning programs. Due to less knowledge, expertise, or human resources trained in these methodologies, there is resistance to change or upgrade to new technologies.
ICT provides developing nations with an unprecedented opportunity to meet vital development goals such as poverty reduction, provision of basic health care, and improving access to quality education. The International Telecommunication Union (ITU) has described the three stages model for ICT development process if a country is transforming into an information society: First stage is known as ICT readiness, which reflects the provision of network infrastructure and access to ICTs. Second stage is ICT intensity, which reflects the level of ICT usage in society. The third stage is ICT outcome, which shows the results of efficient and effective usage of ICT services [4].
The direct transfer of technology from developed countries to developing countries has not been successful in most cases, primarily because of the mismatch between the intended environment for which the technology was designed, and the ground realities of the environments in which they were deployed. There is a need to better understand and choose the appropriate technologies best suited for everyone. Accessibility is more important than availability, as mere availability of technology is not enough and cannot ensure accessibility for all. Technological interfaces for the

general public need to be different from those built for technical people, as level of understanding of educated and non-educated people is substantially different. There is need to provide the information in the local language in order to make it accessible for semi literate people.

5.1.1 Web Contents and Accessibility

Web systems are increasingly important in one's day to day life be it seeking employment through online applications, applying and receiving government services, seeking medical information, or simply becoming aware of current events. Figure 5.2 shows the web contents available in different languages. More than 55% web contents are available in English and it is clearly the dominant language, while less than 0.1% of the contents are available either in Swahili or in Kinyarwanda language[1]. People who can't read English are unable to access the information.

In order to design the interfaces for a variety of users coming from different ethnic groups, having different learning and speaking capabilities demand to think in broader aspect in order to design the interfaces suitable for all. Multilingual interactive assistance through a virtual character on interface can enhance the capability

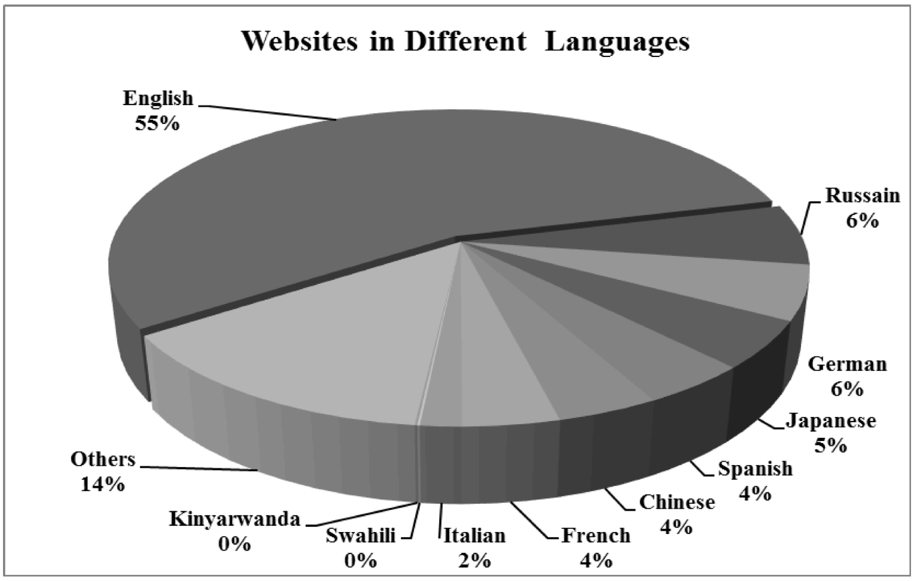


Figure 5.2. Website contents in different languages.

[1]Usage of content languages for websites. Web Technology Surveys, August (2013) http://w3techs.com/technologies/overview/content_language/all

of illiterate or semi literate users to perform required tasks quickly. Virtual characters with capability of voice help in multiple languages, exploiting examples during on-line forms submission and gestures can help users to perform tasks on their own. Interactive interface design, if implemented correctly by keeping view the potential end users and their requirements may eliminate the need of static textual help. The virtual character based interface reported in this chapter is helpful for semi literate users to assist them in their activities i.e.; completing an online application form. People centered interfaces appropriate for everyone need to be designed, developed and deployed on public websites.

5.1.2 Defining Literacy

In English the word 'literate' means knowledgeable, educated or familiar with literature. Since nineteenth century literacy has referred towards reading and writing simple text but in the mid of 20th century the researchers have given considerable attention to define literacy in broader aspect and presented four discrete understandings of literacy [139].

1. Literacy as set of skills: The most common understanding is the cognitive skills of reading and writing, independent from the context and background. The numeracy or basic arithmetic skills are considered supplementary. In a broader context, the literacy is being expanded to include some further skills, for example 'information literacy', 'visual literacy', 'media literacy' and 'scientific literacy'.

2. Literacy as applied, practiced and situated: Acknowledging the limitations of the skill based approach, functional literacy initially emphasized the impact of literacy on socio-economic development. Functional literacy emphasize on teaching literacy as universal set of skills which are neutral and independent from social context.

3. Literacy as a learning process: Literacy is considered as action and continuous process as a person learns, he becomes literate.

4. Literacy as text. This approach particularly emphasizes on passages of text which are consumed or produced by a learner and is heavily dependent on language learning.

The definitions of literacy and semi-literacy are complex. Many countries have their own definitions. In China, e.g., the definition of literacy varies in urban and rural areas. In urban areas a person is considered literate if he has learned minimum of 2,000 characters while in rural areas literacy refer to the knowledge of 1,500 characters minimum. The United Nations Educational, Scientific and Cultural Organization

(UNESCO) defined literacy in the Education for All 2000 Assessment as "Literacy is the ability to read and write with understanding a simple statement related to one's daily life. It involves a continuum of reading and writing skills, and often also includes some basic arithmetic skills (numeracy)" [140]. UNESCO emphasized on formal primary education, and insisted on the fact that it is critical for all to acquire minimum reading, writing and calculating skills.
Purely illiterate persons cannot read or write in any capacity, for all practical purposes. In contrast, a functionally illiterate person can read and possibly write simple sentences with a limited vocabulary, but cannot read or write well enough to deal with the everyday requirements of life in their own society. A more abstract form of semi-literacy is defined as "A person is semi-literate, if he is poorly educated and unable to read or write with ease or fluency" [141]. In this dissertation, the two terms, semi literate and functionally illiterate are replaceable with each other and refer towards inability in information literacy.

5.2 User Interfaces for Functionally Illiterate Users: Literature Review

Designing interfaces for low literate populations involves particular challenges: the needs, expectations, experiences and environment of low literate populations in developing nations are different from those of a designer. Online technologies intended for such groups should be developed in continuous partnership with primary users and include a thorough investigation into their worlds, lives, relationships and concerns [142, 143].
Interfaces are one of the most important aspects of usability as it is communication and interaction medium through which users interact with the system. Users who are not able to read a text can be aided with pictures explaining a text or they can be helped by hearing someone read the text to them. An interesting idea is to use pictograms as reading aids for low literacy users. When the users navigate through the text, the pictures are displayed as a tool-tip to explain the concept. In a pilot study conducted by Goetze, a separate dictionary of pictures is maintained to provide the explanation of each picture [144]. Language-independent icon-based interfaces support users in composing their queries by means of selecting icons. The system is helpful for low literacy users in retrieving information from the web [145]. Huenerfauth suggested some guidelines to design interfaces helpful for low literate users in order to decrease the dependency on other humans. These guidelines include simple text captions to help the user to select the appropriate choice, displaying instructions in a sequence with sequence number and continuous oral assistance [146].

5.2.1 Minimal Use of Text

Researchers proposed minimal use of text with some suggesting text-free user interfaces [147]. The basic features of what we call a text-free user interface are simple to understand: liberal use of graphics and photographs for visual information, and voice for providing information in-place of text. Text free user interfaces are used for employment search in a community of low literacy Indian slums. The results showed that users were generally successful in finding employment opportunities using test-free interfaces, while they were unable to do so on textual interfaces [16]. Microsoft Research India tested three different systems including text-based, spoken dialogue and rich multimedia for mobile money transfer [148]. Results confirmed that text free designs are strongly preferred over text-based designs. Task completion rates for the rich multimedia user interface were better and faster, and less assistance was required on the spoken dialogue system.
Beenish Chaudry studied four graphical user interface (GUI) widgets and three different cross-page navigation styles in mobile applications with low literate users. The purpose of the study was to compare the usability of different non-text based widgets (such as radio buttons, check boxes and icons) on low-literacy populations [149]. Easy Texting messages is a mobile application that allows low literate users to listen to SMS and compose text messages by augmenting words with touch-initiated text-to-speech support, icons for frequent phrases and by re-using words from previous messages [150]. SymAB is a symbol-based address book for the semi-literate mobile users which uses symbols to replace current text based storage and retrieval. It includes symbol-based search functionality and call distribution based address book [151].

5.2.2 Voice Annotation and Speech Feedback

IBM India's research laboratory made a case of spoken web [29]. The spoken web is a network of interlinked voice based sites. The users can create their own websites by simply talking and providing the contents to a voice application. The generated sites will be in local language and it was possible to link the newly created site with other sites using hyper-links. It was designed to work with voice as the primary user interface modality and required the ordinary telephone instrument to make calls. Plauché et al. evaluated a low-cost, scalable speech-driven application to provide interactive agricultural information to low literacy villagers in southern India [152]. In spoken dialogue systems the key challenges include the development of speaker-independent automatic speech recognition (ASR) systems and text-to-speech (TTS) systems that function reliably in regional languages in the developing world [153].

Sherwani et al. designed, developed and evaluated voice-based interfaces to provide health information to low-literacy community health workers in Pakistan [13]. JUPITER is a conversational interface that allows inexperienced users to obtain worldwide weather forecast information over the telephone using spoken dialogue [154]. Taoufik et al. found that more interactive vocal instructions (like a digital assistant that can assist the users step by step) are needed for semi-literate users as they are usually reactive not proactive [155].
VideoKheti is a mobile system using speech, graphics, and touch interaction for low-literate farmers in rural India. It helps farmers to find and watch agricultural videos in their own language and dialect. The results show that farmers could use VideoKheti, but their success still greatly depended on their education level [156]. The video mail interface was designed to assess the feasibility whether asynchronous peer-to-peer communication capabilities of email can be made accessible to low literate populations in the developing world [157].
Automated Teller Machine(ATM) is a common banking service that enables the clients to perform financial transactions independently. An effective interface for ATM can be designed by implementing some design guidelines such as by using large screens, bigger fonts, foreground-background contrasts, simple language; place text close to the associated button, questions close to the answer options; use icons in addition to text, icons should be concrete, should show how to do things (animation) and speech is suitable for establishing personal contact, should neither be too fast, nor too slow, should not be used for personal information [158]. Birnie et al. designed pillar automated teller machine, which is a low cost device designed for all user groups including inexperienced and potentially low literate or illiterate users [159]. Different prototypes were designed to investigate the effective mobile interface for illiterate or low literacy users. Through an iterative process of questioning, probing, participatory design and continuous testing of prototypes, the requirements of low literacy users about favorite phone features were identified. The user centered design techniques were adopted to incorporate these features in final prototype of the phone interface [160].

5.2.3 Use of Virtual Characters

The idea to use virtual characters is not new and different researchers have utilized them in variety of ways. In this section, the role of virtual character in different scenarios will be analyzed.
Usage of virtual characters enhances the motivation of low literacy users and can be considered as an alternative to human assistance [161]. Virtual characters are source of encouragement for users to remain in the learning environment, engage them in learning activities and make them feel more comfortable. The implementation of a

realistic avatar with human-like gestures enhances the sense of human presence in users by simulating face-to-face communication [162].

5.2.3.1 Virtual Characters in Tutoring Systems

Many computer learning programs use virtual characters to help students in learning. Victor is one of them, which is integrated in online tutoring system and capable of providing basic feedback about the learning process by talking to the students, gesturing and displaying facial expressions [163]. Easy with Eve is a functional affective tutoring system which detects student emotions through facial expression analysis, and can display emotions herself. Eve also delivers lectures and reads teaching contents through realistic lip-synching and gestures [164]. The Betty's brain system presents a teachable agent called Betty's brain that combines learning by teaching with self-regulated learning feedback to promote deep learning and understanding in science domains. The students query their agent to monitor their learning and problem solving behavior. This motivates the students to learn more so they can teach their agent to perform better [165].

5.2.3.2 Virtual Characters as Agents

Fast growth of e-business requires process automation to deal with multiple customers at the same time, which requires hiring of more customer service agents to help users in on line shopping, which is costly. Virtual characters are getting more attention as an alternative to real human characters, as they are cost effective with dynamic features and can be customized and trained according to the requirements. The virtual sales clerk agent for a virtual movie store is helping customers in a 3D on-line shop. Based on a product knowledge base, the agent communicates with the customer through text based dialogs and leads the customer through the virtual world using gestures [166]. Intelligent virtual agents for contact center automation [167], context sensitive virtual sales agents [168] and virtual humans for assisted health care [169] are some examples where avatars are helping users in a virtual environment.

5.2.3.3 Virtual Characters in Games

Virtual characters are mostly used in games with the purpose to enrich the game playing experience by showing an engaging and consistent interactive behavior. Previous research on games involve body and facial motion of virtual characters, while recent innovations in game industry include alternative input modalities for games to provide an enhanced, more immersive user experience. Gaze control has recently been explored as an input modality in games [170]. Location-aware virtual characters [171], facial expressions, gestures and posture emotions in virtual characters

[172, 173] realistic and believable virtual humans [174, 175] and virtual characters with memory [176] are some other active areas of research in gaming.

5.2.3.4 Virtual Characters as Facilitators

Virtual characters are used in different applications for different roles, in order to perform a variety of tasks. As discussed in chapter 2, the sign language dictionary provides translation of Italian words into Italian sign language using virtual character animations [37]. The animated signs performed by virtual character are helpful for deaf and low literacy users to access and understand the information on the web. Text-to-sign language dictionary linked to Multiwordnet, provides sign language animations helpful for semi-literate or deaf to improve the vocabulary [38, 39, 40]. Designing interfaces for low literate people resembles designing for a cognitively challenged community, since some of their cognitive abilities may be less developed than those of literate and required extra assistance to complete their task [177].
Similarly in another example, a life-like character plays the role of an accompanying agent which provides continuous assistance to museum visitors. The virtual character follows the direction of visitors and helps them in different ways like how to reach a certain exhibit, and yielding explanations [178]. A group of older adults with low literacy in northern California benefited from the advice and encouragement of a virtual friend. Individuals who participated in an exercise program guided by the virtual adviser had an eight-fold increase in walking compared with those who did not [179].
Literature review reveals that most of the work in this area is application dependent, in which the usage of the application is optimized by providing a broad set of helping tools. However, research to date on user interfaces for semi-literate users remains scant and usage of standardized usability metrics to evaluate user performance are seldom. Virtual characters were used as role player mostly in virtual environments but rarely interfaces are designed to help illiterate or semi literate users in their task completion. Virtual characters discussed in this section are mostly for those who are computer literate. Virtual characters can pass additional instructions in regional language to help naive users to work independently [180].

5.3 Analysis of Different Design Methodologies

A software development methodology refers to a framework that is used to structure, plan, organize, analyze and control the process of developing an information system. The intent of a methodology is to formalize the process which will be repeatable for subsequent projects. A wide variety of such frameworks evolved in last four decades and each methodology is best suited for some specific project as all of them have their own strengths and weaknesses. In the following subsections, different major software development approaches are discussed.

5.3.1 Software Development Life Cycle(SDLC)

There are different models under SDLC category. Following subsections provide an overview of Waterfall Model, Prototyping, Incremental Development, Spiral Model and Rapid Application Development (RAD).

5.3.1.1 Waterfall Model

During 1970 the generic approach of software development was based on code and fix strategy. To overcome the problems produced by code and fix, Winston Royce introduced the waterfall model in 1970 for managing large software developments [181]. In waterfall model the project is divided into set of phases (requirements, design, implementation, verification and maintenance) which are sequential. Each phase consists of different activities and deliverables that should be completed before the beginning of next phase. In waterfall model the traditional engineering approach is applied to software development. The waterfall model also introduced iterations around the phases if changes in the requirements occur, but practically those were ignored [182].
The waterfall methodology suffers from serious limitations including no early feedback (prototyping), high cost for unclear user requirements, its inflexibility to visit different phases according to changed requirements, produces excessive documentation, very late problem discovery and delayed system testing. The approach is effective in projects where the requirements and objectives are clear in start, projects having no immediate deadlines and users are fully knowledgeable.

5.3.1.2 Prototyping

The Prototyping system is like an imaginary system (incomplete version of the software being developed), the users can either point out the missing features in existing system or can describe their vision about new system. In prototyping, the developers do not force users to read excessive documentation for understanding software functionalities. Rather they are provided with a working model of the system. Prototyping is not a complete system development methodology. Rather it focuses on sub-part of more traditional development methodologies by dividing complex projects into smaller segments and facilitating more ease of change in them [183]. It is an iterative process in which the prototypes are refined according to users' feedback. User involvement is an integral part of the development process. Prototyping gradually lowers the cost of analysis, design and programming. Users also feel more satisfaction due to early involvement. The early view of the system in prototyping helps in true understanding of business problem rather starting a process to solve an entirely misunderstood problem. The model is especially useful for unclear user requirements, comparing different design solutions or investigating performance of the system. In prototyping proper documentation is usually ignored which results on implementing more crucial requirements and some less important requirements are missed. If prototypes are not properly checked and verified or prepared by unskilled designers then the resulting system will be less effective. Project management and project control become more difficult with prototyped systems [184].

