Project Number:	**IST-2000-25420**
Project Title:	**FASTY - Faster Typing for Disabled Persons**
Dissemination level:	**PU***

Deliverable:	**D2.7**		
Title:	**3rd Edited Annual Report for publication**		
Contractual Date:	**2003-12-31**	**Date of Delivery:**	**2004-03-31**
Work-Package:	**WP 2**		
Nature:	**R******		
Authors:	**UU**		

FASTY Deliverable Numbering D2.7/R/PU/UU/P4

Abstract:

This third edited annual report for publication is a summary of the FASTY project outcomes at the end of the third year of the 3 years project.

LIST OF AUTHORS

Deliverable Numbering:

ID	*Official Deliverable ID (DX.Y)*	
NATURE	*Nature of Deliverable*	
DLEVEL	*Dissemination Level*	
RESPID	*Responsible Partner ID*	
STATUS	*Wn*	*Working document, version n*
	Pn	*PRELIMINARY complete version for discussion, version n*
	Dn	*DRAFT, before peer review*
	F	*FINAL VERSION, after peer review*

*Dissemination: INT-Project internal, CO-Confidential, RE-Restricted access IST-IST participants, FP5-FP5 participants, PU-Public document

**Nature: P-Prototype, R-Report, D-Demonstrator, S-Specification, T-Tool, O-Other

UU

Uppsala University, Department of Linguistics and Philology

S-75120 Uppsala

Villavägen 4

Sweden

Contact: Professor Anna Sågvall Hein

T: +46-18-471/1412

F: +46-18-471/1416

Financial/Administrative co-ordinator

Dr. Wolfgang Zagler

Favoritenstrasse 11/366-1B A1040 Vienna, Austria

Phone: +43-1-58801-76611 Fax: +43-1-58801-36697

E-mail: fasty-fortec@fortec.tuwien.ac.at

Project website: http://www.fortec.tuwien.ac.at/fasty

Contents

1 Introduction

Verbal communication is a vital need for humanity. In the society of today, computers play an increasing role as a communication tool. Generation of text may, however, be very cumbersome for persons with physical or linguistic disabilities and the process of entering text may thus be slow. For instance, while non-disabled writers have a typing speed of some 200-300 characters a minute; the typing speed of a user operating the keyboard with a mouth stick is not higher than 75-120 characters a minute. In these situations word prediction may be of great help to speed up the text generation rate and improve the quality of the produced text. This is the strategy that has been chosen in the FASTY project.

The project is scheduled for the period from January 2001 to March 2004. Below we will give a brief description of the aims of the project and the achievements made during the third project year.

2 Aims

The concrete goal of FASTY is the creation of a system for increasing the text generation rate of disabled persons by so-called predictive typing and dedicated advanced input devices. A prediction system attempts to predict subsequent portions of the text by analysing the text already entered and using frequency data on the vocabulary of the language. Character-by-character text entry is thus enhanced by the possibility of entering whole words and portions of words as they are proposed by the system. The selection of an alternative should be made by means of a single keystroke. Complementary to the presentation of the proposals on the screen, they will be read aloud by means of speech synthesis. The success of a system of this kind can be measured in terms of keystrokes that are saved using the predictions as compared to traditional character-by-character input. FASTY aims at a keystroke saving rate above 50%. Experiences that were made during the first project year indicate that the linguistic quality of the text will also benefit from using the prediction system.

FASTY is an intelligent system that uses methods of Natural Language Processing (NLP), Artificial Intelligence (AI), a self-adaptive user interface, and linguistic resources such as dictionaries and grammars. The FASTY text prediction system applies to four languages: Dutch, French, German and Swedish. The future inclusion of additional languages has also been taken into consideration. The multilingual aspect is reflected in the design of the system.

User involvement in the project has been strong during the whole period. It is ensured by means of a user panel. The user needs are analysed subjectively by intensive interaction with this user panel. There are two kinds of users in the panel: primary end-users and secondary users such as pedagogues, therapists, carers and family members. The user panel played a very important role during verification and validation of the prototype systems.

An important aspect of the project is the design and development of a dedicated interface adapted to the needs of the users. The user interface design and the features of the predictor program aim at a wide coverage of primary users (various disabilities) and secondary users (various roles in supporting the disabled person). Self-adapting parameters and flexible configuring should ensure a high degree of usability, user friendliness and accessibility to the system. A user simulation tool will be used in testing the system and adapting it to different users.

Innovative and ergonomic user interfaces for various existing input methods (standard keyboard, on-screen keyboard, scanning) are developed together with the predictor thus minimising time and effort for selecting the desired word from a selection list presented on the screen. In addition, a special pressure sensitive switch/keyboard is developed and used to improve the user interface, UI.

A first prototype (PT1) was developed, which contains the language component, the user interface and an Adjustment Tool. A test-phase with 20 users using PT1 has been finished. A prototype two (PT2) based on the results from the user-tests has been developed. A second test-phase with a huge number of users has been finished. The results will influence the further development in a direct way.

Dissemination and Exploitation have been playing a central role throughout the project. The Technological Implementation Plan (TIP), developed as a preparation for the exploitation plan, has been updated. After successful finishing of the project the consortium will co-operate in order to come up with a commercial product.

3 Consortium

The FASTY consortium consists of nine partners from four countries: Austria, Belgium, Germany and Sweden.

There are six principal contractors:

- **fortec** - Research Group for Rehabilitation Technology / Project Coordinator
- **ÖFAI** - Österreichisches Forschungsinstitut für Artifical Intelligence
- **FTB** - Forschungsinstitut Technologie - Behinderthilfe

- **UU** - Uppsala University, Department of Linguistics
- **MULT** - Multitel ASBL
- **IGEL** Elektronische Kommunikationshilfen GmbH

and three assistant contractors:

- **ELI** - Seraphisches Liebeswerk, Elisabethinum Axams
- **IKuT** - Ingenieurbûro für Kunst und Technik II
- **FUNDP** - Facultés universitaires, Notre-Dame de la Paix

fortec is the project coordinator and responsible for managing the project, and for system architecture and internal interfaces.

Öfai is responsible for the implementation of the language components and the provider of language resources for German. It also has the responsibility for system integration and prototyping.

FTB is responsible for user involvement including ethics, quality assurance, and user validation. It also has the responsibility for the implementation of the user components.

UU is responsible for dissemination and public relations. It is also the provider of language resources for Swedish and responsible for the grammar based prediction.