5.3.1.3 Incremental Development

The basic idea is to divide the complex project into sub-parts and a series of mini waterfalls are performed for each sub-part to entertain more ease of change before proceeding to the next phase. The initial system concept, requirements and analysis are defined by waterfall model and then iterative prototyping is used to produce the final product. The incremental development process helps in early risk analysis and necessary documentation is also prepared. Gradual implementation helps in rectifying issues raised in different stages and the stakeholders can be kept well informed about progress of the project. In incremental development initial product delivery is faster and it is generally easier to test and debug the increments.
In incremental development the resulting cost could be increased due to cost of organization and management. The Incremental model itself is not a complete software development life cycle, so all SDLC activities should be performed at each increment. Problems in system architecture could arise while adding new functions which were unexpected in initial prototype.

5.3.1.4 Spiral Model

The spiral model is a meta model which is used by other models developed in 1988 by Larry Boehm, who combined the advantages of the top-down and bottom-up approaches [185]. The model concentrates on risk analysis and checks the project feasibility in each iteration. Four activities are performed in each iteration around the spiral.

1. Determine objectives, alternatives and constraints.
2. Evaluate alternatives, identify, resolve risks.
3. Develop, verify next level product.
4. Plan next phase.

The spiral model incorporates waterfall, incremental model and prototyping and helps the user to select the best as software iteration based on project risk but the selection of right methodology is difficult task while iterating around the spiral. The spiral model is highly customizable where six invariants are also introduced by Larry Boehm in 2000 [186] so it is quite complex and limits its re-usability. There are no specific conditions to terminate a cycle and there are chances that projects may exceed from budget. The complexity of model sometimes results in implementation of waterfall model.

5.3.1.5 Rapid Application Development (RAD)

James Martin introduced the term RAD in 1991 to describe a software development process where the objective was to boost the delivery of high quality system at very low cost [187]. The systems are developed quickly by using iterative prototyping, also by using computerized tools like Computer Aided Software Engineering(CASE) tools, Database Management Systems (DBMS), fourth generation languages and object oriented techniques. Project management involves prioritizing the the tasks in development by defining some time constraints. Necessary documentation required for system extension is maintained and end user involvement is imperative in RAD. The four phases of RAD include (i) requirements planning phase which basically combines the system planning and system analysis phases of SDLC (ii) user design phase in which RAD team uses Joint Application Development (JAD) techniques and CASE tools to translate user needs into working system (iii) construction phase in which programming, unit testing and integration is performed and (iv) cutover phase resembles with SDLC system implementation phase.
In RAD model more speed and lower cost may compromise the quality of the systems. Rapid programming may create some serious flaws and its difficult to reuse the system. Due to the GUI interface, usually clients assume the system completion before time so actual effort remain unnoticed. There is also probability that complex features will be pushed for future implementation to gain early success which may create serious problems in later stages.

5.3.2 Agile Software Development

In 2011, 15 representatives from Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development and Pragmatic Programming discussed lightweight development methods and published a manifesto for agile software development. In the agile manifesto, four items were given more value over others.

1. Individuals and interactions over processes and tools

2. Working software over comprehensive documentation

3. Customer collaboration over contract negotiation

4. Responding to change over following a plan

Agile methods stress the unforgiving honesty of working code and the effectiveness of people working together with goodwill. Using people effectively achieves maneuverability, speed, and cost savings. If the team members work together they can provide better outcome than working alone which wastes a lot of time in understanding documents [188].
There are 12 principles behind agile manifesto.

1. Highest priority is customer satisfaction through faster and continuous software delivery

2. Welcome changes through life cycle of software development

3. Frequent/incremental delivery of software preferably in short period of time

4. Stakeholders and developers must work together daily

5. Empower motivated individuals

6. Promote face to face communication within teams

7. Working software is the principal measure of progress

8. Promote sustainable development with a constant pace

9. Continuous improvement of good design

10. Simplicity, the art of maximizing the amount of work not done, is essential

11. Concentrate on developing self-organizing teams

12. Regular team assessments to become more effective

Unlike waterfall model, limited planning is required to start system development in Agile model. Unclear user requirements could result in poor outcome and there is lack of emphasis on necessary documentation in Agile methods.
Agile is the umbrella term for a family of software development methodologies, including Scrum, eXtreme Programming, Mode Driven Development and Lean Development.

5.3.2.1 Scrum

Scrum is an iterative and incremental agile software development method as opposed to traditional, sequential approach. Scrum is set of principles and practices that helps teams in preparing products in short cycles, enabling fast feedback, continual improvement, and rapid adaptation to change. Scrum can be applied to any project having aggressive deadlines or expected changes in user requirements. Scrum adopts an empirical approach by assuming that user requirements could be misunderstood or could change with passage of time, so it empowers the team to respond those changes quickly. In Scrum, the iterations are typically called sprints, which are 1 to 4 weeks long.
Product backlog is prioritized feature in Scrum which is used to keep track of existing and desired features. Scrum is facilitated by a Scrum Master who is accountable to remove the roadblocks and to ensure that the Scrum process is implemented in letter and spirit and product goals are met successfully. Daily Scrum meetings require substantial resources and for the the Scrum master it is difficult to implement plan, structure and organize a project without clear definition of user requirements. It's good for small projects as it works well within small teams.

5.3.2.2 Extreme Programming (XP)

Extreme Programming focuses on the code and keeping the production environment stable, where Scurm is focusing on project management. Rather than planning, analyzing, and designing for the far-flung future, XP programmers perform these activities but little throughout development [189]. XP describes four basic activities: coding, testing, listening, and designing for software development. Scurm teams do not allow changes in sprints while XP willingly complies changes in development. The key XP practices are as following.

- Planning: Make a rough plan in start and refine it with passage of time.
- Pair programming: Two programmers at the same time will be involved in programming activity at single computer.
- Small releases: The working system will be deployed as soon as possible before implementing the whole system and small increments will be added regularly.
- Test driven development: For user stories, the developers write tests and then business decision is taken to implement the critical ones.
- Refactoring: The design of the system evolves through transformations in which code is improved without changing functionality.

- Continuous integration: The new code is regularly integrated several times in a day after unit testing.

- On-site customer: The customer regularly works with team.

XP is code-centric rather design centric and complex projects are hard to implement in increments, as a detailed plan is required before starting coding. Lack of documentation and less customers availability results in poor outcome. In most projects XP is used with other models like Scrum.

5.3.2.3 Model Driven Development (MDD)

Models help in understanding complex systems and provide potential solutions through abstraction [106]. Models are used in multiple ways to define characteristics of a system or changes, their impact or reasoning about specific properties. The major advantage is that models are designed using concepts and that they are less dependent on technological details and closer to the problem domain, while the actual programs are automatically generated from these models. Paperwork and documentation can not be eliminated entirely, but the model can act as a map to navigate through the system which helps in understanding the complex system with less text. In start, applying automation to software development was limited and only diagramming support was provided or some code skeletons were generated which was not a substantive difference. We can benefit fully from MDD if the programs are fully automatically generated from models and the models can be verified and tested on computers.
Model-driven development focuses on creating and exploiting domain models and bringing software development closer to conceptual design, so that architecture of a software solution could be understood by someone unfamiliar with programming concepts. In terms of speed MDD drastically decreases development time of simple projects, with little customization, since graphical interfaces can be faster in making code than writing code by hand. MDD can be a better design solution if relevant CASE tools are available but how much models are flexible depends on the tool in which they are designed.
Extreme model-driven development (XMDD) is relatively new software development paradigm which continuously involves customers/application experts throughout software development life cycle. XMDD differs radically from classical software development, in particular when it comes to heterogeneous, cross-organizational systems which must adapt to rapidly changing market requirements. XMDD puts models into the center of the design activity, becoming the first class entities of the global system design process rather just using models as means of specification [190].

5.3.2.4 Lean Development

Lean is a production practice that considers the expenditure of resources for any goal other than the creation of value for the end customer to be wasteful, and thus a target for elimination. Working from the perspective of the customer who consumes a product or service, value is defined as any action or process for which a customer is ready to pay. Lean thinking can be summarized in five principles as following [191]:

- Precisely specify value by specific product
- Identify the value stream for each product
- Make value flow without interruptions
- Let the customer pull value from the producer
- Pursue perfection .

Lean software development (LSD) is a translation of lean manufacturing and lean IT principles and practices to the software development domain, adapted from the Toyota Production System. Mary and Tom Poppendieck applied lean techniques from an industrial setting to software development [192]. LSD can be summarized in seven principles.

1. Eliminate waste: The first lean principle considers everything as waste which does not add value to the customers while UCD focuses on the product's potential users from the very beginning and checks at each step of the way that product design is according to the users' requirements.

2. Amplify learning: Iterations with re-factoring, improving the design as the system develops have been found to be one of the most effective ways to generate knowledge. Short learning cycles helps in amplifying learning.

3. Decide as late as possible: LSD encourages the concurrent design approach, which defers decisions as late as possible. Concurrent design reduces the number of high-stake constraints, gives a breadth-first approach to high-stakes decisions, making it more likely that they will be made correctly and dramatically decreases the cost escalation factor for most changes.

4. Deliver as fast as possible: Delivering as fast as possible recommends a good framing practice to achieve competitive advantage. Users also like rapid delivery. Software development schedules, software pull systems (short time boxes), information radiators (visual control), decreasing queuing time, steady service and more than all the cost analysis of delay helps in boosting software delivery.

5. Empower the team: Let the team make its own commitments. Intrinsic motivation requires a feeling of belonging, a feeling of safety, a sense of competence, and sense of progress.

6. Built-in-integrity : Integrity is achieved through excellent, detailed information flow [193]. Software development integrity is two dimensional: perceived integrity and conceptual integrity. Perceived integrity means that the totality of the product achieves a balance of function, usability, reliability, and economy that delights customers. Conceptual integrity means that the system's central concepts work together as a smooth, cohesive whole.

7. See the whole: A system consists of interdependent and interacting parts joined by a purpose. A system is not just the sum of its parts, it is the product of their interactions. The best parts do not necessarily make the best system; the ability of a system to achieve its purpose depends on how well the parts work together, not just how well they perform individually.

5.3.3 User Centered Design

User-Centered Design (UCD) is a methodology that focuses on user characteristics, usability goals, environment, tasks, and workflows in the design of an interface. UCD follows a series of well-defined methods and techniques for analysis, design, and evaluation of user interfaces [194]. The goal of UCD is to optimize user's experience and produce products which have a high degree of usability. UCD is a methodology while usability is an outcome of UCD practices. There are four important UCD principles.

- A clear understanding of user and task requirements.
- Incorporating user feedback to refine requirements and design.
- Active involvement of user to evaluate designs.
- Integrating user centered design with other development activities.

A typical UCD methodology has an analysis phase, design phase, implementation phase and deployment phase. There are six basic steps in implementing UCD methodology: (i) Get to know the users (ii) Analyze user tasks and goals (iii) Establish usability requirements (iv) Prototype some design ideas (v) Usability test and (vi) Repeat as necessary. The biggest cost benefit that UCD can provide is by more accurately defining users' requirements. The chief difference from other product design philosophies is that user-centered design tries to optimize the product around how users can, want, or need to use the product, rather than forcing the users to change their behavior to accommodate the product. Table 5.1 provides the major characteristics of Waterfall model, Agile and UCD methodologies.

Table 5.1. Major characteristics of Waterfall, Agile and UCD Methodologies.

Waterfall	Agile	UCD
• 3-6 months releases • Requirements definition in start • Sequential process • Used heavily pre-Web • Delays testing to end	• "XP" Extreme Programming • Reduce the cost of change • 30 day releases • Includes users involvement • Test as you go along	• Users as center of the focus • Iterative by nature • Rapid (UI) design prototyping • User testing often

In the comparative study of the virtual character based interactive interface and the traditional (textual) interface, the UCD methodology is adopted in which all development proceeds with the user as the center of focus and it covers all aspects of end user experience including users' needs, goals, motivation, triggers, obstacles, limitations, tasks, activities, behaviors, geography, environment, work and language.

5.4 Case Study 1: Evaluating Virtual Character Based Interfaces

Success of e-services is dependent on e-participation and availability to the common person. Many countries are offering different on-line services to their people; internship program started by Rwandan development board Republic of Rwanda is one such step. The youth internship program provides opportunities for students and other young people to gain valuable work experience. This also eases the transition between school and professional life or to get the job first time. It also contributes significantly to improve employability whilst at the same time help addressing the human resource constraints.

It was a two-phases study in which 100 semi literate users were selected from Republic of Rwanda by using a convenience sampling method technique [195] and divided randomly into two equal groups. Each group was composed of 50 semi literate users including 40 men and 10 women whose age was between 21 and 40 years. It is difficult to get access to women in Rwandan society, so their participation in the experiment was less as compared to men. The hypotheses that which kind of interface (original or with assistance through virtual character) enhances the usability and learnability of low literate users are being evaluated.
Weak analysis of end user capabilities discourages the e-participation of low literate community in the society. The author believes that abilities of low literate people should be taken into account while designing the interfaces for the broader target community, especially in the case of countrywide programs. For true understanding of the problem and analysis, the first group was instructed to use the original interface and fill out an online form to apply for the internship opportunity offered by the Rwandan development board. The reason behind selection of this specific interface is that it contains almost all types of common controls including text boxes, drop down lists, radio buttons, check boxes and the other reason is the vast variety of target users. Users were asked to perform the required activity within 30 minutes. The participants were not provided any human assistance during task completion in order to avoid any biased effect.

5.4.1 Education in Rwanda

In Rwanda the education system is distributed in three stages i.e. six years primary school, three years junior secondary and three years senior secondary school[2]. The country's principal language is Kinyarwanda while Swahili is also commonly used in communication. The teaching language for primary years 1 to 3 is Kinyarwanda. In years 4 to 6 the syllabus is in English or French.
There is a big ICT skills shortage at the present time, which limits ICT education to tertiary institutions and elite secondary schools, while there are merely 4.1% internet users till June 30, 2010 in Rwanda[3]. From 2004-2008, literacy rate in Rwanda was 77%, which is a relatively higher percentage, however, transition into secondary school drastically decreases to mere 31%[4]. For this case study users from Rwanda

[2] Ministry of Education. Rwanda Education Statistics, Jan (2012) http://www.mineduc.gov.rw/IMG/pdf/2011_RWANDA_EDUCATION_STATISTICS.pdf

[3] Internet World Stats, Africa Internet Usage and Population Stats, July 2012. http://www.internetworldstats.com/stats1.htm#africa

[4] The United Nations Children's Fund (UNICEF) Country Wise Statistics, Rwanda Statistics, July 2012.

were selected who got at least one year formal or informal education but left school without passing 6th grade (>=1 & < 6).

5.4.2 Requirements of Semi Literate Users

The requirements of low literate users were identified through an exploration exercise in which they were asked to read all instructions. The users could write a note or question on the paper during the task so that it can be discussed or asked after the completion of the assignment. It was quite interesting that thirty seven out of total fifty subjects wrote their confusions on the paper during completion of assigned task (filling out an on-line form). Thirty-one of these were unable to understand all instructions given in English. The users were low literate and stressed to provide verbal instructions along with the written text. The users also raised different queries related to different instructions and web controls, like what to do in case of dropdown list or if there are checkboxes.
The first step towards improving existing user interfaces is to simplify the text. It should be less cluttered and more target-oriented. There are several software development approaches which act as basis for applying specific frameworks in designing software. After analyzing major software development methodologies discussed in next section 5.3, a new interface for National Internship Program was designed by using a user centered design (UCD) approach by keeping semi literate users' requirements in mind.

5.4.3 Designing Virtual Character Based Interactive Interfaces

The existing interface of the internship program is refined and a virtual character is added to the user interface to assist semi literate users in task completion. Figure 5.3 shows the generic framework used for the inclusion of the virtual character [180]. International Organization for Standardization (ISO) described an international standard (ISO 13407: Human-centred design process) which defines a general process for including human-centered activities throughout a development life-cycle [196]. Once the necessity to adopt UCD methodology is identified then these four activities are part of human-centered design process.

1. Specify the context of use: Identify the people who will use the product, what they will use it for, and under what conditions they will use it.

---

http://www.unicef.org/infobycountry/rwanda_statistics.html

2. Specify requirements: Identify any business requirements or user goals that must be met for the product to be successful.
3. Create design solutions: This part of the process may be done in stages, build from a rough concept to a complete design.
4. Evaluate designs: The most important part of this process is the evaluation, ideally through usability testing with actual users and it is as integral part as the quality testing is part of good software development.

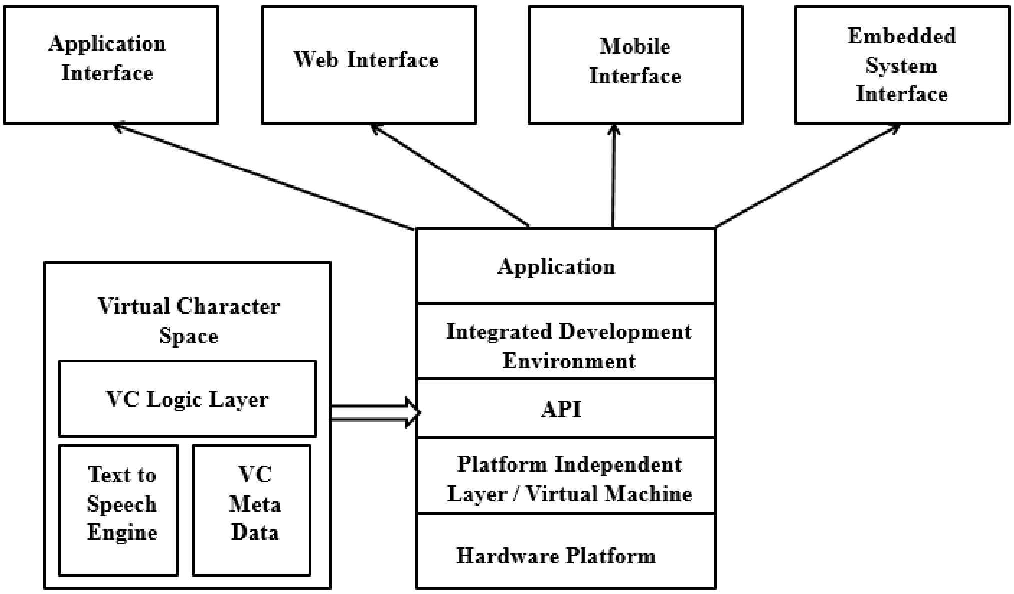


Figure 5.3. A generic framework for inclusion of virtual characters in different interfaces.

All these four activities are mapped in development of virtual character based interactive interface and described in figure 5.4. The instructions given in internship application form were translated in Kinyarwanda language using text to speech translation, and a virtual character is added at user interface to help users in understanding instructions at each step. The users can acquire help by clicking question mark button available with each instruction. Clear verbal instructions in Kinyarwanda language were provided in parallel with written instructions in English for the ease of users having different linguistic backgrounds. The translation from English to Kinyarwanda was performed very carefully and was revised and improved on the basis of valuable feedback provided by the students of National University of Rwanda.
The lip synchronization feature of virtual character with Kinyarwanda instructions

used to enhance the interaction of users. This got more attraction and interest of users and kept them busy for longer time to apply for the internship program with more devotion. Figure 5.5 shows new interface designed to facilitate the low literate users to file an application for internship program started by the Rwandan Development Board of Rwanda.

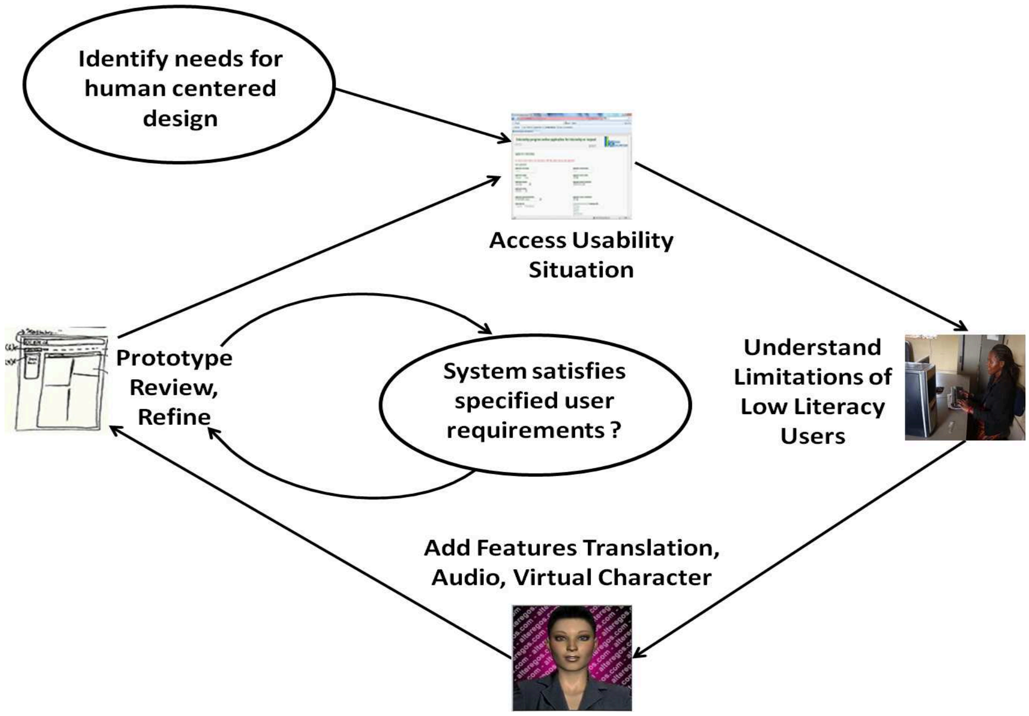


Figure 5.4. Design process of virtual character based interface.

5.5 Usability of Textual and Virtual Character Based Interfaces

Usability is ease of use and learnability of a system to carry out a task successfully. ISO defined usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" [197]. The word usability also refers towards methods used for enhancing ease of use in design process. Usability is part of usefulness and can

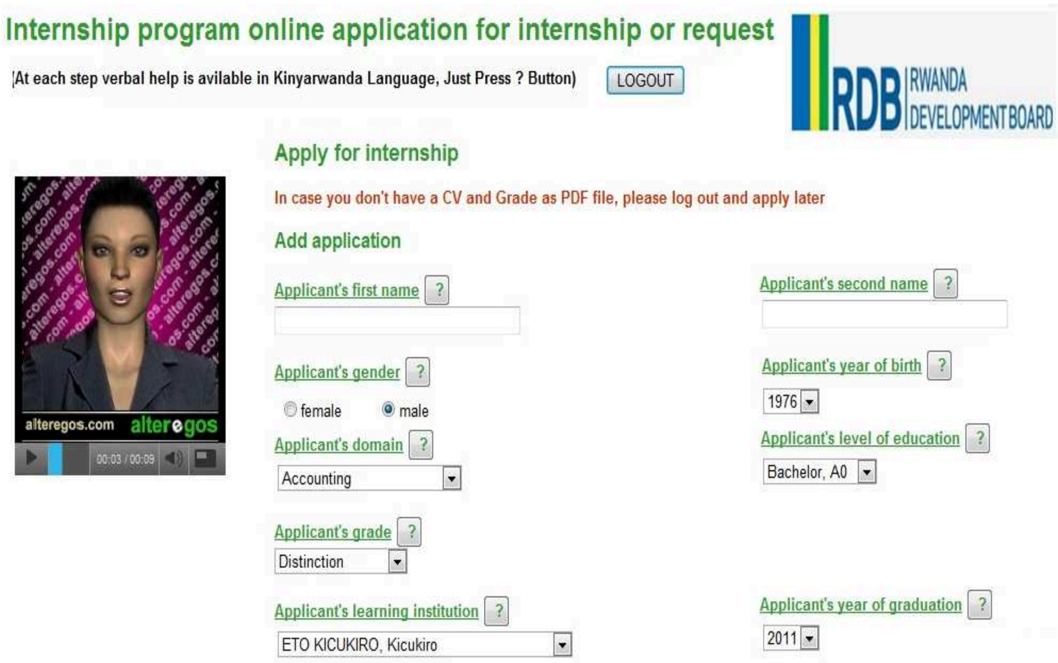


Figure 5.5. New virtual character based interface for internship program started by Republic of Rwanda.

be better defined with 5 quality components[5]:

1. Learnability: If the users are using the interface for the first time, how much it is easy and whether it helps in learning process?
2. Efficiency: How quickly user can perform the task?
3. Memorability: When users returns to the interface after some time, how easily they adjust?
4. Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
5. Satisfaction: How pleasant was the experience to use the interface was pleasant?

Now the question is whether the usability can be measured, and whether the learnability, ease of use and satisfaction can be quantified. Different usability metrics are

[5]Usability 101: Introduction to Usability, Jakob Nielsen, Retrieved 2014-02-12. http://www.nngroup.com/articles/usability-101-introduction-to-usability/

used to quantify the user experience, some of them can be easily calculated than others e.g. task success rates, task times, number of mouse clicks or keystrokes and self-reported ratings of frustration and delight are all examples of usability metrics [198].

5.5.1 Post Session Usability Metrics

The most obvious way to learn about usability of a system is to ask the users about their experience but there are number of ways to ask and organize these questions as one might ask to write down three major problems which seems an accessibility barrier to use this interface. Some further questions may include overall satisfaction, ease of use, memorability, awareness of certain features, clarity of terminology, visual appeal, and many others. The most efficient way could be to use some standardized tools for post session ratings.
In our study, four usability metrics are used to evaluate more suitable interface (textual or virtual character based) for semi literate users. Two usability metrics including task completion percentage and the mean time consumed are based on users' performance. After submission of the internship application form, two self reported post session usability metrics including After-Scenario Questionnaire (ASQ) and System Usability Scale (SUS) are used to evaluate the ease of use and learnability of both interfaces.

5.5.1.1 After-Scenario Questionnaire

The After-Scenario Questionnaire (ASQ) is developed by IBM [199]. ASQ contains few questions and its structure is simple to understand, furthermore little time is required to complete the questionnaire. In ASQ three questions were asked from users.

1. Ease of use: overall, I am satisfied with the ease of completing this task.
2. Amount of time: I am satisfied with the amount of time it took to complete the tasks in this scenario.
3. Available online help: I am satisfied with the support information (online help, messages, documentation) when completing the tasks.

Each of these questions in the ASQ touches on three areas of usability: effectiveness (question 1), efficiency (question 2), and satisfaction (all three questions). Users rate these questions on a 7 point rating scale from strongly agree to strongly disagree. If a user does not answer any question then report the average of remaining scores.

5.5.1.2 System Usability Scale

The System Usability Scale (SUS) is a simple questionnaire containing ten-items scale giving a global view of subjective assessments of usability. All 10 items are rated on 5 point Likert scale ranging from strongly agree to strongly disagree [200]. SUS contains 10 questions in which half are positively worded while the others half are negatively worded.

1. System reuse: I think that I would like to use this system frequently.
2. System complexity: I found the system unnecessarily complex.
3. Ease to use: I thought the system was easy to use.
4. Required human help: I think that I would need the support of a technical person to be able to use this system.
5. System integration: I found the various functions in this system were well integrated.
6. System inconsistency: I thought there was too much inconsistency in this system.
7. System learning: I would imagine that most people would learn to use this system very quickly.
8. System difficulty: I found the system very cumbersome to use.
9. Confidence: I felt very confident using the system.
10. Pre-learning/ Literacy: I needed to learn a lot of things before I could get going with this system.

Each item's score contribution will range from 0 to 4. To calculate a SUS score first add the score contributions from each item. For items 1, 3, 5, 7, and 9, the score contribution is the scale position minus 1. For items 2, 4, 6, 8, and 10, the contribution is 5 minus the scale position. Multiply the sum of the scores by 2.5 to obtain the overall SUS score. SUS is reliable, valid and shows consistent user's response on scale items. Low SUS scores indicate to the researchers that they needed to review and identify problems encountered with the interface [201].
In start SUS was only intended to measure perceived ease-of-use (a single dimension) but Lewis and Sauro showed that it provides a global measure of system satisfaction and sub-scales of usability and learnability [202]. Items 4 and 10 provide the learnability dimension and the other 8 items provide the usability dimension. SUS scores have a range of 0 to 100. Each item's score contribution will range from

0 to 4. For items 1,3,5,7,and 9 the score contribution is the scale position minus 1. For items 2,4,6,8 and 10, the contribution is 5 minus the scale position. The learnability is measured by adding the scores for items 4 and 10 and by multiplying result by 12.5. Usability is measured by adding scores for the rest eight items and by multiplying the result by 3.125. The SUS is preferred in this study due to its simple structure and property to be easily translated for non native English speakers.

5.5.2 Results

In the textual interface only ten persons out of fifty could complete their task while in the virtual character based interface sixteen out of fifty low literate users completed their task. In the textual interface, four users (8%) showed almost no interest due to limited literacy level and skills, while in the virtual character based interface there was only one (2%) such user who did not answer any question. Figure 5.6 shows the comparison of the results between the two groups and displays the percentage of users according to the number of questions answered. There were 50% of the users of the textual interface who answered more than ten questions while in virtual character based interface 74% of the users answered more than ten questions. 32% of the users completed the form by answering all questions in virtual character based interface while only 20% could complete the form in the case of the textual interface. The mean time spent by first group on textual interface was 17 minutes (S.D = 3.7) and the mean time spent by second group on the virtual character based interface was 25 minutes (S.D = 3.1). A single-factor ANOVA test conducted shows statistical significance ($F = 120.51 > F\ crit = 3.94$, $P\text{-value} < 0.05$) and rejects the null hypotheses that mean time consumed by all user groups is equal. More time spent on virtual character based interface indicates that users were more motivated and involved in the task.

5.5.2.1 Usability Issues

To figure out the actual problems behind the users' difficulty to perform the task and to collect their feedback, a short interview was conducted. The subjects were also requested to give their suggestion about possible changes or improvements in the interface. Figure 5.7 shows the summary of the interview session in which major usability issues are categorized among different types. The figure reveals that participants working with the virtual character based interface pointed out less usability issues and two areas including language problem and requiring human assistance are significantly improved. In virtual character based interface still three users showed inability to understand language of instructions. When the users were

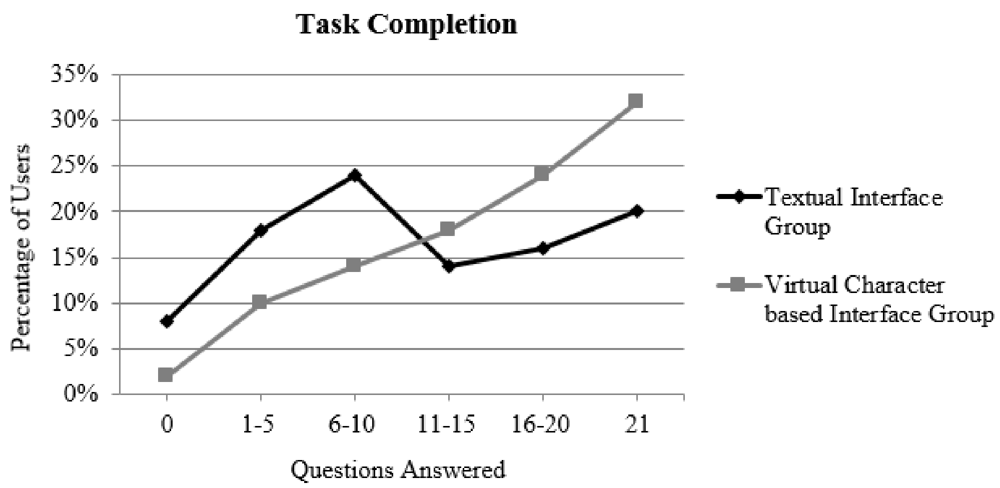


Figure 5.6. Comparison of both groups according to number of questions answered (total number of questions: 21).

inquired further, they informed that they were not familiar about some terminologies used in instructions. Some further explanation is required for the understanding and clarification. Five users of the virtual character based interface demanded external human assistance and informed that the recorded script of virtual character is not sufficient to answer all questions arising in their mind. Some users also mentioned other problems including inability to understand banking terms and abbreviations along with the study domains.

5.5.2.2 Help Acquired from Virtual Character

The virtual character could be interrupted, stopped and replayed again without any limitation. Table 5.5.2.2 shows the average number of times users acquired help from the virtual character. Some users repeatedly acquired help from virtual character for the same instruction. In interview session, the users informed that in some cases, they could not understand the complete instructions first time and had to hear the instructions again. There were also some users who enjoyed the lip synchronization of virtual character with Kinyarwanda language and tried to listen to the instructions multiple times.

At the end of the experiment the users feedback about both interfaces were collected by using System Usability Scale (SUS). Figure 5.8 shows the average score of usability and learnability by both groups. In virtual character based interface usability is improved by at least 41% and learnability by 39% in average.

Figure 5.9 shows the average user's response against all three questions of ASQ.