MULT is responsible for verification and redesign of the system. It is the provider of language resources for French and Dutch and of speech synthesis solutions for all languages.

IGEL is responsible for the exploitation of the system and for technical implementation. It also contributes to specifications, architecture and user involvement.

ELI's main task is to provide feedback to the developers. It also participates in user related tasks such as prototype testing and assessment of user needs.

IKut is the developer of new input devices. It has associated with it a local user group.

FUNDP provides an interface between the users and the developers. It participates in the analysis of user capabilities and needs, and assists disabled people in using the product.
In addition, there are a number of sub contractors. A listing of the project partners with contact information is to be found in Appendix.

4 Innovative Aspects

Predictive typing systems for English have proved to be useful and efficient for a long time, but for other European languages there are only a few such systems powerful enough to improve the communication rate for disabled persons. Adapting the English programs to highly inflecting languages, like Dutch, French, German and Swedish, by replacing the English dictionaries, often leads to a significant reduction of the keystroke saving rate. These effects are due to the simplistic language description that is used for predicting English text and that fails to predict the correct inflectional form of a word as required by inflectional languages. The English language description is, as a rule, limited to frequency data on individual words (unigrams) and sequences of words (bigrams, trigrams). Attempts have been made in research systems for Swedish and Spanish to use a more elaborate language description, including n-grams of word classes [1] and syntactic grammars [2]. The experiences made in these projects are taken into account in the FASTY project. They do not, however, present solutions that will ensure a keystroke saving rate of above 50% for the FASTY languages. An additional problem with most of the FASTY languages is the fact that new compounds can be created on the fly, thus making it hopeless to strive for a complete lexicon. Other methods need to be employed for coping with dynamic word formation processes. Being able to cope with compounds, even if they are new, is of great importance, since compounds are usually rather long words and failing to predict them can cause a significant drop of the keystroke saving rate. Since no existing word prediction system is able to handle new compounds, this aspect of the FASTY system is a true innovation.

At a general level, the innovative aspects of the FASTY predictor are represented by:

- The predictive power of the prediction engine that is based on a sophisticated language component
- A dedicated, flexible and adaptable user interface that is an integral part of the system
- New input devices

In particular, the innovative nature of the FASTY predictor is reflected in the following features:

- Prediction of compounds
- Prediction of proper inflectional forms based on the use of parsing
- Generic algorithms to ensure cross-language portability
- Dictionaries based on general language corpora and on the users' own texts
- Adaptation of the dictionaries based on actual use of the predictor
- Initially supported languages: Dutch, French, German, Swedish
- User interface that is an integral part of the predictor and
    - Adaptable by primary and secondary users

  - o Capable of using different kinds of input devices
  - o Automatic adaptation to the performance of the user
- New input device
  - o Pressure sensitive switch
  - o Pressure sensitive keyboard

5 The FASTY System

The specification of the FASTY system is based on a market study, a study of available technology and a study of the users' particular needs. Special care has been taken do define an open and flexible architecture that is adaptable to the users' needs. Another focal point has been do define a generic system that is not geared to any specific language. Thus the language components will be independent of the operating system and the prediction strategies will be evaluated with regard to all the FASTY languages. The system should also be portable to platforms other than MS-Windows.

The main components of the FASTY system can be divided into three parts:

- **Runtime system**, comprising the User Interface (UI) and the Language Component (LC)
- **User Adjustment Tool (UAT)**
- **Four developer tools,** including the User Ability Assessment Tool (UAAT), the Text Collection Tool (TCT), the User Simulation Tool (UST) and the Simple Word Prediction program (SWP)

An overview of the parts is to be seen in the figure below.

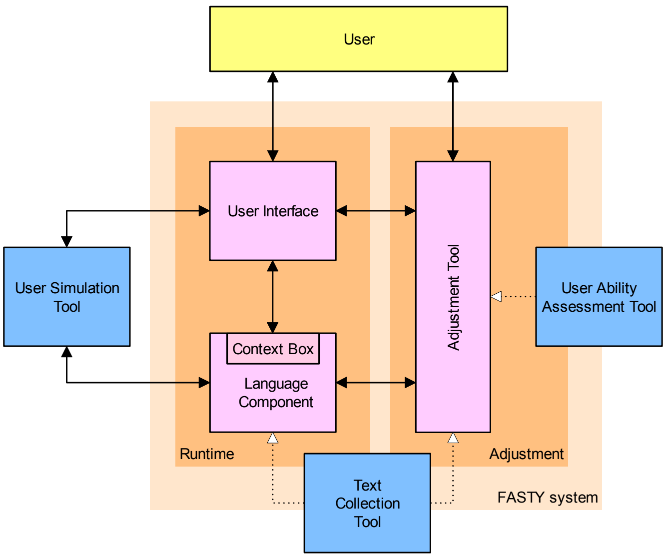

Fig. 1 General System Structure

Dotted lines indicate no direct interfaces, but used information and the possibility to use part of the code or methods in later versions of the FASTY system

During the third project year the language component has been integrated with the user interface, and the first prototypes were released. The prototypes contain, beside the kernel and drivers of the runtime system, the Adjustment Tool, and the language component. They represent a fully functional word prediction system for German, Swedish, Dutch and French.
Below we give a brief functional description of the different components. For further details, see [3].

5.1 The User Interface

The acceleration of text-input has two main starting points: A good prediction engine and a good input-system in combination with a good user interface. The purpose of the User Interface (UI) is to provide input and output facilities, to interact with the Language Component, and to support users in their effort to adjust the system to their individual needs by offering a variety of options concerning colour, font size, position, sorting etc.
The first prototypes of the UI feature input facilities for keyboard users. This includes standard keyboards as well as keyboard emulating equipment.

5.1.1 User Interface software

The FASTY system uses standard interfaces like keyboard interface, mouse interface, serial or parallel interfaces that humans use to enter data into a computer. The drivers to operate those interfaces are part of the particular used operating system. In addition to these drivers special purpose FASTY drivers are needed to achieve a consistent data format that serves as input to the FASTY kernel no matter what kind of physical input device is currently connected to the computer.
In general a FASTY driver has to perform mainly three tasks:

- Initialisation of the hardware interface on start up
- Data transfer
- Releasing occupied system resources when the job is done

The FASTY runtime system will use sets of different drivers. An entry of the driver name in the systems initialising file (INI file) determines, which driver will be loaded during program start up and used during runtime. All drivers make a consistent set of function available to the FASTY system. They deliver the following information:
Input Driver:

- Virtual-key codes
- Key states (key up, down or toggled)

- Pressure values
- Cursor position, cursor move and other adequate data of the pointing device

Output Driver:

- Language ID
- Prediction list
- Part of speech information

Program Driver:

- Caret position
- Cursor position
- Focus change
- Active menus
- Active dialogs

5.1.2 Pressure Sensitive Devices

Parts of the FASTY user interface are new devices that enable the users to use their motor abilities better for computer input. There are two areas of applications of the pressure sensitive devices within the FASTY project: single switches and keyboards. The switches are made for persons who use scanning methods with special devices or on-screen-keyboards for text input. For persons, who use standard or special keyboards, the pressure sensitive devices allow faster access to specific keyboard functions and/or additional functions.

The idea behind the Pressure Sensitive Keyboard is to evaluate the pressure as additional information in the input process. To press a key or not to press it is one bit of information. By evaluating the pressure it is possible to gain more information from pressing a key. E.g. to press weakly or strongly can be used for different input actions.

The principle of the pressure sensitive sensors is as follows: The user applies pressure on the sensor: hence the *Force Sensing Resistor*-sensor changes its resistor value. This resistor value is converted into digital data by the FASTY-SensorBox. The SensorBox transmits the data to the computer, where the FASTY software evaluates the values.

5.1.2.1 The SensorBox

None of the commercially available interfaces for connecting input devices to PCs are able to analyse the output signal of pressure sensitive sensors. A new hardware interface is therefore being developed within the FASTY project: the SensorBox. The SensorBox measures the resistor values of pressure sensitive elements and converts them into digital data. This data is sent to the PC for further evaluation. The SensorBox provides up to 8 inputs for sensors. The range of measurement is adapted in such a way that very soft pressure can be used as well as quite hard pressure like in a foot switch.

Fig. 2 SensorBox 2nd prototype

5.1.2.2 The keyboard

The Pressure Sensitive Keyboard (PSK) is a standard PC keyboard with a pressure sensitive element under each key inside the keyboard case. This device allows the evaluation of the pressure on the keys and uses the pressure information for additional functions. At the present time the PSK exists only for a standard PC keyboard. However, this covers most of the potential users. It is planned to build also enlarged PSKs in the future. This would cover the next big group of keyboard users.

Fig. 3 Pressure Sensitive Keyboard prototype

For more details, especially on the technical issues, see [3].

5.2 The Language Component

Below we focus on the functionality of the language component. For further detail, see [4].

5.2.1 Overall Specification

A statistic language model based on word n-grams and part-of-speech tag n-grams in conjunction provides FASTY's core functionality. Moreover, the possibility to create user specific dictionaries both during a session and on the basis of previous entered texts, serves as a method for further increasing the prediction accuracy.

The variety of languages to be supported and methods to be integrated into the FASTY system demands a modular architecture. The combination and integration of prediction components needs to be handled in a very flexible way, for a variety of reasons:

- Different languages may put different emphasis on different modules, so it must be possible to arrange these modules in a different way.
- The effects of the different prediction methods are not yet known precisely; experimental adjustments as well as parameter tuning have to be possible.
- Different application scenarios (varying from text writing up to spontaneous dialogs) may require different combination and weighting of components.
- Adaptation to users with different degrees and types of disabilities will also be required.

Thus the backbone of the linguistic prediction component of the FASTY system will be a controller that is flexibly driving the different prediction modules and combining their results. Thus it will be easy to optimise the overall prediction behaviour and also adaptation of FASTY to another language without modifying the whole system will be made possible.

5.2.2 Methods used in existing prediction systems

When one considers methods for saving keystrokes in text typing, one has to differentiate between keystroke-saving methods in the UI and methods involving linguistic or statistical prediction of words, word sequences and phrases. Here we will put our emphasis on the latter. However, for the sake of completeness a short list of methods belonging to the User Interface side is given below:

- Automatically *inserting a space* after every predicted word accepted by the user. This method compensates for the extra keystroke needed for selecting the prediction.
- Automatically *removing preceding white space* immediately before punctuation characters (and inserting the appropriate amount of spaces afterwards). This method complements the previous one, as the need for the user to backspace the automatically inserted white space is alleviated.
- *Auto-capitalisation*. This method in fact also needs at least some linguistic knowledge. It is listed here just because of the requirement to be able to change characters the user

already has typed. Auto-capitalisation may occur after sentence ending periods, on words recognised as proper names or (in some languages, e.g. German) on nouns in general.

5.2.2.1 String-based statistical methods

All systems on the market that we are aware of use some kind of frequency statistics on words and (sometimes) word combinations. Given a prefix of a word, with a frequency annotated lexicon the most probable continuation(s) of that word can be retrieved easily. Sometimes, not only word-based frequency counts are maintained (unigrams), but also bigrams and even trigrams are used for enhancing prediction accuracy. N-gram language models are widely used in speech recognition systems, and their benefits are also exploited in some predictive typing systems. The key observation behind this kind of models is, that the probability of a word occurring in the text depends on the context.

5.2.2.2 Syntactically motivated statistics

The superiority of n-gram based predictions over simple frequency lexicons stems from the fact, that n-grams are able to capture some of the syntactic and semantic regularities intrinsic to language. However, a severe drawback of word-based n-grams is, that, even with very large training texts, the data still is rather sparse, and thus in many actual cases during prediction no information is available. The usual technique to cope with syntactic regularities uses class-based n-grams (usually n=3), the classes being defined by the part-of-speech information of a tagged corpus. Copestake [5] reports on an improvement in KSR of 2.7 percent points by just taking part-of-speech bigrams into account. A good description on the integration of part-of-speech trigrams into a statistical word prediction system for Swedish is given in [1].

5.2.2.3 Capturing semantics with statistics

For a human language user it is obvious that in a given context some words are more probable than others just because of their semantic content. Factors influencing word probability due to semantics are (among others):

- The user and the type and topic of the text s/he writes (global factors)
- Constraints due to the lexical semantics of words (e.g. sub-categorisation requirements); these are local factors that mostly operate on sentence level.

Collocation analysis (in a broader sense, not reduced to idioms only) can reveal some of these dependencies. However, very large corpora are needed. Rosenfeld [6] uses the concept of “trigger pairs” to capture these relationships statistically (basically these are bi-grams of words occurring together in a window of a certain size in a corpus). If a word that has been recently entered occurs in such a trigger pair the probability of the other word of the pair should be increased. Recency, as implemented as a heuristic in some prediction systems, can be seen as a self-trigger and is a (rather crude) measure to exploit semantic or topical appropriateness of a word.