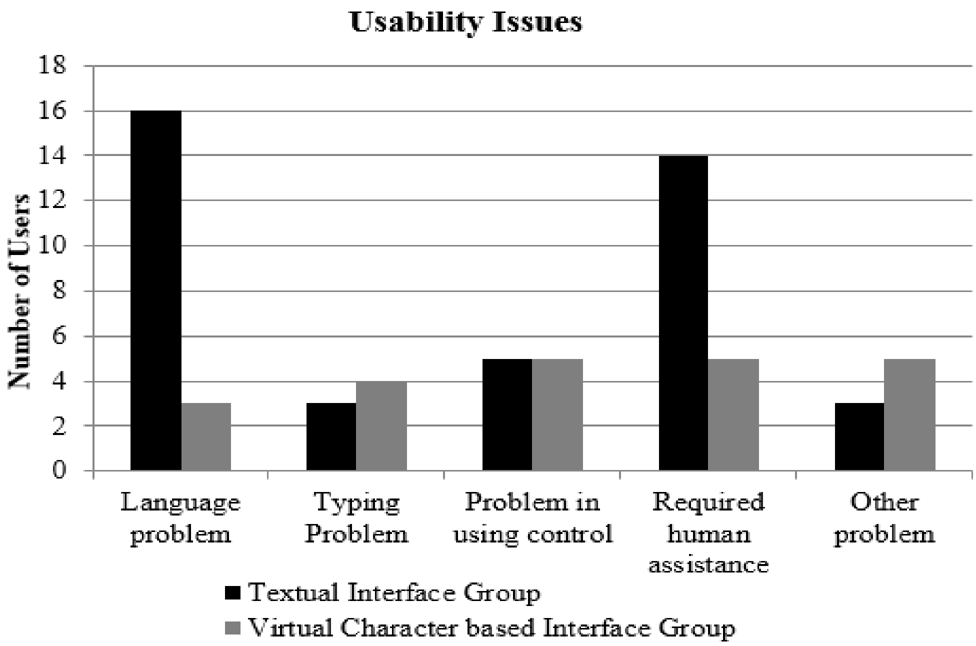


Figure 5.7. Usability issues categorized according to different types identified by both groups.

Table 5.2. Number of times help acquired by users from virtual character in virtual character based interface.

	Average	Standard Deviation
Number of Times Help Acquired from Virtual Character	21	10
Repeatedly Maximum Number of Times Help Required at Same Instruction	2	1

Low scores are better than high scores due to the anchors used in the 7-point scales and show user's degree of satisfaction. Users were given 30 minutes to complete their task so both user groups found this time reasonable to complete the task, but comparatively users of virtual character based interface spent more time on interface as discussed earlier. From the results it is clear that virtual character based interfaces were more effective and efficient. User's participation is encouraged by virtual character based interface and users liked to involve themselves with this. Users answered more questions, raised less number of usability issues and frequently acquired help from virtual character.

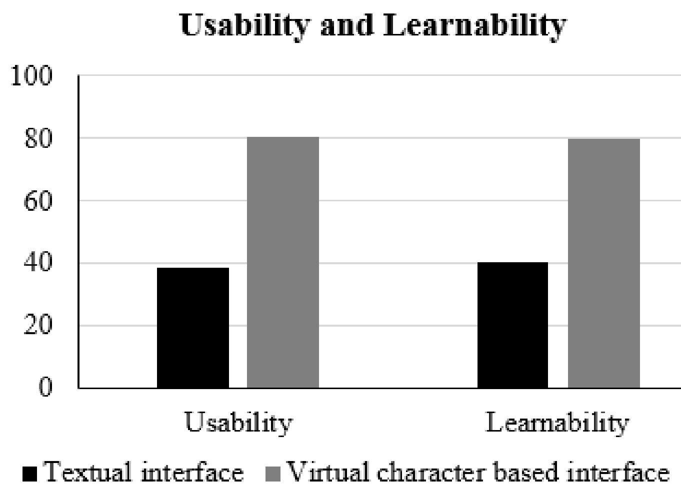


Figure 5.8. Usability and learnability Score of both groups.

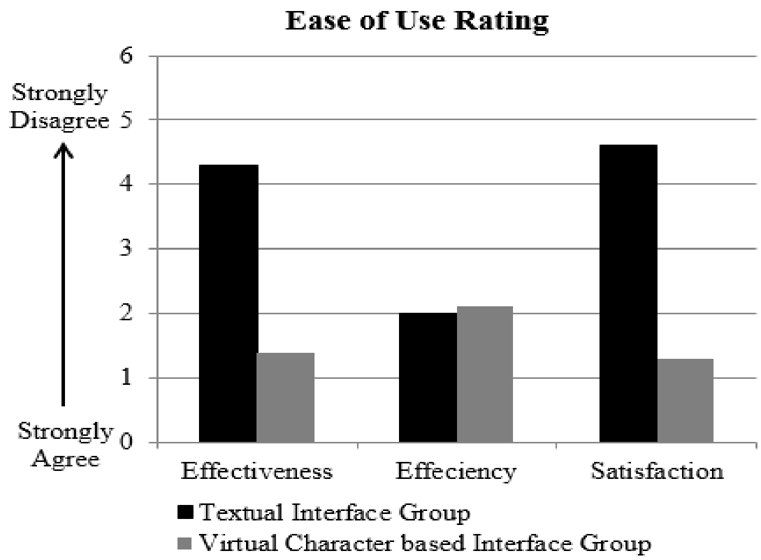


Figure 5.9. After-Scenario Questionnaire (ASQ), ease of use rating by users of both groups.

5.5.3 Discussion

Citizens' participation in e-Government and the usability of ICT is increasing effectively in the developing countries. Governments have demonstrated a real interest to transform relationships between government services and the citizens, particularly by improving the usability of ICT and offering information services online. A number of initiatives are taken for this purpose, but citizens participation in reality,

is still in its infancy and exists on an abstract scale. The developing countries are facing immense challenges to implement ICT because of low literacy level of the public. It is therefore essential to emphasis the crucial importance of establishing innovative participation channels and citizen-centric policies for the global effort to promote human development and access to the resources.[6]
Different steps are taken and variety of methods are applied and tested to determine the efficiency and effectiveness to enhance accessibility and adaptation of ICT for the common people. Through this study, the author discovered that people were using the online services but they had to depend others and were reluctant to acquire help from anyone due to the privacy. People also have to wait and pay considerable service charges to helpers. Localization of interfaces with regional verbal instructions pronounced by virtual character with the synchronized lips movements, can be one useful and potential method for designing helpful interfaces and providing the guidance by means of the virtual character.
There are no previous applications providing additional multimedia help features, on e-government sites for semi literate users in Kinyarwanda. Our findings during this studies provided evidence that using online services is not a problem for most low literate users, but the problem is in terms of lacking confidence or technological usage which could be provided with helping interfaces. It was found that illiterate or semi literate users can benefit from online services in conjunction with visual help [143]. Stronger ease of use agreement for such interface in Rwanda reveals that virtual character based interfaces enhanced their understanding and could be a possibility and a way to motivate and attract people's attention towards the usage of ICT and hence enhance e-participation. The users interviewed were interested in virtual characters and favored its usage. In virtual character based interface the users pointed out lesser number of usability issues and two areas including the language problem and requiring human assistance are significantly improved.
Although usability of virtual character interface is improved by 41%, it was not clear what exactly is the role of different features in improvement i.e., whether the translation of text into local language played major role or whether the vocal instructions improved their task completion ability or the usage of a life like character helped the semi literate users more was unclear. For true understanding about the role of online multimedia assistance, another experiment is performed on low literate users in which four separate interfaces are tested by involving 180 users in three different cities of Rwanda. Chapter 6 provides the details related to second experiment.

[6]E-governance and Citizen Participation in West Africa, Report by United Nations Development Program (UNDP), 2011
http://www.undpegov.org/WA-epart.html

Chapter 6

Helpful Interfaces

Despite many years of research in human computer interaction, user interfaces still pose access challenges for many semi-literate users who have little interaction with technology; lack of literacy may be partly to blame. A person who cannot read and understand online information, requires assistance to compensate for his or her limitations. In previous chapter 5, a case study is discussed in which textual interface is compared with virtual character based interface from semi literate users' perspective. Although the results showed that multimedia assistance on interface can enhance the performance of semi literate users but it was not clear that which help mode(the way the help is provided) is more suitable for semi literate users and whether the online assistance has same impact on all semi literate users or it further depends on their literacy skills.

This chapter presents an extended case study contrasting four different interface design styles providing online assistance for the task of completing an online internship application. 180 subjects in the Republic of Rwanda belong to three different cities with low literacy skills were divided into four different groups to complete online task by using one of these interfaces.

The author has already discussed necessary background about ICT and accessibility and importance of usability studies in chapter 5. Here in this chapter, the Section 6.1 discusses the consistent help feature from semi literate users' perspective and provides overview of second case study. Section 6.2 provides the details about methodology, target community and experimental set-up. Section 6.3 discusses initial interaction of semi-literate users with the original interface, usability barriers and design recommendations by end users. Section 6.4 presents newly designed interfaces and reports the evaluation results based on usability of helpful interfaces. In Section 6.5, the conclusion of this dissertation and future work are discussed.

6.1 Case Study 2: Comparison of Textual, Vocal and Virtual Character Based Interfaces

In most of the studies discussed in section 5.2, researchers are agreed upon providing continuous help features in the form of voice, video or animations during ICT usage for low literacy users [146, 147, 157]. User manuals, user tutorials or user helps are mostly designed to assist the users to perform tasks without external help. Despite a large body of multidisciplinary research on helpful and user-oriented interfaces design, help facilities found in most commercial software are so ill-conceived that they are often unhelpful. The help system must be seen as an integral part of the user interface. Its role should be users assistance in order to accomplish the required task by answering the users queries or describing software's functionality. The help should be provided in useful and easy to understand way for the accessibility of contents in achieving the required goals [203].
The previous work done for low literate users mostly emphasized consistent help features using different graphical aids. In this study, the effects and usability of online assistance (continuous help) are analyzed and different help modes including textual, vocal and virtual character are compared to check the performance of semi-literate users. The task involved filling online form by providing basic information such as name, gender and level of education to apply for an internship program. The purpose of study was to evaluate the usability of different user interfaces from the semi-literate users' perspective. Specifically, we evaluated the following hypotheses:

- Users facilitated with online assistance will have better task completion percentages compared to those who were using interfaces without help.
- The subjective usability of the interface with online assistance will be higher than the average usability of the interface without assistance.
- The mean time required to complete the task for each user group will differ. In case of additional online help, more time will be required to read and understand the task.

6.2 Methodology

The experimentation was carried out in two phases. In the first phase, subjects were asked to use the original interface provided by the Rwandan Government and apply for an internship program. At the end of a users' interaction with the original interface, user suggestions, feedback and design recommendations were acquired. In the next phase three different types of interfaces were designed according to users' feedback. Each newly designed interface is tested by another group of semi-literate users.

6.2.1 Literacy Level

School education in Rwanda is compulsory and free of charge for the first nine years. As discussed in section 5.4.1, traditionally the education system in Rwanda is divided into primary (6 years) and lower secondary schooling (3 years). To clearly understand the capabilities of low literacy users, the participants were divided in three categories; the users who left the school in first 3 years of education, or users who completed primary level of education and the users who studied in lower secondary level schooling.
Information literacy is the ability to solve problems, taking benefit from information technology and networks [204]. In order to evaluate users' literacy level in information technology, the participants were asked about their experience of using computers.

6.2.2 Participants

The purpose of this study was to evaluate different interfaces from a semi-literate users' perspective, so our target subjects were the people who either were not completely illiterate (absolutely can not read or write) nor properly literate (studied in higher classes beyond the compulsory education in Rwanda). Appendix A shows the pre-experiment questionnaire distributed among participants to evaluate their traditional literacy and acquaintance with technology. The subjects participated voluntarily in the study.
We excluded some participants on the basis of the pre-experiment questionnaire. This questionnaire helped to identify, which subjects can cope and interact with technology. 65 subjects were discarded out of 250. Twenty Six participants were found completely illiterate having insufficient writing and reading skills. Fourteen users were confused when performing basic interaction with the mouse and keyboard; they could not control the mouse and some would click both buttons and could not point the mouse to the appropriate location. Sixteen subjects were discarded because they had more than nine years of school education or had used computers for more than one year regularly. Two subjects with vision problems were also excluded. Finally, seven subjects were excluded for other reasons i.e. to maintain gender ratio and age between 21 and 40 years (mean age 29 years).
As described earlier, 180 subjects were chosen to participate in the experiment and divided into four equal groups named alphabetically Group A, Group B, Group

C and Group D. Each group consisting of 45 users was further divided into sub-categories of basic users, moderate users and expert users. The details are given in table 6.1. The number of females was less than the number of males due to the limited number of women work at vehicle refurbishing sites. But the male to female ratio was similar among the selected groups.
The subjects' primary languages were Kinyarwanda, English and French but semi-literate users usually speak local languages like Kinyarwanda or Swahili language as Swahili is considered as lingua-franca in much of East Africa. In our experiment users were asked about their English and Swahili language skills and users found with insufficient reading or writing skills were discarded as described earlier.

Table 6.1. Categorization and literacy level of participants in each group.

	Education (Years)	Number of Participants		
		Male	Female	Total
Basic Users	Non Formal Education or Regular Schooling (< = 3 Years)	12	3	15
Moderate Users	Primary School Education (> 3 & < =6 Years)	12	3	15
Expert Users	Junior Secondary School Education (> 6 & < =9 Years)	12	3	15

6.2.3 Tasks

The participants were given a ten minute briefing through a multimedia presentation to explain them task before the actual experimental. They were also given time to get acquainted with the system before the actual experiment. There were three tasks in the experimentation process. The first task was to fill out a pre-experiment questionnaire to identify the literacy level and previous exposure with computer technology. The second task was to fill out an on-line form to apply for an internship program offered by the Rwandan Government[1]. The final task was the post-experiment questionnaire and short interview to take feedback from users. The post experiment questionnaire was used to get quantitative feedback on the

[1] Rwanda Development Board. Rwanda Youth Internship Program by Republic of Rwanda, Oct (2012)
http://www.rdb.rw/departments/human-capital-and-institutional-development/rwanda-youth-internship-program.html

set of instructions provided while filling out the online application form.The interviews were helpful for open ended qualitative feedback about missing features and suggestions to make the process more easy, robust and user friendly.

6.2.4 Experimental Setup and Procedure

The study was conducted mainly in three different vehicle refurbishing (auto garages) workshops situated in three major cities(Kigali, Butare and Gisenyi) of Republic of Rwanda. Due to the limited availability of computers on site, one group of fifteen participants and one user class (basic, moderate or expert) were called at a time. All the computers used during the experiment were homogenous with respect to operating system and environment.

All users were informed in the beginning that if they do not feel comfortable with the experiment, or they feel like this is beyond their capability or due to any other reason, they wish to quit from the experiment at any stage, then they should not submit the form and should inform the mentor. For the internship application task, the users had to register themselves on the internship website initially by providing user-name and password of their own choice. After registration, the users were asked to complete online internship form comprised of fifteen different questions. Users were not provided any human assistance during task completion to avoid any biased effect.

The experimental studies were conducted in two different sessions. In the first session, the consent form and the pre-experiment questionnaires were filled out. This activity lasted for about twenty minutes. The second session was conducted almost after three weeks in which users applied for the internship and answered post-experiment questionnaires. All the users were given 30 minutes to complete online application form during the experiment and fifteen minutes were given for post-experiment questionnaire and interview.

6.2.5 Evolution Criteria

All the interfaces were evaluated on the basis of task accomplishment, the time consumed and System Usability Scale(SUS) rating. SUS is a reliable, low-cost usability scale that can be used for global assessments of systems usability [42]. SUS is a highly robust and versatile tool for usability professionals. It has been used across a wide range of user interfaces, including standard OS-based software interfaces, web pages and web applications [205]. SUS is considered the simplest questionnaire, yielding the reliable results across sample sizes (See details in section ref{label:SUS) [206].

It was found that a significant proportion of non-native English speakers failed to understand the word “cumbersome” in question 8 of the SUS so it should be re-worded [207]. The SUS was translated in the Swahili language and question 8 was re-worded in Swahili as “I found the system very inconvenient/ difficult to use (Nimeona mfumo sana hazifai/ vigumu kutumia)”. It was observed how many times users acquired help in newly designed interfaces, the detailed statistics are given in section 6.4.1.

6.3 Users’ Interaction with Original Interface

Group A was given the task of submitting an online application for internship Program by using original interface. The original interface was available only in English. The activity was closely observed by one mentor since one group was performing the task at a time. The mentor was responsible for smooth operation of systems and to make sure that the subjects perform the assigned task themselves without getting any help from other users and eventually conducted post experiment interviews to take the users’ feedback. During the first interaction with the original interface, it was observed that the semi-literate users faced difficulty in filling out forms without help from other humans. Twenty-four out of forty-five users were unable to answer all questions while only six out of fifteen from basic users class could answer all questions.

6.3.1 Usability Barriers

From post experiment questionnaires and interviews, it is observed that semi-literate users were facing the following challenges.

(i) Language difficulties: Mostly basic (6 out of 15), moderate (3 out of 15) and expert (2 out of 15) users found it difficult to understand some instructions and key terms. The key terms like ‘ID’, ‘domain’ and ‘distinction’ were not clear for them. During the interview, most of the subjects requested that the contents be translated to the local language in order to understand instructions and key terms easily.

(ii) Vocal Instructions: Six out of forty-five users mentioned in interviews that if instructions are given orally in the regional language then it will be easier to understand them as compared to textual instructions.

(iii) Seeking Human Assistance: During the interview session nine basic users and four moderate users emphasized and requested that some human should assist them in task completion. Two moderate users asked for human assistance but also were concerned and raised the issue of privacy.

(iv) More Detailed Instructions: Three basic and two moderate users insisted that instructions are too short, e.g. in registration the instruction 'login:' is too short to be comprehended, it will be better to provide more detailed instructions like select or enter your user-name to login. Some users were confused in selecting their preferred districts as it was mentioned to tick at least three check boxes, so some of them either ticked more than three districts in each province and some of them just selected 1^{st} available three districts in each province.

(v) Difficulty in understanding abbreviations: Four basic users, two moderate and one expert user could not understand the abbreviations of bank names, they insisted to provide complete bank name in the interface i.e. in place of BNR, the complete name of bank "The National Bank of Rwanda" should be used. Some expert users suggested to give abbreviations in parenthesis after the complete name of bank and logo of bank may also help to recognize bank name.

(vi) Others: Four users mentioned some other usability issues. One user recommended bigger font size. Three users were unable to select the right domain for them as some similar choices were ambiguous for them i.e. Agriculture: crop production or Agriculture: vegetable production. The reason is obvious and understandable as with low literacy it is difficult to differentiate closely related fields.

Figure 6.1, shows the overall percentage of users from Group A who highlighted specific usability issues.

6.3.2 Design Recommendations

The following design recommendations were learned from semi-literate users' feedback.

1. Provide translation in local language.
2. Add vocal instructions especially for more low literate users.
3. The purpose of the experiment was to decrease dependency of semi-literate users on human assistance, so it was decided to add life-like characters to replace human assistance and to maintain privacy.
4. Include some additional functional instructions to assist in task completion. In case of short instruction like "Enter your name", a complete functional instruction that "Move your cursor and click in following text box and then write your name" should be given.

5. Provide the full names in place of abbreviations.

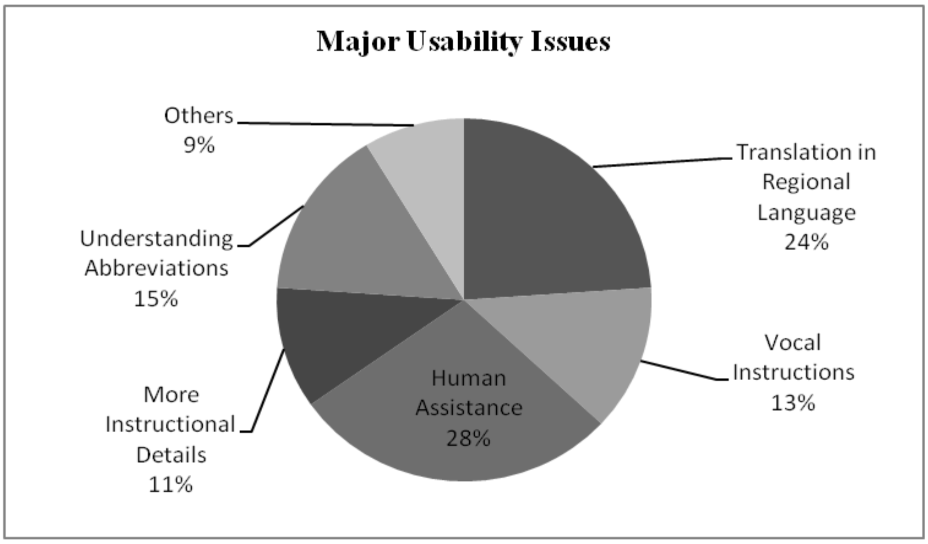


Figure 6.1. Major usability issues found in 1st interaction with original interface by users of Group-A

6.4 Designing New Interfaces

The feedback from semi-literate users with the original interface of internship program led us to redesign the original interface. This interface was modified and redesigned with some additional help for semi-literate users to assist them in task completion as well as translation of contents into local language. New interfaces were designed in such a way that in very first contact with the interface, the users will get clear instructions about what they are supposed to do and how it can be achieved.

We designed three new interfaces known as textual interface, vocal interface and virtual character based interface. For all the new interfaces, structure and overall layout were kept the same. The users' interaction (the way the users get help) in the newly designed interfaces is the same but the mode of help is different i.e. it is either textual or vocal or through virtual character. In the newly designed interfaces help was provided on demand. A button labeled with an interrogation mark was added with each instruction. By clicking the button, users received help about the specific task in Swahili language. The help is shown in a pop-up window alongside the button and usually remains visible until the user clicks the close button. Figure 6.2 summarizes the approach adopted in the newly designed interfaces.

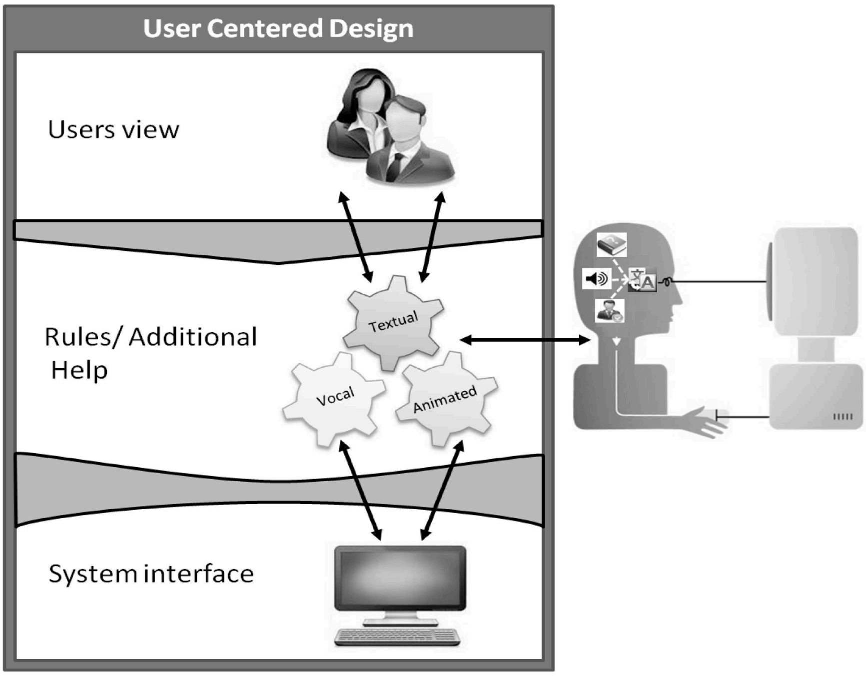


Figure 6.2. User Centered Design approach adopted in newly designed interfaces

In the textual interface, additional help is provided on screen in textual form in Swahili. The translation is initially carried out using Google translation application programming interface (API) and on the later stage it was cross checked and verified by a native language speaker from Rwanda. The translated text consisted of the two parts. The 1^{st} part contained the details about the steps required to fill up a field of the internship application form while the 2^{nd} part was about the actual instructions, e.g. in case of selection of birth year from drop down list, a complete functional instruction was given as "Please move your cursor, click the small drop down arrow button given at the right side of text box and select your birth year (Tafadhali sogeza mshale, bonyeza kwenye kifungo kidogo na ushushe chini upande wa kulia kwenye kisanduku na chagua mwaka wako wa kuzaliwa)".
In the vocal interface, recorded audio instructions in Swahili were played on users' request. In the virtual character based interface, an expressive virtual human was added to the user interface which provided functional instructions vocally. The virtual character was lip synchronized with Swahili and looked like a female assistant. Figure 6.4, shows a user testing newly designed virtual character based interface. In the newly designed interfaces, the complete bank names were used instead of abbreviations, e.g. the abbreviation "BCR!", was replaced by its complete name "Banque Commerciale Du Rwanda(BCR)".

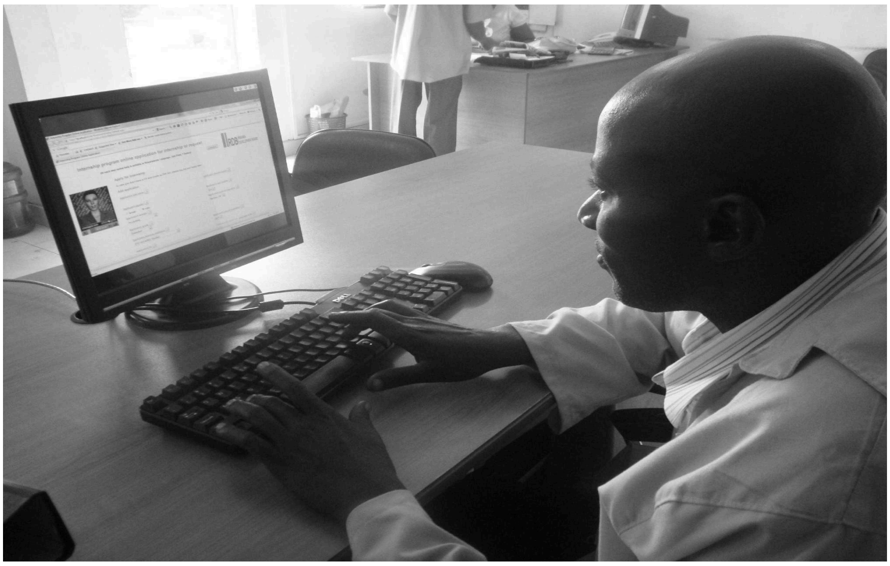


Figure 6.3. A semi-literate user is testing newly designed virtual character based Interface in Republic of Rwanda.

6.4.1 Evaluation of Different Helpful Interfaces

Group B used the redesigned textual interface. Group C used vocal interface and Group D used virtual character based interface. For a better understanding the variations observed in performance metrics including task success, time on task and users' preference, the results are not only compared between groups but also compared within the groups.
Figure 6.4 shows the Mean time variation found within basic, moderate and expert user categories of each group. Group A spent the least time in interaction with original interface, in contrast to Group D, which consumed the most time in interaction with virtual character based interface. In newly designed interfaces help was provided in English and Swahili. Acquiring help and reading the bilingual instructions could be the possible factors and reason for spending extra time on newly designed interfaces. A single-factor ANOVA test conducted shows statistical significance ($F = 29.21 > Fcrit = 2.66$, $P\text{-value} < 0.05$) and reject the hypothesis that average time consumed by all user groups is equal. Expert users comparatively consumed less time and were more efficient in task completion.

6.4.2 Task Completion Percentage

The important question is whether the performance and task completion percentage of users consuming more time on newly designed interface was improved. To answer this question, we counted the number of fields where the users had successfully entered valid data. If any field in the form is incomplete, empty, filled with invalid data or makes no sense then it is not considered complete e.g. numerical values in name or email address without proper structure.

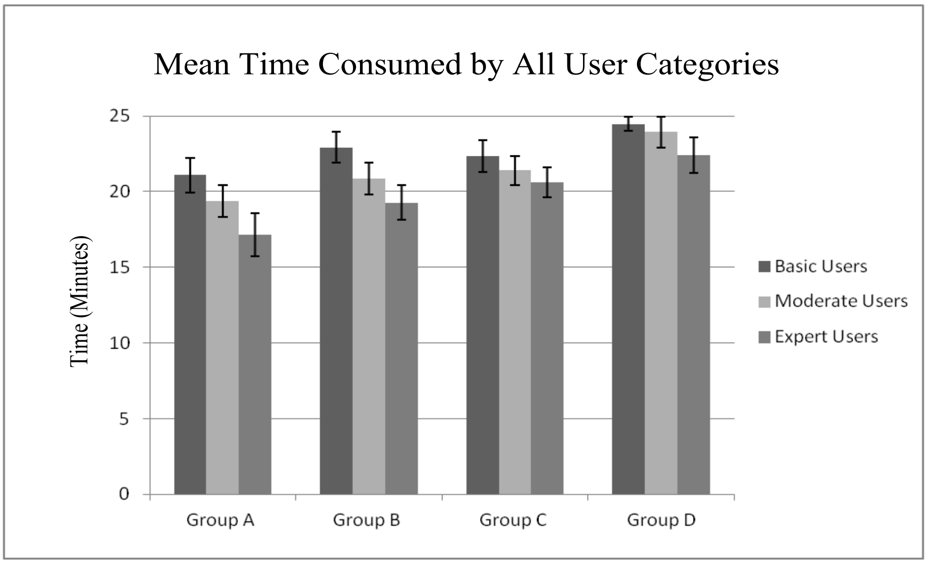


Figure 6.4. Comparison of Mean time consumed by each category of users in every group for online internship application. Error bars represent a 95 percent confidence interval.

Figure 6.5 compares the percentage of task completed by all types of users in each group. In average Group A completed 52%, Group B completed 70%, Group C 84%, while Group D completed 94% of the assigned task.
Overall, 47% of users in Group A successfully filled and completed all fields of internship application. There were 18% users who gave up and did not provide answer of any field, while 35% users partially filled the form. Five out of fifteen (33%) of basic users, showed inability to understand instructions and gave up. In group B, 67% users answered all fields of the internship application, while the percentage of users who failed decreased significantly and only 7% users gave up. In Group C, 73% users were completely successful while in Group D, 87% of users successfully filled out all the fields. From figure 6.5, it is clear that new features added to the existing interface had significant impact on the performance of semi-literate users. Although Group B, Group C and Group D consumed more time, their task completion percentages were higher than Group A.

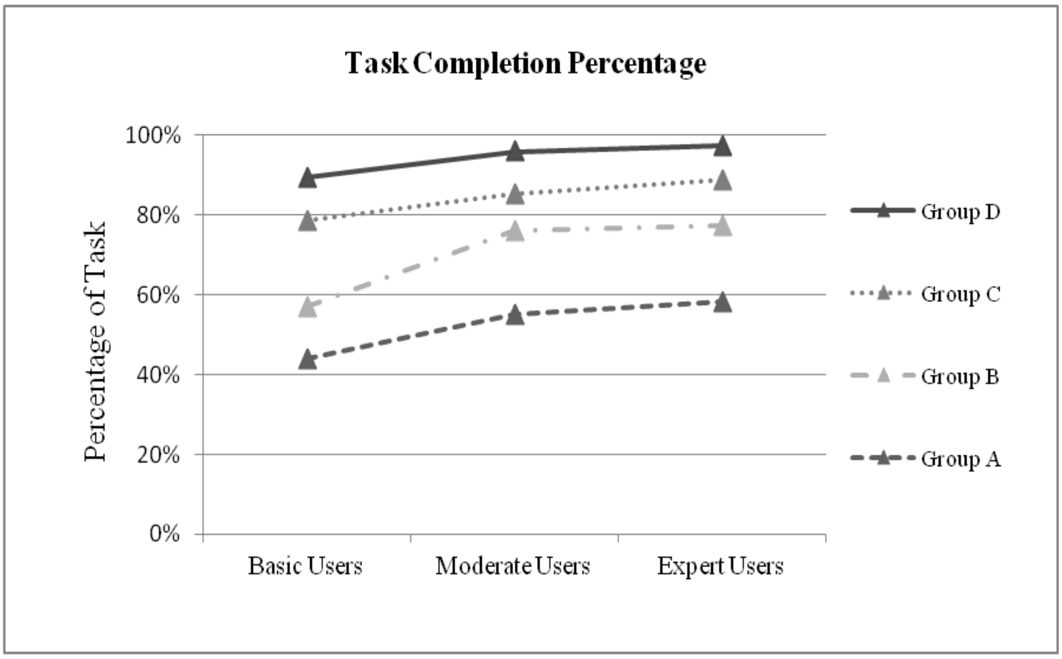


Figure 6.5. Percentage of task completion by all types of users in each group

6.4.3 Average System Usability Rating

fter submitting an internship application, all users were given a SUS questionnaire to collect their feedback on the subjective usability of each interface. The same questionnaire was used for all types of users and groups, so that their results can be compared. Figure 6.6, shows the average SUS score of all user categories of the four groups. Error bars represent the 95 percent confidence interval for the mean. The average subjective usability score is almost double of Group D (SUS score 80) for the virtual character based interface as compared to Group A (SUS score 39) that used original interface. The SUS mean of basic users in group B was 49 as they showed least interest in textual interfaces among newly designed interfaces. A single-factor ANOVA test conducted on system usability scale ($F = 57 > F_{crit} = 2.66$, P-value <0.05) shows that the average subjective usability score of each user group is different in all four types of interfaces.

Independent samples t-tests were conducted to compare newly designed interfaces with the original one. The average SUS score of Group B (M=60.49, SD=18.23) was significantly greater than Group A (M=39.22, SD=17.27) conditions; t (88)=5.68 >Tcrit(one-tail) = 1.66, P-value <0.05. Similarly, the average SUS score of Group C (M=72.06, SD=14.84) conditions; t (86)=9.67 >Tcrit(one-tail) = 1.66, P-value <0.05 and Group D (M=80.17, SD=12.14) conditions; t (79)=13.01 >Tcrit(one-tail) = 1.66, P-value <0.05 were significantly greater than Group A. These results suggest that online assistance have an effect on average subjective usability of interfaces.