5.2.2.4 Rule-based approaches

Several methods of integrating grammar rules into statistics based prediction have been tried, but none of them had made it into a commercially successful product. Such integration, however, is seen as a major challenge in the FASTY system.

5.2.3 Linguistic components and resources for text prediction

Basic to our approach is the modular architecture of our system. In addition to the flexibility such an approach provides for the adaptation to different languages, application scenarios and users – as described in the introduction – it also ensures robustness and graceful degradation in the case one module should be missing or fail. Furthermore, this type of architecture allows for the possibility of exploring various more advanced – and albeit more risky – methods without endangering the successful implementation of the language component in case some of these methods should not prove successful.

The core of the system will be a module based on the prediction of word forms due to their absolute frequency and the probability of their associated part-of-speech. Such a module is state-of-the-art and guarantees a basic performance. A number of other methods to improve prediction quality will be investigated. All methods will be evaluated with respect to their performance for different target languages and language specific phenomena (e.g., compounding). Those that prove to be successful for one or more of the target languages will be integrated with the core component – either alone or in combination with others.

5.2.3.1 General word n-gram-based Prediction

As stated above, predictions based on frequencies of word sequences are usually more reliable than predictions solely based on simple word frequencies. Frequency tables of word bigrams are thus used as a base in the FASTY language model. As is customary, the bigram model is however supplemented by simple word frequencies. This is due to that no matter how much data is used for extracting bigram frequencies, there will always be a problem of sparse data - most bigrams will have low frequencies and many possible word sequences will not be attested in the training material. A common solution, also implemented in the FASTY language model, is to *interpolate* the probabilities obtained from using larger n-grams (here: bigrams) with the probabilities acquired from smaller n-grams (here: unigrams, i.e. simple word frequencies). The process of interpolation is further described in [7].

5.2.3.2 Part-of-Speech n-gram-based Prediction:

The FASTY language model is further based on part-of-speech frequencies. Since a part-of-speech tag captures a lot of different word forms in one single formula it is possible to represent contextual dependencies in a smaller set of n-grams. One major advantage of making use of part-of-speech frequencies is thus that the problem of sparse data is reduced and a larger context may be taken into consideration. The FASTY language model uses frequencies of part-of-speech tag trigrams, which are supplemented with frequency data of smaller part-of-speech n-grams (uni- and bigrams).

5.2.3.3 User- and Topic-specific n-gram-based Prediction:

The FASTY language model is adjustable to the language of specific users in two respects. It applies *short-term learning* whereby the words from the current text are dynamically added to user-specific uni- and bigram frequency lists. During prediction the user-specific frequency lists and the general frequency lists are combined using relative weights. In this way, new words that are repeated (e.g. proper names) can be predicted on their second occurrence. Further, the system will provide for *long-term learning*, by making it possible to permanently save changed user dictionaries from time to time.

Topic-specific words and expressions will be possible to generate from previously written, and electronically readable, texts.

The use of several user- and topic-specific lexicons at the same time will be allowed and all activated lexicons will be searched. Words and expressions with highest probabilities will be offered in a prediction list.

5.2.3.4 Morphological processing and Backup Lexicon:

As stated above, the part-of-speech n-grams provide means to account for larger contexts by representing the distribution of word forms at a generalized level. For this to work though, the language model requires information about the part-of-speech of each word form. Further, the grammar module, described below, bases its analysis on a morpho-syntactic description of the input word forms. Put together, the FASTY language model needs some kind of lexicon, providing all relevant information. Retrieving information from a huge lexicon may be very time-consuming, even though it is done automatically. Special care has therefore been taken to provide a storage format that is easy to search and further, that makes it possible to compress the enormous amount of data to a manageable size. Such an implementation was provided as a prototype at the initiation of the project and has been upgraded and adjusted to suit the FASTY language component. Further, the morphological data required has been gathered for all FASTY languages.

5.2.3.5 Abbreviation Expansion:

Abbreviation expansion is a technique in which a combination of characters, an *abbreviation*, is used to represent a word, a phrase or a command sequence. Whenever the user types a predefined abbreviation, it is expanded to the assigned word, phrase or command sequence. The abbreviation module is integrated in the prototype language component. The integrated version has the following basic functionality:

- Given a prefix, all abbreviation codes starting with that prefix is returned
- Given an abbreviation, the expansion string is returned
- Given an abbreviation and its expansion the system stores that abbreviation in an abbreviation table

Lists of useful abbreviations are to be entered by the user or its care person.

5.2.3.7 Grammar checking as a filter of suggestions:

The FASTY language component is further based on a grammar-checking module. While the n-gram based prediction modules never consider contexts exceeding a limited number of words, the grammar-based module may take an arbitrarily large sentence fragment into consideration. The grammar module does not by itself generate any prediction suggestions, rather it filters the suggestions produced by the n-gram model so that the grammatically correct word forms are presented to the user prior to any ungrammatical ones.

Input to the grammar module is a ranked list of the most probable word forms according to the other language components. The grammar module will assign a value to each word form based on whether the word form is confirmed (grammatical), turned down (ungrammatical) or outside the scope of the grammar description. Based on these three values the word forms are then re-ranked whereby the grammatical suggestions are ranked the highest and the ungrammatical are ranked the lowest. Since only a subset of the re-ranked suggestions will be presented to the user, the lowest ranked word forms will not be displayed. This way grammatically impossible suggestions will hopefully not be presented at all, leaving room for possibly intended continuations.

The grammar descriptions are however not complete but covers selected constructions that are identified as crucial from a prediction point of view for the individual languages of the project.

Typically, they contain constructions with fairly fixed word order and feature constraints. Examples of such constructions are nominal phrases, prepositional phrases and verbal clusters. Sentence initial position and the placement of the finite verb are further focal points. Grammar rules have been written for all FASTY languages, albeit grammar rules will be further developed during the third project year.

For a system to analyse input texts in relation to a grammar description, special software, such as a *parser* is required. The Swedish partner supplied this at the initiation of the project. A description of the core-parsing engine used in the project can be found in [8]. In most applications though, a parser takes whole language structures as input (usually sentences). In the context of word prediction, the parser must allow for language structures that are about to be produced and thereby only are fragmentary. In other words the parsing process must be step-wise and there must be means to dynamically output the analysis made so far. The parser is therefore adjusted to these conditions and made compatible with the rest of the FASTY system.