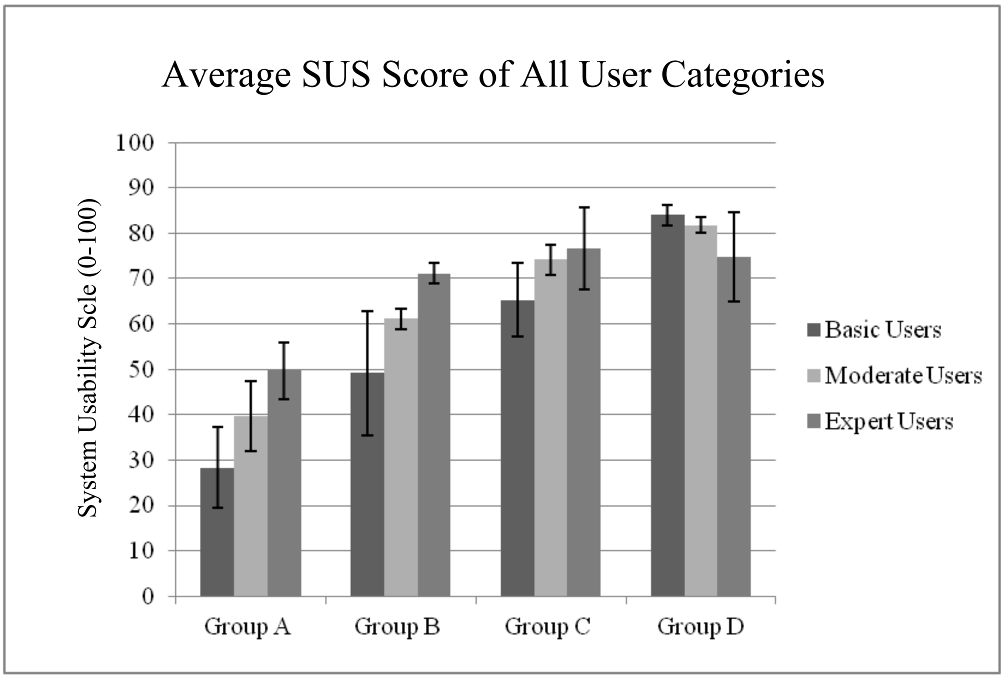


Figure 6.6. The average SUS Score of all user categories in four groups used different interfaces. Error bars represent the 95 percent confidence interval for the mean.

6.4.4 Help Acquired

Table 6.2 provides the details about the number of times the users clicked question mark button and acquired help in newly designed interfaces. There were total 15 instructions in internship application form while in virtual character based interface the help seeking average was ten. The maximum number of clicks for seeking help were 19, which suggests some users tried to acquire help multiple times from the virtual character to understand each single instruction.

The data also reveals that users acquired more help in the newly designed interfaces than the original one and it also improved their performance in task completion as shown in figure 6.5. According to our knowledge, it is first in depth study of semi-literate users' interaction with a real life application, in which semi-literate users are further divided into three sub categories according to their formal schooling. The role of online assistance is evaluated and measured by usability metrics.

The results indicate that in applications subjected to low literacy users, performance

Table 6.2. Average number of times the functional help is acquired in newly designed interfaces.

	Help Acquired (Average)	Standard Deviation	95% C.I	Minimum Clicks	Maximum Clicks
Group B	7 	2	1	2	12
Group C	8 	3	1	3	15
Group D	10 	4	1	2	19

greatly depends on the assistance provided to cope with their inability. Thanks to the online assistance, about 87% users were able to complete their task, while without online assistance only 47% users could complete the assigned task.
It is also pertinent to mention that literacy or some knowledge of information technology decreases users' dependency on extra assistance as in average moderate users acquired less number of times help than basic users. The expert users acquired help even less times as compared to moderate users as shown in table 6.2. The major reason behind the failure of Mr. Clippy (Microsoft Office assistant) was the impolite behavior as it was hijacking user control and trying to help everyone without demand [208]. In our experiment the help is provided on demand. The subjective usability rating (SUS score) for virtual character based interface which was 84 by basic users shows that such systems are more effective, efficient and satisfactory for more low literate users.

6.5 Conclusion and Future Perspective

Interfaces for semi literates and naturally challenged users are emerged as new domain in last few years. The author has investigated existing frameworks and techniques used to enhance the understanding of low literacy users. The interface of Platform Independent LIS Dictionary is developed for deaf users. The Sign Language Dictionary accepts Italian lemmas as input and provides their representation in the Italian Sign Language as output. The Sign Language dictionary has 3082 signs as set of Avatar animations in which each sign is linked to a corresponding Italian lemma. It is available in the form Web application online and as a mobile application for Windows Phone7, iPhone and Android. The LIS Dictionary is implemented in such a way that its functionality is enhanced according to the availability

of resources and new signs can be easily added in future. For offline users, lexicon data set along video signs is stored locally while for online users, dictionary is mapped with MultiWordNet synsets in client server fashion. Future work includes the extension of the LIS dictionary in order to cover an even larger data set. A scalable videos visualization system could be implemented in order to overcome network overloading issues. A portal can be developed in order to enhance the sign verification by deaf users. Through the portal they could eventually propose new variants and improvements of existing signs. In addition to the scalability some experiments can be performed to get the feedback from the community on the lexical aspects.
The author integrated LIS lexical resources with MultiWordNet (MWN) database to form first LIS MultiWordNet (LMWN). LMWN contains information about lexical relations between words, semantic relations between lexical concepts (synsets), correspondences between Italian and Sign language lexical concepts and semantic fields (domains). The LMWN not only provides the LIS meanings of the lemma but also provides the set of words with similar meanings and the sign associated with each lemma in the database. All semantical and syntactical relationships of LIS Signs with MWN are defined and the words existing in both databases are determined. The approach ehances the deaf users' understanding of written Italian language and shows that a relatively small set of lexicon can cover a significant portion of MWN. Integration of LIS signs with MWN made it useful tool for computational linguistics and natural language processing. The process of translation from Italian written text to LIS is transformed into service oriented paradigm. Different components of Rule-based translation are integrated into translation workflows that improves the flexibility to entertain the languages other than Italian. The workflows are implemented in Java Application Building Center (jABC) which is a framework for extreme model driven design (XMDD).The XMDD paradigm is a very rigorous approach for tackling the complexity challenge of service-oriented design of complex applications. A new web portal can be implemented in future with objective to enhance deaf learning of written Italian Language in schools. Different scenarios of LIS vocabulary can be created to check the coherence and understanding of different signs. The subjects would be set of deaf users, normal people having different literacy level and aptitude. The community feedback will be helpful to check the originality of the material and comprehensiveness of lexical resources.
The author developed four different interfaces to analyze the effects and usability of online assistance (consistant help) on semi-literate users and compared different help modes including textual, vocal and virtual character on the performance of semi-literate users. The experiments were completed in Republic of Rwanda in which 180 semi-literate users participated. In the newly designed interfaces the instructions were automatically translated in Swahili language. The results show that the performance of semi-literate users improved significantly when using the online assistance. Moreover, the percentage of completed tasks increased from 52% to 94%. Statistical

analysis of the System Usability Scale (SUS) indicates that their average subjective usability score boosted from 39 to 80. The proposed virtual character based interfaces not only assisted deaf users but also enhanced the applications utility for low literacy users by translating contents in their own language. In future, The idea of online assistance can be enhanced by providing an independent API to the designers with all features of multimedia help. The translation process can be linked with the existing API and can be enhanced by text to voice features. The virtual characters can be used in the public websites containing only textual information to enhance the performance of low literacy users as mere availability of interfaces is not sufficient for a user who lacks basic reading and writing skills.

Bibliography

[1] P. Prinetto, U. Shoaib, and G. Tiotto. The italian sign language sign bank: Using wordnet for sign language corpus creation. In *Communications and Information Technology (ICCIT), 2011 International Conference on*, pages 134 –137, march 2011.

[2] E. Radutzky, C. Torossi, and M.P.D. Fund. Dizionario bilingue elementare della lingua dei segni italiana lis:oltre 2.500 significati .(ed.) kappa. *Rome, Italy*, 2001.

[3] Leonardo Lesmo, Alessandro Mazzei, and Daniele Radicioni. Linguistic processing in the atlas project. In *International Workshop on Sign Language Translation and Avatar Technology(SLALT)*, Berlin, Germany, 2011.

[4] International Telecommunication Union. *Measuring the information society.* Telecommunication Development Bureau (BDT), International Telecommunication Union, Geneva, Switzerland, 2013.

[5] Deaf fact sheet. Deafness and hearing loss. Key facts 300, World Heath Organization (WHO), Geneva, Switzerland, February 2013.

[6] N. Bertoldi, G. Tiotto, P. Prinetto, E. Piccolo, F. Nunnari, V. Lombardo, R. Damiano, L. Lesmo, and AD Principe. On the creation and annotation of a large-scale italian-lis parallel corpus. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, CSLT Valetta, Malta*, pages 19–22, 2010.

[7] C. Geraci, K. Battaglia, A. Cardinaletti, C. Cecchetto, C. Donati, S. Giudice, and E Mereghetti. The lis corpus project: A discussion of sociolinguistic variation in the lexicon. *Sign Language Studies*, 11(4):528–574, 2011.

[8] R.P. Meier, K. Cormier, and D. Quinto-Pozos. *Modality and structure in signed and spoken languages.* Cambridge University Press, 2002.

[9] Mark Kutner, Elizabeth Greenberg, and Justin Baer. A first look at the literacy of america's adults in the 21st century. nces 2006-470. *National Center for Education Statistics*, 2006.

[10] Christina Zarcadoolas, Mercedes Blanco, John F Boyer, and Andrew Pleasant. Unweaving the web: an exploratory study of low-literate adults' navigation skills on the world wide web. *Journal of Health Communication*, 7(4):309–324, 2002.

[11] Kaushik Ghosh, Tapan S Parikh, and Apala Lahiri Chavan. Design considerations for a financial management system for rural, semi-literate users. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pages 824–825. ACM, 2003.

[12] Krithi Ramamritham, Anil Bahuman, Subhasri Duttagupta, Chaitra Bahuman, and Srividya Balasundaram. Innovative ict tools for information provision in agricultural extension (december 2005). In *Information and Communication Technologies and Development, 2006. ICTD'06. International Conference on*, pages 34–38. IEEE, 2006.

[13] Jahanzeb Sherwani, Nosheen Ali, Sarwat Mirza, Anjum Fatma, Yousuf Memon, Mehtab Karim, Rahul Tongia, and Roni Rosenfeld. Healthline: Speech-based access to health information by low-literate users. In *Information and Communication Technologies and Development, 2007. ICTD 2007. International Conference on*, pages 1–9. IEEE, 2007.

[14] UIS fact sheet. Adult and youth literacy. International literacy data 26, UNESCO Institute for Statistics, Montreal, Canada, Sep 2013.

[15] Deaf fact sheet. Deaf statistics. Deaf populations overseas, Galaudet university library, Washington, D.C. , US, June 2010.

[16] I Medhi, B Pitti, and K Toyama. Text-free ui for employment search. In *Proc. Asian Applied Computing Conference (AACC2005)*, Kathmandu, Nepal, 2005.

[17] Kathryn P Meadow. Early manual communication in relation to the deaf child's intellectual, social, and communicative functioning. *Journal of Deaf Studies and Deaf Education*, 10(4):321–329, 2005.

[18] Teresa V Mitchell and Melissa T Maslin. How vision matters for individuals with hearing loss. *International journal of audiology*, 46(9):500–511, 2007.

[19] Nina Timmermans. *The Status of Sign Languages in Europe: Report*. Council of Europe, 2005.

[20] Scott K Liddell. *Grammar, gesture, and meaning in American Sign Language.* Cambridge University Press, 2003.

[21] Andrea Falletto, Paolo Prinetto, and Gabriele Tiotto. An avatar-based italian sign language visualization system. In *Electronic Healthcare*, pages 154–160. Springer, 2009.

[22] I. Zwitserlood. Sign language lexicography in the early 21st century and a recently published dictionary of sign language of the netherlands. *International Journal of Lexicography*, 23(4):443–476, 2010.

[23] T.A. Johnston. Corpus linguistics and signed languages: no lemmata, no corpus. In *3rd Workshop on the Representation and Processing of Sign Languages,Paris, France*, pages 82–87, 2008.

[24] Matt Huenerfauth. A survey and critique of american sign language natural language generation and machine translation systems. *Computer and Information Sciences, University of Pennsylvania, Philadelphia*, 2003.

[25] Cecilia Conrath Doak, Leonard G Doak, and Jane H Root. Teaching patients with low literacy skills. *AJN The American Journal of Nursing*, 96(12):16M, 1996.

[26] Sabine Linden and Anita H Cremers. Cognitive abilities of functionally illiterate persons relevant to ict use. In *Proceedings of the 11th international conference on Computers Helping People with Special Needs*, volume 5105 of *Lecture Notes in Computer Science*, pages 705–712. Springer Berlin Heidelberg, 2008.

[27] Neesha Kodagoda, William Wong, and Nawaz Kahan. Identifying information seeking behaviours of low and high literacy users: combined cognitive task analysis. In *Proceedings of the 9th Bi-annual international conference on Naturalistic Decision Making*, NDM'09, pages 58–58, Swinton, UK, 2009. British Computer Society.

[28] Jakob D Jensen, Andy J King, LaShara A Davis, and Lisa M Guntzviller. Utilization of internet technology by low-income adults: the role of health literacy, health numeracy, and computer assistance. *Journal of aging and health*, 22(6):804–826, 2010.

[29] Arun Kumar, Sheetal K. Agarwal, and Priyanka Manwani. The spoken web application framework: User generated content and service creation through low-end mobiles. In *Proceedings of the 2010 International Cross Disciplinary*

Conference on Web Accessibility (W4A), W4A '10, pages 2:1–2:10, New York, NY, USA, 2010. ACM.

[30] Jahanzeb Sherwani, Sooraj Palijo, Sarwat Mirza, Tanveer Ahmed, Nosheen Ali, and Roni Rosenfeld. Speech vs. touch-tone: Telephony interfaces for information access by low literate users. In *Information and Communication Technologies and Development (ICTD), 2009 International Conference on*, pages 447–457. IEEE, 2009.

[31] ISO DIS. 9241-210: 2010. ergonomics of human system interaction-part 210: Human-centred design for interactive systems. *International Organization for Standardization (ISO), Switzerland*, 2009.

[32] Donald A Norman and Stephen W Draper. *User centered system design; new perspectives on human-computer interaction.* L. Erlbaum Associates Inc., 1986.

[33] Debora Shaw. Handbook of usability testing: How to plan, design, and conduct effective tests, 1996.

[34] Tiziana Margaria and Bernhard Steffen. Service-Orientation: Conquering Complexity with XMDD. In Mike Hinchey and Lorcan Coyle, editors, *Conquering Complexity*, pages 217–236. Springer London, 2012.

[35] Tiziana Margaria and Bernhard Steffen. Business Process Modelling in the jABC: The One-Thing-Approach. In Jorge Cardoso and Wil van der Aalst, editors, *Handbook of Research on Business Process Modeling.* IGI Global, 2009.

[36] T.A. Johnston. Language standardization and signed language dictionaries. *Sign Language Studies*, 3(4):431–468, 2003.

[37] Nadeem Ahmad, Davide Barberis, Nicola Garazzino, Paolo Prinetto, Umar Shoaib, and Gabriele Tiotto. A virtual character based italian sign language dictionary. In *Proceedings of the 13th International Conference on Computers Helping People with Special Needs*, Linz, Austria, Jul 2012.

[38] Davide Barberis, Nicola Garazzino, Paolo Prinetto, Gabriele Tiotto, Alessandro Savino, Umar Shoaib, and Nadeem Ahmad. Language resources for computer assisted translation from italian to italian sign language of deaf people. In *Proceedings of Accessibility Reaching Everywhere, AEGIS Workshop and International Conference*, pages 363–370, Brussels, Belgium, Nov 2011.

[39] Umar Shoaib, Nadeem Ahmad, Paolo Prinetto, and Gabriele Tiotto. A platform-independent user-friendly dictionary from italian to lis. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2435–2438, Istanbul, Turkey, May 2012.

[40] U. Shoaib, N. Ahmad, P. Prinetto, and G. Tiotto. Integrating multiwordnet with italian sign language lexical resources. *Expert Systems with Applications*, 41(5):2300 – 2308, 2014.

[41] T. Margaria and B. Steffen. Service Engineering: Linking Business and IT. *Computer*, 39(10):45–55, 2006.

[42] John Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, pages 189–194, 1996.

[43] Elena Pizzuto, Paolo Rossini, Tommaso Russo, and Erin Wilkinson. Formazione di parole visivo-gestuali e classi grammaticali nella lingua dei segni italiana (lis): dati disponibili e questioni aperte. *Maria GROSSMANN e Anna M. THORNTON (a cura di), La formazione delle parole*, pages 443–463, 2005.

[44] Marc Marschark. Intellectual functioning of deaf adults and children: Answers and questions. *European Journal of Cognitive Psychology*, 18(1):70–89, 2006.

[45] Loes N Wauters, Wim HJ Van Bon, and Agnes EJM Tellings. Reading comprehension of dutch deaf children. *Reading and Writing*, 19(1):49–76, 2006.