5.2.3.8 Compound prediction:

In three of the FASTY languages: German, Dutch and Swedish, compounds constitute a group of words that is particularly hard to predict within a word prediction system. In these languages compounds can be productively formed to fill a contextual need. It is of course impossible to predict such a word formation by means of traditional n-gram frequency counts. On the other hand, compounds tend to be long words, which means that successful prediction would save a great deal of keystrokes. Within the FASTY language model, compounds have hence been given a special treatment. Since compound prediction is a true innovation in word prediction systems, the way to infer new compounds from the input evidence has been subject to research. The solution has been not to predict a productively formed compound as a whole, but to predict its parts separately. More specifically, the current implementation supports the prediction of right-headed nominal compounds, since these, according to a corpus study of German corpus data, are by far most common.

The split compound model provides two quite different mechanisms for predicting the respective parts of a compound, i.e. modifier (the left hand side of a compound) prediction and head prediction (the right hand side of a compound). Below we will give a simplified description of how the model functions. Since the system has no means of knowing when a user wants to initiate a compound, the prediction of modifiers is integrated with the prediction of ordinary words. If the user selects a noun that has higher probability of being a compound modifier, the system assumes this use was intended and starts the prediction of the head part instead of inserting the default white space after the selected noun. The head of a compound determines the syntactic behaviour, and the basic meaning of the compound as a whole. Hence, we may expect a productively formed compound to appear in the same type of contexts as the head does when it functions as an independent word. When predicting the head, the system therefore makes use of the word preceding the modifier, as if the modifier wasn't there. Assume the user has written *en god äppel (a tasty apple),* and intends to write *en god äppelpaj (a tasty apple pie).* When searching for possible compound continuations, the system will then search for bigrams with the first position held by *god,* and if the training corpora contained instances enough of the sequence *god paj, paj* is suggested as a possible head of the compound. Further, the head prediction model gives precedence to words that, in the training material, functioned as heads in many compounds. According to studies of German and Swedish compounds, some words occur much more often in compounds as heads, than other words. A secondary feature that has been explored is the semantic relation between the modifier and the head. If the semantic class of the modifier is known (for instance *apple* above may be assigned to a class containing fruits and berries), this information may be used to search for probable heads (following the given example these may be words belonging to classes of baked and cooked things).

5.2.4 Interaction of components and control structure

The operation of the Language Component (LC) is driven by the User Interface (UI). Depending on the request type, the LC returns one or more values as a response to the UI (e.g. the *n* most likely predictions).

A central part of the interaction between LC and UI is the *Context Box*. The role of the context box is to provide a repository for the context data needed by the prediction component. The context box contains textual data that may be manipulated by the UI via interface functions and LC-internal data structures that are not visible to the outside. The size of the context box is limited, it is only big enough to store the context needed by the predictor. It is not intended to be a cache of the whole file the user is editing. (If such caches are needed for some reason they should be handled separately by the user interface).

To manipulate the content of the context box, interface functions are provided to:

- Extend the context box by a character (or a string)
- Remove the last n characters from the context box. Note that for syntactic constraints on prediction the current sentence is sufficient as context.
- Clear the context box
- Replace the whole context by a string
- Insert an accepted prediction into the context box. This amounts to removing the prefix and inserting the selected string.

At any time the content of the context box changes, the relevant portion of its text buffer is (re)tokenised.

The *Controller* receives requests from the User Interface and is responsible for:

- Extracting the input data required for the different prediction components from the Context Box
- Selecting which prediction components to use (depending on the current parameter settings)
- Feeding the different prediction components with the appropriate input
- Possibly enriching the Context Box with data returned from some components (e.g. part-of-speech information)

The *Prediction Generator* receives the predictions made by the different components, together with their probabilities, and combines them to a prediction list, which is delivered to the User Interface. How the Prediction Generator comes up with the combined prediction list depends on:

- Parameter settings, which may be user and language specific.
- Interpolation weights that have to be determined empirically.

Each of the components relies on language specific resources, where some are shared between different components. Also the possibility exists, that a component uses the results of other components, e.g. grammar-based prediction uses compound analysis and morphological processing. For further information please see [3].

5.2.5 Speech Synthesis

For users with dyslexia or other language impairments, it may be hard to recognize an intended word form among the prediction suggestions, since there may be problems distinguishing similar words from each other. The same may hold for users with bad eyesight. Therefore most of the current word prediction systems on the market make use of a speech synthesizer, providing an audible presentation of the suggested word forms.

The speech synthesizer used in the FASTY system is a concatenation of a grapheme to phoneme converter (a program translating letters to a phonemic representation) and a phonetiser that converts phonemes into sound.

The conversion from letters to phonemes, is based on a so-called decision tree, i.e. a machine-learning algorithm stemming from the field of Information Theory. By means of this technique, rules stating how letters should be mapped to phonemes may be inferred automatically from a training dictionary in which word forms are listed along with their phonemic descriptions.

The award winning MBROLA phonetiser, made available through the Multitel partner, performs the second conversion, from phonemes to actual sound. MBROLA bases its speech synthesis on diphones, which means that it takes into account how the pronunciation of a phoneme is influenced by the preceding and succeeding phonemes. More information on MBROLA synthesizer may be found at http://tcts.fpms.ac.be/synthesis/mbrola.

5.3 User Adjustment Tool

Before a user starts working with the FASTY system, it has to be adjusted to her or his specific needs and situation. The adjustment consists in the setting of a large number of parameters and options. This is not a trivial task, and it will be carried out in co-operation between the primary user and the carer. The Adjustment Tool offers support in this process. The Adjustment Tool contains some functionality based on functions of the User Ability Assessment Tool, but also some extended features.

5.3.1 Definition of the look of the FASTY Adjustment Tool

The general structure of the Adjustment Tool is similar to modern Windows adjustment programs. On the left side is the tree of all possible adjustable main parameters. It is possible to define different user profiles. If a main parameter is selected on the left side, the right side opens a number of different parameters, again organized as a tree structure.

The structure of the Adjustment Tool reflects the functional range of the prototype. The main window contains the user administration and a set up menu with the main items: Input, Output and Prediction. The Input menu is divided into two parts: one displays details of the initialised drivers, and the other offers selections how to choose from the prediction list. The Output menu allows adjustments of graphical display, sound and speech.