[46] Ronnie B Wilbur. The use of asl to support the development of english and literacy. *Journal of Deaf Studies and Deaf Education*, 5(1):81–104, 2000.

[47] Margaret Harris and Constanza Moreno. Speech reading and learning to read: A comparison of 8-year-old profoundly deaf children with good and poor reading ability. *Journal of Deaf Studies and Deaf Education*, 11(2):189–201, 2006.

[48] Susan Goldin-Meadow and Rachel I Mayberry. How do profoundly deaf children learn to read? *Learning Disabilities Research & Practice*, 16(4):222–229, 2001.

[49] Paul Miller. What the visual word recognition skills of prelingually deafened readers tell about their reading comprehension problems. *Journal of Developmental and Physical Disabilities*, 18(2):91–121, 2006.

[50] Virginia Volterra, Olga Capirci, and M Cristina Caselli. What atypical populations can reveal about language development: The contrast between deafness and williams syndrome. *Language and Cognitive Processes*, 16(2-3):219–239, 2001.

[51] Elizabeth Keating and Gene Mirus. American sign language in virtual space: Interactions between deaf users of computer-mediated video communication and the impact of technology on language practices. *Language in Society*, 32(05):693–714, 2003.

[52] Mary R Power, Des Power, and Louise Horstmanshof. Deaf people communicating via sms, tty, relay service, fax, and computers in australia. *Journal of Deaf Studies and Deaf Education*, 12(1):80–92, 2007.

[53] Mary R Power and Des Power. Everyone here speaks txt: Deaf people using sms in australia and the rest of the world. *Journal of Deaf Studies and Deaf Education*, 9(3):333–343, 2004.

[54] Shawn P Saladin and Sandra E Hansmann. Psychosocial variables related to the adoption of video relay services among deaf or hard-of-hearing employees at the texas school for the deaf. *Assistive Technology*, 20(1):36–47, 2008.

[55] Elena Pizzuto, Paolo Rossini, and Tommaso Russo. Representing signed languages in written form: questions that need to be posed. In *VETTORI, C. Proceedings of the Second Workshop on the Representation and Processing of Sign Languages–5 International Conference on Language Resources and Evaluation, Genoa*, pages 1–6, 2006.

[56] W. C. Stokoe. *Sign language structure : an outline of the visual communication systems of the American deaf.* Bufffalo, N.Y, 1960.

[57] Lynn Friedman. Formational properties of american sign language. *On the other hand: New perspectives on American Sign Language*, pages 13–56, 1977.

[58] V Sutton and R Gleaves. Signwriter-the world's first sign language processor. *Deaf Action Committee for SignWriting, La Jolla, CA*, 1995.

[59] Siegmund Prillwitz. *HamNoSys: Hamburger Notations System for Sign Language; an Introduction.* Zentrum für Dt. Gebärdensprache, 1987.

[60] S Prillwitz, R Leven, H Zienert, T Hanke, and J Henning. Hamburg notation system for sign languages-an introductory guide. *International Studies on Sign Language and the Communication of the Deaf*, 5, 1989.

[61] W.C. Stokoe, DC. Casterline, and Carl G. Croneberg. *A dictionary of American sign language on linguistic principles.* Gallaudet Press, Washington, D.C., 1965.

[62] Scott K Liddell and Robert E Johnson. American sign language: The phonological base. *Sign language studies*, 64:195–278, 1989.

[63] Joe Martin. A linguistic comparison two notation systems for signed languages: Stokoe notation & sutton signwriting. Western Washington University, 2000.

[64] Amy L McCarty. Notation systems for reading and writing sign language. *The Analysis of verbal behavior*, 20:129, 2004.

[65] Thomas Hanke. Hamnosys-representing sign language data in language resources and language processing contexts. In *LREC*, pages 1–6, 2004.

[66] E Antinoro Pizzuto, I Chiari, and P Rossini. Representing signed languages: theoretical, methodological and practical issues. *Spoken communication. Newcastle upon Tyne: Cambridge Scholars Publishing*, pages 205–241, 2010.

[67] Liz Scott-Gibson. 7.4 sign language interpreting: An emerging profession. *Constructing deafness*, page 253, 1990.

[68] M. Vendrame and G. Tiotto. Atlas project: Forecast in italian sign language and annotation of corpora. In *Proceedings of 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, pages 239–243, 2010.

[69] V. Lombardo, F. Nunnari, and R. Damiano. A virtual interpreter for the italian sign language. In *Intelligent Virtual Agents*, pages 201–207. Springer, 2010.

[70] Antoinette Hawayek, Riccardo Del Gratta, and Giuseppe Cappelli. A bilingual dictionary mexican sign language-spanish/spanish-mexican sign language. In *LREC*. European Language Resources Association, 2010.

[71] Chiara Vettori, Oliver Streiter, and Judith Knapp. From computer assisted language learning (call) to sign language processing: the design of e-lis, an electronic bilingual dictionary of italian sign language & italian. In *Proceedings of the Workshop on the Representation and Processing of Sign Languages, LREC*, 2004.

[72] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Ashwin Thangali, Haijing Wang, and Quan Yuan. Large lexicon project: American sign language video corpus and sign language indexing/retrieval algorithms. In *Proc. Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora*, 2010.

[73] Y.A. Ohali. Identification of most desirable parameters in sign language tools: A comparative study. *Global Journal of Computer Science and Technology*, 10(6):23–29, 2010.

[74] M. Kipp, A. Heloir, and Q. Nguyen. Sign language avatars: Animation and comprehensibility. volume 6895, pages 113–126. Springer, 2011.

[75] Ralph Elliott, John RW Glauert, JR Kennaway, and Ian Marshall. The development of language processing support for the visicast project. In *Proceedings of the fourth international ACM conference on Assistive technologies*, pages 101–108. ACM, 2000.

[76] Ed Sims and Dan Silverglate. Interactive 3d characters for web-based learning and accessibility. In *ACM SIGGRAPH 2002 conference abstracts and applications*, pages 314–314. ACM, 2002.

[77] Anna Cavender, Richard E Ladner, and Eve A Riskin. Mobileasl:: intelligibility of sign language video as constrained by mobile phone technology. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 71–78. ACM, 2006.

[78] JR Kennaway, J.R.W. Glauert, and I. Zwitserlood. Providing signed content on the internet by synthesized animation. *ACM Transactions on Computer Human Interaction*, 14(3), 2007.

[79] Yu-Hsien Chiu, Chung-Hsien Wu, Hung-Yu Su, and Chih-Jen Cheng. Joint optimization of word alignment and epenthesis generation for chinese to taiwanese sign synthesis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):28 –39, jan. 2007.

[80] M. Huenerfauth. *Generating American Sign Language classifier predicates for English-to-ASL machine translation*. PhD thesis, University of Pennsylvania, 2006.

[81] I. Marshall and E. Safar. Grammar development for sign language avatar-based synthesis. *In Proc. of the 3rd International Conference on Universal Access in Human-Computer Interaction (UAHCI), Las Vegas*, 5, 2005.

[82] D Stein, J Bungeroth, and H Ney. Morpho-syntax based statistical methods for sign language translation. In *11th Annual conference of the European Association for Machine Translation, Oslo, Norway*, pages 223–231, 2006.

[83] Guillem Massó and Toni Badia. Dealing with sign language morphemes for statistical machine translation. In *4th workshop on the representation and processing of sign languages: corpora and sign language technologies, Valletta, Malta*, pages 154–157, 2010.

[84] Jan Bungeroth, Daniel Stein, Philippe Dreuw, Morteza Zahedi, and Hermann Ney. A german sign language corpus of the domain weather report. In *Fifth International Conference on Language Resources and Evaluation*, pages 2000–2003, 2006.

[85] Thomas Hanke. Synergies between transcription and lexical database building: the case of german sign language. In *Proceedings of the Corpus Linguistics Conference 2009 (CL2009),*, page 346, 2009.

[86] Penny Boyes-Braem. A multimedia bilingual database for the lexicon of swiss german sign language. *Sign language & linguistics*, 4(1-2):1–2, 2001.

[87] Adam Schembri et al. British sign language corpus project: Open access archives and the observer's paradox. *Deafness cognition and language research centre, University college London. LREC*, 2008.

[88] Onno Crasborn, Inge Zwitserlood, et al. The Corpus NGT: an online corpus for professionals and laymen. In *Construction and Exploitation of Sign Language Corpora. 3rd Workshop on the Representation and Processing of Sign Languages at LREC*, pages 44–49, 2008.

[89] Trevor Johnston. From archive to corpus: transcription and annotation in the creation of signed language corpora. *International journal of corpus linguistics*, 15(1):106–131, 2010.

[90] Trevor Johnston. The lexical database of auslan (australian sign language). *Sign language & linguistics*, 4(1-2):1–2, 2001.

[91] E.A. Pizzuto, P. Rossini, M.A. Sallandre, E. Wilkinson, et al. Deixis, anaphora and highly iconic structures: Cross-linguistic evidence on american (asl), french (lsf) and italian (lis) signed languages. *Sign Languages: spinning and unraveling the past, present and future.*, pages 475–495, 2008.

[92] D. Barberis, N. Garazzino, E. Piccolo, P. Prinetto, and G. Tiotto. A web based platform for sign language corpus creation. *Computers Helping People with Special Needs*, pages 193–199, 2010.

[93] E. Pianta, C. Girardi, and R. Zanoli. The textpro tool suite. In *Proceedings of Conference on Language Resources and Evaluation (LREC), Marrakech,Morocco*, volume 8, 2008.

[94] G.A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[95] Piek Vossen. Wordnet, eurowordnet and global wordnet. *Revue française de linguistique appliquée*, 7(1):27–38, 2002.

[96] P. Buehler, A. Zisserman, and M. Everingham. Learning sign language by watching tv (using weakly aligned subtitles). In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2961 –2968, 2009.

[97] M.S.J. Lewis and D.W. Jackson. Television literacy: Comprehension of program content using closed captions for the deaf. *Journal of Deaf Studies and Deaf Education*, 6(1):43–53, 2001.

[98] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244, 1990.

[99] E. Pianta, L. Bentivogli, and C. Girardi. Multiwordnet: Developing an aligned multilingual database. In *Proc. 1st Int'l Conference on Global WordNet , Maysor, India*, pages 293–302, 2002.

[100] Piek Vossen. *EuroWordNet : a multilingual database with lexical semantic networks*. Kluwer Academic, Dordrecht, The Netherlands, 1998.

[101] H. Rodríguez, S. Climent, P. Vossen, L. Bloksma, W. Peters, A. Alonge, F. Bertagna, and A. Roventini. The top-down strategy for building eurowordnet: Vocabulary coverage, base concepts and top ontology. *Computers and the Humanities*, 32:117–152, 1998.

[102] D. Tufis, D. Cristea, and S. Stamou. Balkanet: Aims, methods, results and perspectives. a general overview. *Romanian Journal of Information Science and Technology*, 7(1-2):9–43, 2004.

[103] Christiane Fellbaum. A semantic network of english : The mother of all wordnets. *Cognitive Science*, 32(2-3):209–220, 1998.

[104] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[105] J.D. Novak and A.J. Cañas. The theory underlying concept maps and how to construct and use them. *Florida Institute for Human and Machine Cognition Pensacola Fl*, 2006.

[106] Bran Selic. The pragmatics of model-driven development. *IEEE software*, 20(5):19–25, 2003.

[107] Tiziana Margaria and Bernhard Steffen. Agile IT: Thinking in User-Centric Models. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation*, volume 17 of *Communications in Computer and Information Science*, pages 490–502. Springer Berlin / Heidelberg, 2009.

[108] Sven Jörges. Extreme model-driven development and jabc. In *Construction and Evolution of Code Generators*, volume 7747 of *Lecture Notes in Computer Science*, pages 39–71. Springer Berlin Heidelberg, 2013.

[109] Tiziana Margaria and Bernhard Steffen. Lightweight coarse-grained coordination: a scalable system-level approach. *Software Tools for Technology Transfer*, 5(2-3):107–123, 2004.

[110] Anna-Lena Lamprecht. *User-Level Workflow Design - A Bioinformatics Perspective*, volume 8311 of *Lecture Notes in Computer Science*. Springer, 2013.

[111] Bernhard Steffen, Tiziana Margaria, Ralf Nagel, Sven Jörges, and Christian Kubczak. Model-driven development with the jabc. In Eyal Bin, Avi Ziv, and Shmuel Ur, editors, *Hardware and Software, Verification and Testing*, volume 4383 of *Lecture Notes in Computer Science*, pages 92–108. Springer Berlin Heidelberg, 2007.

[112] Sven Jörges, Christian Kubczak, Felix Pageau, and Tiziana Margaria. Model Driven Design of Reliable Robot Control Programs Using the jABC. In *Proceedings of 4th IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (EASe 2007)*, pages 137–148, 2007.

[113] Stefan Naujokat, Anna-Lena Lamprecht, Bernhard Steffen, Sven Jörges, and Tiziana Margaria. Simplicity Principles for Plug-In Development: The jABC Approach. In *Proceedings of the Second International Workshop on Developing Tools as Plug-Ins (TOPI 2012), Zürich, Switzerland*, pages 7–12, 2012.

[114] Stefan Naujokat, Anna-Lena Lamprecht, and Bernhard Steffen. Loose Programming with PROPHETS. In Juan de Lara and Andrea Zisman, editors, *Proc. of the 15th Int. Conf. on Fundamental Approaches to Software Engineering (FASE 2012), Tallinn, Estonia*, volume 7212 of *LNCS*, pages 94–98. Springer Heidelberg, 2012.

[115] Martina Hörmann, Tiziana Margaria, Thomas Mender, Ralf Nagel, Bernhard Steffen, and Hong Trinh. The jabc approach to rigorous collaborative development of scm applications. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation*, volume 17 of *Communications in Computer and Information Science*, pages 724–737. Springer Berlin Heidelberg, 2008.

[116] Bernhard Steffen and Tiziana Margaria. METAFrame in Practice: Design of Intelligent Network Services. In Ernst-Rüdiger Olderog and Bernhard Steffen, editors, *Correct System Design*, volume 1710 of *Lecture Notes in Computer Science*, pages 390–415. Springer, 1999.

[117] Tiziana Margaria, Christian Kubczak, Marc Njoku, and Bernhard Steffen. Model-based Design of Distributed Collaborative Bioinformatics Processes in the jABC. In *Proceedings of the 11th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'06)*, pages 169–176, Los Alamitos, CA, USA, August 2006. IEEE Computer Society.

[118] Tiziana Margaria, Christian Kubczak, and Bernhard Steffen. Bio-jETI: a service integration, design, and provisioning platform for orchestrated bioinformatics processes. *BMC Bioinformatics*, 9 Suppl 4:S12, 2008.

[119] Anna-Lena Lamprecht, Tiziana Margaria, and Bernhard Steffen. Seven Variations of an Alignment Workflow - An Illustration of Agile Process Design and Management in Bio-jETI. In *Bioinformatics Research and Applications*, volume 4983 of *Lecture Notes in Bioinformatics*, page 445â456, Atlanta, Georgia, 2008. Springer.

[120] Martin Karusseit and Tiziana Margaria. Feature-based Modelling of a Complex, Online-Reconfigurable Decision Support Service. *Electronic Notes in Theoretical Computer Science*, 157(2):101 – 118, 2006.

[121] Tiziana Margaria, Oliver Rüthing, and Bernhard Steffen. ViDoC - Visual Design of Optimizing Compilers. In *Program Analysis and Compilation, Theory and Practice, Essays Dedicated to Reinhard Wilhelm on the Occasion of His 60th Birthday*, volume 4444 of *LNCS*, pages 145–159. Springer, 2006.

[122] Christian Kubczak, Ralf Nagel, Tiziana Margaria, and Bernhard Steffen. The jabc approach to mediation and choreography. In *2nd Semantic Web Service Challenge Workshop*, volume 234. Citeseer, 2006.

[123] Markus Bajohr and Tiziana Margaria. Matrics: A service-based management tool for remote intelligent configuration of systems. *Innovations in Systems and Software Engineering*, 2(2):99–111, 2006.

[124] Sven Jörges, Christian Kubczak, Felix Pageau, and Tiziana Margaria. Model driven design of reliable robot control programs using the jabc. In *Proceedings of the Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems*, pages 137–148. IEEE Computer Society, 2007.

[125] Tiziana Margaria and Bernhard Steffen. Lightweight coarse-grained coordination: a scalable system-level approach. *International Journal on Software Tools for Technology Transfer*, 5(2-3):107–123, 2004.

[126] Sven Jörges, Tiziana Margaria, and Bernhard Steffen. FormulaBuilder: a tool for graph-based modelling and generation of formulae. In *Proc. of 28th Int.*

Conf. on Software Engineering (ICSE 2006), Shanghai, China, pages 815–818, 2006.

[127] Marco Bakera, Sven Jörges, and Tiziana Margaria. Test your Strategy: Graphical Construction of Strategies for Connect-Four. In *Proceedings of the 2009 14th IEEE International Conference on Engineering of Complex Computer Systems*, ICECCS '09, pages 172–181, Washington, DC, USA, 2009. IEEE Computer Society.