Settings for the graphical display:

- Prediction window position
- Sorting
- Font and font size
- Amount of displayed predictions
- Colour settings

Settings for the Sound Output allow the assignment of events and sounds. Four assignments are available.

- A selection has been performed

- New predictions are available
- Autoexpansion has been performed
- No more predictions are available

Settings for Speech Output enable to adjust speech options and the assignment of event and speech output. Two items can be selected to be spoken, the wordlist and the chosen prediction.

The Prediction menu covers settings related to the language component. The general settings allow the control of:

- Dialog language
- Prediction language
- Handling of blanks in combination with punctuation

But also advanced settings, which require profound knowledge of the predictor, can be performed in this menu. Besides, the general settings the maintenance of dictionaries and abbreviation lists are managed in the prediction menu.

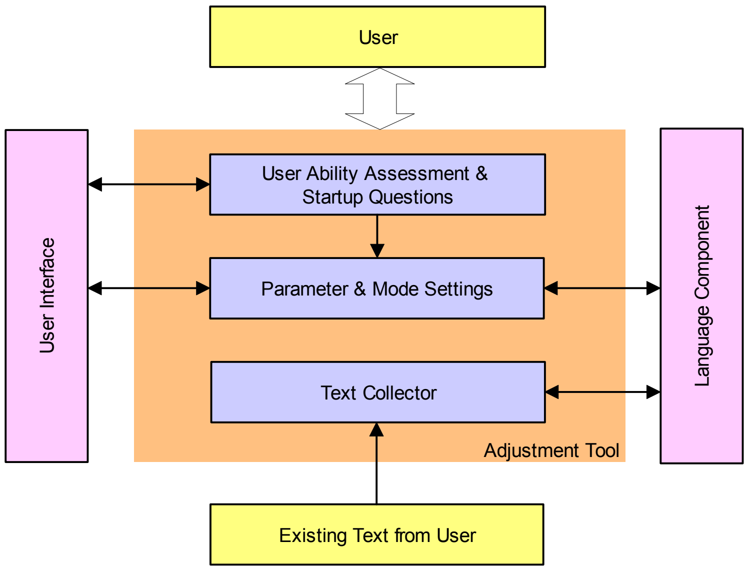

Fig. 4 General Structure of the Adjustment Tool

5.3.2 User Manual and Tutorial

Two parts of the FASTY system that support the users in getting to know the relevant system features and in adapting the system to their needs are the User Manual and the Tutorial. In order to make them accessible to several disabled computer users, they are written in HTML and can be accessed by standard or alternative browsers.

The FASTY User Manual is realized as a HTML-document to serve the following main purposes:

- Easily extendable and adoptable content of the help-files for different versions of the FASTY system.
- Easily editable, fast to view for errors.
- Available for the majority of platforms.
- Easy compilation into a windows-help file available from within the running FASTY system.

The FASTY Tutorial is also realized as a HTML-document. It has for objective to allow the users to familiarize themselves with the FASTY software. The training is based on exercises and examples of configurations through which the user can surrender account of the multiple possibilities offered. The Tutorial should be seen as a complement to the User Manual. The combination of the two tools assures an optimal exploitation of the software and prevents the user from uncomfortable situations. The exercises and examples are proposed according to three categories:

1) How to configure the software according to the user's system of entry
2) How to define the linguistic performances of the software
3) How to configure the window of prediction

5.4 Developer tools

As mentioned above, the FASTY system comprises four developer tools, i.e.

- User ability assessment tool, UAAT
- Text collection tool, TCT
- User simulation tool, UST
- Simple word prediction tool, SWP, for the developers of the language component

5.4.1 User Ability Assessment Tool (UAAT)

A User Ability Assessment Tool (UAAT) was developed to collect data about basic user performance, such as typing speed, or reaction time. The program supports standard input devices as well as special input devices connected via serial port and the prototype of the pressure sensitive keyboard developed by IKuT. The collected information is used to determine the applicability of further tests to the situation of the user. It is also used to get an impression of the user's current hardware and software status and the way text input is written.

The results from the questionnaires and the User Ability Assessment Tool do not show a uniform picture of the potential users. It is a wide variety of abilities and demands. Although FASTY will not contain communication assistance, the desire for an appropriate support was expressed.

5.4.2 Text Collection Tool (TCT)

The objective in the FASTY project has been to create a linguistic resource reflecting the language used by the target users - a resource that has been used to supplement the standard project corpora in the compilation of linguistic resources. The texts have primarily been collected from members of the user panel by means of the Text Collection Tool (TCT) that was developed during the first project year. The TCT eases the collection process by automatically retrieving relevant textual material from the hard drives of target computers. In order to protect the privacy of the participating users, the TCT replaces personal data, such as names and addresses, with neutral tags. Apart from texts retrieved from the members of the user panel, texts have been collected from the Internet, which has been searched for websites maintained by motor impaired persons. The resulting text collection comprises more than nine hundred text samples in German, French and Swedish. Throughout the collection process special care has been taken to fulfil the ethical guidelines.

5.4.3 User Simulation Tool (UST)

The aim of the user simulation tool is to help the user interface and language component developers to evaluate different algorithms and settings. Later commercial versions may help carers to find the best settings for the primary users in an iterative way without burdening the primary user herself with the testing of all possible settings and options.

The UST is composed of both hardware and software parts. The hardware part is used to simulate input from special input devices connected to a serial port. Below is a picture of the hardware part.

Figure 5: UST Hardware with removed cover

5.4.4 Word Prediction Tool for Developers (SWP)

As should be clear from the description of the language component, the FASTY system bases its predictions on several linguistic sources, out of which word form frequencies are only one. During the development of the language component it has therefore been vital to have a test bed for experimenting with the effect of the different language models. Along with the progress made in compiling the linguistic data and the implementation of the different language models, a development tool has continuously been updated. This (dynamically changing) tool, labelled A Simple Word Predictor (SWP), has been in daily use at the three language development sites (ÖFAI - German, MULT- Dutch and French, UU-Swedish). By means of SWP, baselines for the core functionality for the four languages have been set in terms of keystroke savings for different parameter settings of the program.