[128] Makoto Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. Of the International NATO Symposium on Artificial and Human Intelligence*, pages 173–180, New York, NY, USA, 1984. Elsevier North-Holland, Inc.

[129] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[130] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[131] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[132] Bonnie J. Dorr, Pamela W. Jordan, and John W. Benoit. A survey of current paradigms in machine translation. volume 49 of *Advances in Computers*, pages 1 – 68. Elsevier, 1999.

[133] Leonardo Lesmo, Vincenzo Lombardo, and Cristina Bosco. Treebank development: the tut approach. In *In Proceedings of the International Conference on Natural Language Processing*. Citeseer, 2002.

[134] N. Hatami, Paolo Ernesto Prinetto, and Gabriele Tiotto. An editor for assisted translation of italian sign language. In *EWDTS 2009: 7th IEEE East-West Design & Test Symposium*, pages 415–418, Kharkov (Ukraine), 2009. SPD FL Stepanov V.V. Ukraine, 61168, Kharkov, Ak. Pavlova st., 311.

[135] R. Borgotallo, C. Marino, Elio Piccolo, Paolo Ernesto Prinetto, Gabriele Tiotto, and M. Rossini. A multilanguage database for supporting sign language translation and synthesis. In *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, volume W13, pages 23–26. European Language Resources Association, 2010.

[136] Alice Ruggeri, Cristina Battaglino, Gabriele Tiotto, Carlo Geraci, Daniele Radicioni, Alessandro Mazzei, Rossana Damiano, and Leonardo Lesmo. Where should i put my hands? planning hand location in sign languages. In *Proceedings of the Workshop on Computational Models of Spatial Language Interpretation and Generation, Boston*, pages 24–31, 2011.

[137] Liwei Zhao, Karin Kipper, William Schuler, Christian Vogler, Norman Badler, and Martha Palmer. A machine translation system from english to american sign language. In *Envisioning Machine Translation in the Information Future*, pages 54–67. Springer, 2000.

[138] Z Yonah. Icts as tools for poverty reduction. *Proceeding of the Discourse on Engineering Contribution in Poverty Reduction*, 2005.

[139] Education For All. Understandings of literacy. Education for all global monitoring report, United Nations Educational, Scientific and Cultural Organization (UNESCO), Paris, UNESCO, 2006.

[140] UNESCO Education Sector. The plurality of literacy and its implications for policies and programs: Position paper. *Paris: United National Educational, Scientific and Cultural Organization*, 2004.

[141] Peter Roberts. Defining literacy: Paradise, nightmare or red herring? *British Journal of Educational Studies*, 43(4):412–432, 1995.

[142] Zereh Lalji and Judith Good. Designing new technologies for illiterate populations: A study in mobile phone interface design. *Interacting with Computers*, 20(6):574–586, 2008.

[143] Nadeem Ahmad. Virtual character based interactive interfaces for deaf and functionally illiterate users. In *Software Business. From Physical Products to Software Services and Solutions*, volume 150 of *Lecture Notes in Business Information Processing*, pages 223–224. Springer Berlin Heidelberg, 2013.

[144] Marcel Goetze and Thomas Strothotte. An approach to help functionally illiterate people with graphical reading aids. In *Smart Graphics Symposium UK*, pages 21–23. Citeseer, 2001.

[145] Santa Maiti, Debasis Samanta, Satya Ranjan Das, and Monalisa Sarma. Language independent icon-based interface for accessing internet. In *Advances in Computing and Communications*, volume 191 of *Communications in Computer and Information Science*, pages 172–182. Springer Berlin Heidelberg, 2011.

[146] Matthew Paul Huenerfauth. Design approaches for developing user-interfaces accessible to illiterate users. *MSc thesis, Dept of Computer Science, University College Dublin, Ireland*, 2002.

[147] I. Medhi, A. Sagar, and K. Toyama. Text-free user interfaces for illiterate and semi-literate users. In *Information and Communication Technologies and Development, 2006. ICTD'06. International Conference on*, pages 72–82. IEEE, 2006.

[148] Indrani Medhi, S.N. Nagasena Gautama, and Kentaro Toyama. A comparison of mobile money-transfer uis for non-literate and semi-literate users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1741–1750, New York, NY, USA, 2009. ACM.

[149] Beenish M. Chaudry, Kay H. Connelly, Katie A. Siek, and Janet L. Welch. Mobile interface design for low-literacy populations. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, IHI'12, pages 91–100, Miami, Florida, USA, 2012. ACM.

[150] Elsa Friscira, Hendrik Knoche, and Jeffrey Huang. Getting in touch with text: designing a mobile phone application for illiterate users to harness sms. In *Proceedings of the 2nd ACM Symposium on Computing for Development*, ACM DEV '12, pages 5:1–5:10, Atlanta, Georgia, 2012. ACM.

[151] Anuradha Bhamidipaty and P. Deepak. Symab: Symbol-based address book for the semi-literate mobile user. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction*, INTERACT'07, pages 389–392, Berlin, Heidelberg, 2007. Springer-Verlag.

[152] Madeline Plauché and Udhyakumar Nallasamy. Speech interfaces for equitable access to information technology. *Information Technologies & International Development*, 4(1):pp–69, 2007.

[153] Etienne Barnard, Marelie H Davel, and Gerhard B Van Huyssteen. Speech technology for information access: a south african case study. In *AAAI Spring Symposium: Artificial Intelligence for Development*, 2010.

[154] Victor Zue, Stephanie Seneff, James R Glass, Joseph Polifroni, Christine Pao, Timothy J Hazen, and Lee Hetherington. Juplter: a telephone-based conversational interface for weather information. *Speech and Audio Processing, IEEE Transactions on*, 8(1):85–96, 2000.

[155] Imane Taoufik, Hind Kabaili, and Driss Kettani. Designing an e-government portal accessible to illiterate citizens. In *Proceedings of the 1st international conference on Theory and practice of electronic governance*, ICEGOV '07, pages 327–336, Macao, China, 2007. ACM.

[156] Sebastien Cuendet, Indrani Medhi, Kalika Bali, and Edward Cutrell. Videokheti: Making video content accessible to low-literate and novice users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2833–2842. ACM, 2013.

[157] Archana Prasad, Indrani Medhi, Kentaro Toyama, and Ravin Balakrishnan. Exploring the feasibility of video mail for illiterate users. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 103–110, Napoli, Italy, 2008. ACM.

[158] Anita HM Cremers, Jacomien GM de Jong, and Johan S van Balken. User-centered design with illiterate persons: the case of the atm user interface. In *Computers Helping People with Special Needs*, pages 713–720. Springer, 2008.

[159] Steven Birnie, Jon Rogers, and Charlie Rohan. The pillar atm: Ncr and community centered innovation. In *Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces*, DPPI '11, pages 48:1–48:8, Milano, Italy, 2011. ACM.

[160] Zereh Lalji and Judith Good. Designing new technologies for illiterate populations: A study in mobile phone interface design. *Interacting with Computers*, 20(6):574–586, 2008.

[161] Agneta Gulz. Benefits of virtual characters in computer based learning environments: Claims and evidence. *International Journal of Artificial Intelligence in Education*, 14(3):313–334, 2004.

[162] Lingyun Qiu and Izak Benbasat. An investigation into the effects of text-to-speech voice and 3d avatars on the perception of presence and flow of live help in electronic commerce. *ACM Trans. Comput.-Hum. Interact.*, 12(4):329–355, December 2005.

[163] Z Grujic, B Kovacic, and IS Pandzic. Building victor-a virtual affective tutor. In *Telecommunications, 2009. ConTEL 2009. 10th International Conference on*, pages 185–189. IEEE, 2009.

[164] Samuel Alexander, Abdolhossein Sarrafzadeh, and Stephen Hill. Easy with eve: A functional affective tutoring system. In *Workshop on Motivational and Affective Issues in ITS. 8th International Conference on ITS*, pages 5–12. Citeseer, 2006.

[165] Krittaya Leelawong and Gautam Biswas. Designing learning by teaching agents: The betty's brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208, 2008.

[166] Christopher Mumme, Niels Pinkwart, and Frank Loll. Design and implementation of a virtual salesclerk. In *Intelligent Virtual Agents*, pages 379–385. Springer, 2009.

[167] Mazin Gilbert, Jay G Wilpon, Benjamin Stern, and G Di Fabbrizio. Intelligent virtual agents for contact center automation. *Signal Processing Magazine, IEEE*, 22(5):32–41, 2005.

[168] K Kamyab, F Guerin, P Goulev, and E Mamdani. context sensitive virtual sales agents. In *Proceedings of the Conference on Artistic, Cultural and Scientific Aspects of Experimental Media Spaces (cast 2001), FhG-Institut Medienkommunikation (IMK), Germany*, pages 79–83. Citeseer, 2001.

[169] Patrick Kenny, Thomas Parsons, Jonathan Gratch, and Albert Rizzo. Virtual humans for assisted health care. In *Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*, page 6. ACM, 2008.

[170] Poika Isokoski, Markus Joos, Oleg Spakov, and Benoît Martin. Gaze controlled games. *Universal Access in the Information Society*, 8(4):323–337, 2009.

[171] Ning Tan, Gaëtan Pruvost, Matthieu Courgeon, Céline Clavel, Yacine Bellik, and Jean-Claude Martin. A location-aware virtual character in a smart room: effects on performance, presence and adaptivity. In *Proceedings of the 16th international conference on Intelligent user interfaces*, pages 399–402. ACM, 2011.

[172] Rachel McDonnell, Sophie Jörg, Joanna McHugh, Fiona Newell, and Carol O'Sullivan. Evaluating the emotional content of human motions on real and virtual characters. In *Proceedings of the 5th symposium on Applied perception in graphics and visualization*, pages 67–74. ACM, 2008.

[173] Shawna CG Tan and Alexander Nareyek. Integrating facial, gesture, and posture emotion expression for a 3d virtual agent. In *Proceedings of the 14th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGames 2009 USA)*, pages 23–31, 2009.

[174] Yvonne Jung, Arjan Kuijper, Dieter Fellner, Michael Kipp, Jan Miksatko, Jonathan Gratch, and Daniel Thalmann. Believable virtual characters in human-computer dialogs. In *Eurographics*, pages 75–100, 2011.

[175] Nadia Magnenat-Thalmann and Daniel Thalmann. Virtual humans: thirty years of research, what next? *The Visual Computer*, 21(12):997–1015, 2005.

[176] Zerrin Kasap, Maher Ben Moussa, Parag Chaudhuri, and Nadia Thalmann. Making them remember, emotional virtual characters with memory. *Computer Graphics and Applications, IEEE*, 29(2):20–29, 2009.

[177] A Thatcher and M Ndabeni. Hci accessibility guidelines and illiteracy: developing a model of illiteracy and engagement with technology. *Ergonomics SA*, 17:13–24, 2005.

[178] Cesare Rocchi, Oliviero Stock, Massimo Zancanaro, Michael Kruppa, and Antonio Krüger. The museum visit: generating seamless personalized presentations on multiple devices. In *Proceedings of the 9th international conference on Intelligent user interfaces*, pages 316–318. ACM, 2004.

[179] Abby C. King, Timothy W. Bickmore, Maria Ines Campero, Leslie A. Pruitt, and James Langxuan Yin. Employing virtual advisors in preventive care for underserved communities: Results from the compass study. *Journal of Health Communication*, pages 1–16, 2013.

[180] Nadeem Ahmad. Virtual character based interactive interfaces for deaf and functionally illiterate users. In *Software Business. From Physical Products to Software Services and Solutions*, volume 150 of *Lecture Notes in Business Information Processing*, pages 223–224. Springer Berlin Heidelberg, 2013.

[181] Winston W Royce. Managing the development of large software systems. In *proceedings of IEEE WESCON*, volume 26. Los Angeles, 1970.

[182] David E Avison and Guy Fitzgerald. Where now for development methodologies? *Communications of the ACM*, 46(1):78–82, 2003.

[183] Christiane Floyd. A systematic look at prototyping. In *Approaches to prototyping*, pages 1–18. Springer, 1984.

[184] RN Burns and Alan R. Dennis. Selecting the appropriate application development methodology. *ACM Sigmis Database*, 17(1):19–23, 1985.

[185] Barry W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, 1988.

[186] Barry Boehm and Wilfred J Hansen. Spiral development: Experience, principles, and refinements. Technical report, DTIC Document, 2000.

[187] James Martin. *Rapid application development*. Macmillan Publishing Co., Inc., 1991.

[188] Jim Highsmith and Alistair Cockburn. Agile software development: The business of innovation. *Computer*, 34(9):120–127, 2001.

[189] Kent Beck. Embracing change with extreme programming. *Computer*, 32(10):70–77, 1999.

[190] Tiziana Margaria and Bernhard Steffen. Agile it: thinking in user-centric models. In *Leveraging Applications of Formal Methods, Verification and Validation*, pages 490–502. Springer, 2009.

[191] James P Womack and DT Jones. *Banish waste and create wealth in your corporation*. New York: Free Press, 2003.

[192] Mary Poppendieck. *Lean software development: an agile toolkit*. Addison-Wesley Professional, 2003.

[193] Kim B Clark. *Product development performance: strategy, organization and management in the world auto industry*. Harvard Business Press, 1991.

[194] Shawn Lawton Henry. *Just ask: integrating accessibility throughout design*. Lulu. com, 2007.

[195] Amy Koerber and Lonie McMichael. Qualitative sampling methods a primer for technical communicators. *Journal of business and technical communication*, 22(4):454–473, 2008.

[196] ISO HCDProcess. ISO 13407: Human-centred design processes for interactive systems. International Organization for Standardization, Geneva, Switzerland, 1999.

[197] ISO WD. Iso 9241-11. ergonomic requirements for office work with visual display terminals (vdts). International Organization for Standardization, Geneva, Switzerland, 1998.

[198] William Albert and Thomas Tullis. *Measuring the user experience: collecting, analyzing, and presenting usability metrics.* Newnes, 2013.

[199] James R Lewis. Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78, 1995.

[200] J Brooke. System usability scale (sus): A quick-and-dirty method of system evaluation user information. digital equipment co ltd. *Reading, UK*, 1986.

[201] Jeff Sauro. *A practical guide to the system usability scale: Background, benchmarks & best practices.* Measuring Usability LCC, 2011.

[202] James R Lewis and Jeff Sauro. The factor structure of the system usability scale. In *Human Centered Design*, pages 94–103. Springer, 2009.

[203] Sylvain Delisle and Bernard Moulin. User interfaces and help systems: from helplessness to intelligent assistance. *Artificial intelligence review*, 18(2):117–157, 2002.

[204] Taizo Nishimuro. Information literacy: how does it differ from traditional or computer literacy? *TechKnowLogia*, 1(1):13–14, Sep 1999.

[205] Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008.

[206] Thomas S Tullis and Jacqueline N Stetson. A comparison of questionnaires for assessing website usability. In *Usability Professional Association Conference*, Minneapolis, USA, June 2004.

[207] Kraig Finstad. The system usability scale and non-native english speakers. *Journal of usability studies*, 1(4):185–188, 2006.

[208] Brian Whitworth. Polite computing. *Behaviour & Information Technology*, 24(5):353–363, 2005.

Appendix A

Pre-Experiment Questionnaire

Pre-experiment questionnaire distributed among participants to evaluate their traditional and information literacy

1. How old are you (Number of years)?

2. How many school years have you attended? Please circle the corresponding number:

   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 >

   (a) Have you attended any informal training to get or enhance your reading/ writing skills? If yes, how many months?

3. What are your English language skills?

Read							
	No Knowledge	○	○	○	○	○	Fluent
Speak							
	No Knowledge	○	○	○	○	○	Fluent
Write							
	No Knowledge	○	○	○	○	○	Fluent

4. What are your Swahili language skills?

Read							
	No Knowledge	○	○	○	○	○	Fluent
Speak							
	No Knowledge	○	○	○	○	○	Fluent
Write							
	No Knowledge	○	○	○	○	○	Fluent

5. Have you ever used a computer (PC)?
   - Just sometimes (a very few times in a year)
   - From time to time (a very few times in a month)
   - Quite often (weekly)
   - Usually (quite daily)

   (a) If you often or usually use a PC

      i. How many months or years have you been using a PC?

      ii. What kind of system you used?
         - Desktop computer
         - Laptop / Notebook
         - Touch pad / tablet

      iii. Have you ever submitted any online application? If yes How many times almost?

      iv. Have you ever made your own resume?

      v. Have you ever performed any online financial transaction?

      vi. Do you have a personal email address?

   (b) If you used computer very rarely, sometimes or from to time

      i. Where did you use the computer?
         - Own
         - Through a friend
         - At job

- In net cafe

ii. Are you comfortable in using mouse or Touchpad?
- No
- With little difficulty
- Yes

iii. Can you write through a keyboard?
- Very easily
- With little difficulty
- Cannot write

6. How is your vision? (How do you see?)
- Normal
- Corrected to normal (use of glasses)
- I can see things better up close than far away
- I can see things better in the distance but not up close
- I generally can't see well

Question 5 has two parts, Either answer 5(a) or 5(b) but not both.