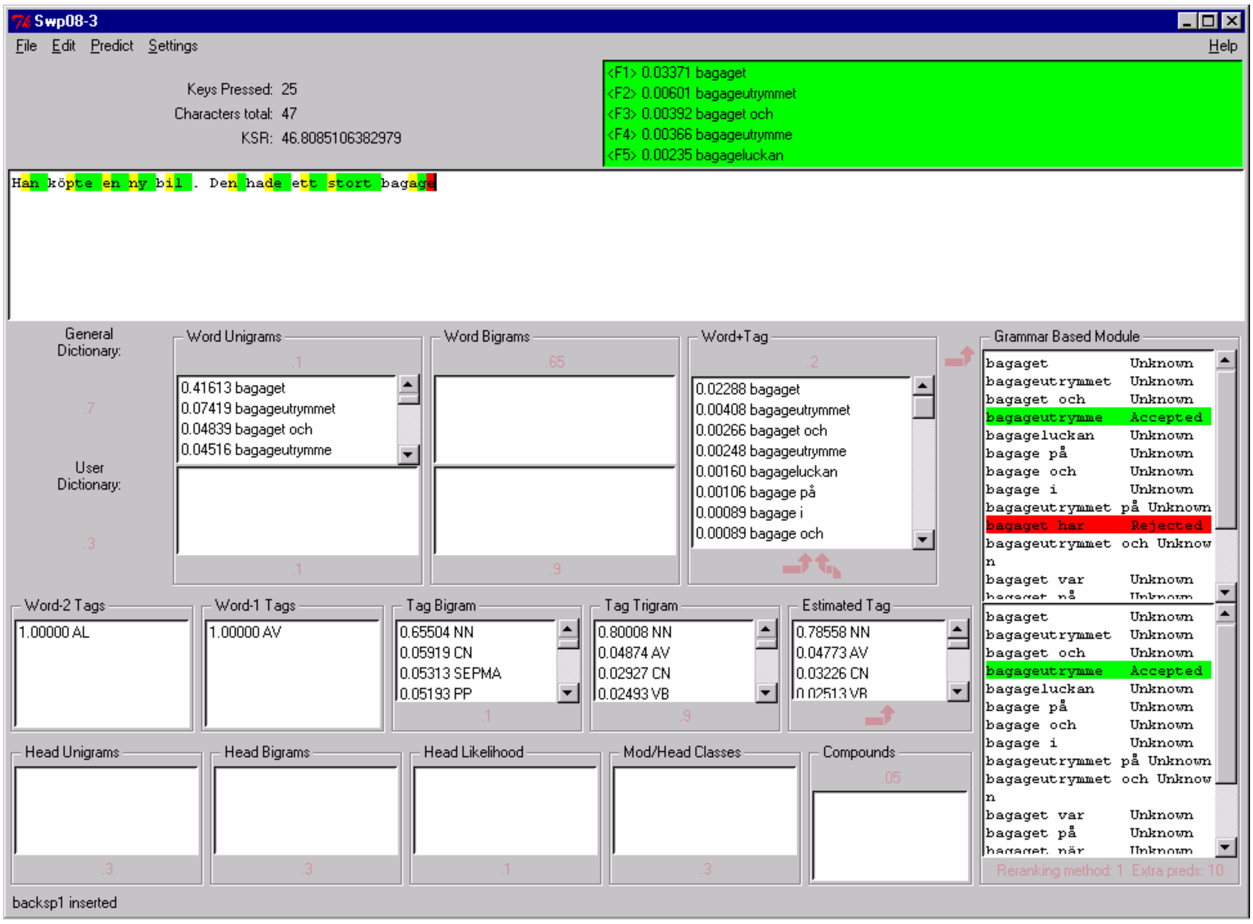

Figure 6: SCREENSHOT of the SWP

In its final state SWP provides for

- Word n-gram based prediction
- Part-of-speech n-gram based prediction
- Morphological backup lexicon
- Compound prediction
- Combination of a (dynamically learning) user dictionary
- Grammar checking as a filter of predictions
- Parameter configuration file

- User dictionary

The effect of the different language models may be viewed in the different windows at the lower part of the graphical interface as letters are entered via the keyboard. Furthermore, prediction parameters can be easily changed and a prediction simulation facility allows for examining the effects of different parameter settings on large batch files.

6 User Verification Phase

6.1 Ethical Aspects

The Report on Ethical Aspects [10] describes the actions and efforts that have been and will be taken to assure an appropriate ethical standard in the FASTY project and will serve as a guideline. The Community Directive on the protection of individuals with regard to the processing of personal data and on the free movement of such data (Directive 95/46/EC) together with the opinions of the European Committee of Ethic (ECE) have been the basis for this report.

The variety of subjects includes the behaviour towards the other partners involved in the project as well as the handling of collected data such as user questionnaires, user ability assessment data, and text samples. Since sensitive and confidential data is involved special care has to be taken in protection of the privacy of the participating persons. Particularly persons suffering from any kind of communication disorder, which is one of the target groups in the FASTY project, deserve in general a high degree of protection.

All data should be transformed and kept in a form, which avoids a trace back to a specific user or permits identification of a specific user for no longer time than is necessary. This was a guideline in the development of the Text Collection Tool that was mentioned above.

6.2 User Test Procedure

When the first prototype (PT1) was ready a first user evaluation was performed. The purpose of the evaluation was to test the functionality and usability of the first prototype of FASTY. Not only the prediction was evaluated, but also the interface of the system into which it is integrated.
20 voluntary test users tested the prototype: five from each of the involved language areas. They all had various kinds of physically or linguistically disabilities. The test users received FASTY for eight weeks, starting in April and used FASTY for their every-day typing and communication tasks. The test followed the procedure stated in the User Test Time Table described in figure 7 below.

	Week								
		INTRODUCTION							
	1	31	1	2	3	4	*	5	6
Phase I (April)		CALIBRATION + Report on Friday							
	2	7	8	9	10	11	R	12	13
		Report on Friday							
	3	14	15	16	17	18	R	19	20
		Draft QUESTIONNAIRE + Report on Friday							
	4	21	22	23	24	25	Q+R	26	27
Phase II (April,May)		CALIBRATION + Report on Friday							
	5	28	29	30	1	2	R	3	4
		Report on Friday							
	6	5 P	6	7	8	9	R	10	11
		Report on Friday							
	7	12	13	14	15	16	R	17	18
Phase III (May,June)		CALIBRATION + Report on Friday							
	8	19 P	20	21	22	23	R	24	25
		HARDWARE Keyboards + Report on Friday							
	9	26	27	28	29	30	R	31	1
		Final QUESTIONNAIRE + Report on Friday							
	10	2 P	3	4	5	6	Q+R	7	8

Figure 7: Timetable

Legend

(*) First delivery of PT1 with logging functions

(P) Install Patch/Update of FASTY

(Q) Questionnaire

(R) Report

During the ‘calibration’ the users were asked to write a defined text in order to discover changes over time. The tests started with version 1.02.05 and were continued with the subsequent releases of the FASTY prototype. The updates were delivered as automated patch-packages, so even partners with slow Internet connections could retrieve these packages without any problem and use them without any special knowledge about computer internals.

The result of this evaluation was partly based on log files and partly on a questionnaire that the users were asked to fill in. The result of the user test helped the consortium in working on the revised specifications of the system.

During the verification phase, a special password protected web site was provided, to give users and the user partners the possibility to retrieve the latest full versions and patches of PT1. Additionally, FAQs about the program and the installer and other user related information were accessible via this web site.

FASTY

Faster Typing for Disabled Persons

This site will help you to get the latest updates and information during the verification phase of the prototype 1 (PT1). Please bear in mind, that the programs, patches and other stuff at this site are given to you for free, but licence agreements are included in these packages and must be observed.

FASTY Prototypes

EXE: One single setup.exe (good for internet distribution); will be likely distributed later than other versions (compiletime longer than 1 hour!)

ISO: A zipped ISO-file for direct burning without authoring. You may use tools like IsoBuster to extract the files and make your own CD Images.

Light: One single setup.exe **without speech output**.

An update from a full version will not work with this light version!

Please note: Don't mix different installer versions for updates. Please decide which installer packet fits your local needs best and then **stick to this packet type**. You will get many internal errors, if you mix different installer packet types.

Updates using the patches are the easiest and safest way.

The comments show the information given by the provider of the files. Please note, that this information is not checked by other partners before the release is given to you. Therefore a bug reported as fixed may not be fixed in your configuration. Please report about your local behaviour (fixed/not fixed).

Version	EXE	ISO	Light	Comments
1.02.04SE	182MB	188MB	N/A	
1.02.05	182MB	189MB	N/A	
1.02.06	182MB	187MB	N/A	new dictionaries and lc.dll
1.02.07	182MB	187MB	N/A	installer can do updates, corrections on Swedish texts, fixes for bug #008, #009, #010, #017, #025, #075, #080, #081, #086
1.02.08	182MB	195MB	22MB	fixed installer in all languages, new lc.dll 1.5, new grm.dll, new german gramar rules, new version management of most *.exe and *.dll, fixed logging, fixed keyboard mouse settings
1.02.09b	182MB	205MB	22MB	new version information in abex.dll, slightly bugfixed help, bugs #93, #95, #96, #118 and #141 are fixed b version fixes problems with different 1.02.09 versions distributed by FTB.
1.02.10	182MB	205MB	22MB	new version information in some mbrola dlls, one bug fix in help file, bugs #80, #115, #127, #132, #144, #146, and O, P, R fixed.
1.02.11	160MB	165MB	26MB	enhanced help file, new mbrola SAPI engine, gb.dll with version information, tutorial included, PT1 has only internal changes -> no bugfixes
next update	no information available			

FASTY Prototype Patches

These small updates should ensure a convenient alternative to the full installations. Once you have installed FASTY and no major version change takes place, a patch for upgrading will be provided. Just start the patch and upgrade your version to the latest available one.

min. Version	upgrade Version	Size	comments
1.02.04se	1.02.08	8017kB	upgrades all versions from 1.02.04se and up to 1.02.08
1.02.06	1.02.08	433kB	upgrades all versions from 1.02.06 and up to 1.02.08
1.02.04se	1.02.09b	8068kB	upgrades all versions from 1.02.04se and up to 1.02.09b
1.02.06	1.02.09b	1290kB	upgrades all versions from 1.02.06 and up to 1.02.09b
1.02.09b	1.02.10	416kB	upgrades from 1.02.09b to 1.02.10
any	1.02.11	N/A	Due to major changes in mbrola, patches are not available to this version. Sorry for any inconvenience due to this.

Tutorials and Help

Tutorial
A short tutorial, which will help you to get familiar with FASTY PT1.

Help
The FASTY help files.

FAQ (Frequently Asked Questions)
Questions ans Answers concerning problems and possible bugs with PT1.

Discussion Board
Here you can discuss about the PT1, give tips to other users and ask questions about problems.
Just follow the link, register (if you haven't done so far) and access the forum. The password for the forum is the same as for this site.

Installer FAQ
Information about problems and workarounds / fixes concerning the installer.

Important information about uninstallation
A short description including important information about uninstallation of PT1.

The information in this document and all following documents is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. The contents in no way represents the views of the European Commission or its services.

Figure 8: Screenshot of the start page of the User Download Site (after the verification phase)

A Discussion Board was also established during the verification phase for the users to discuss with other users about FASTY: http://www.fortec.tuwien.ac.at/cgi-bin/ikonboard/ikonboard.cgi

6.2.1 Result of the log files

In order to obtain precise measurements during the user tests, a logging functionality was added to the prototype. All relevant events, such as keystrokes made by the user, selection events, changes in parameter setting, etc., were written to a log file. The log files that contain sensitive and confidential material were filtered through an anonymity module in order to protect the users identity. For reasons of respect for the users' privacy, logging of events that could be used to reveal the text that has been typed by the user, may be switched off by the user. If turned off, the logger will not record any keystrokes or selections; however, changes of dictionary or parameter settings will still be recorded. The status of logging is indicated in a small window on the screen.

The logged data is collected in a separate subdirectory. Each session creates a new file. The logs are stored as plain text files. Every line in the log files starts with a numeric record identifier, explained below. Every log file contains only one start record as its first line. After the start line all parameter settings are logged as parameter setting records. User specific dictionary and user abbreviation file names are logged as well as all relevant settings of the language component. Information concerning the settings of the UI is however completely missing, so an evaluation of the log files with respect to the UI is not possible with the available data.

After collecting all produced log files, consisting of the information explained above, they were calculated using the keystroke saving rate (KSR) formula. KSR is the percentage number of keystrokes that a user saves by using a word predictor to type, compared to the total number of keystrokes that are required to generate the same text without a predictor.
The total number of produced log files is given in the table below. The number of log files is not uniformly distributed across the participating languages.

Language	**Total Number of log files**
German	220
French	25
Dutch	26
Swedish	141
Total	*412*

Table 1: Total number of logs.

When inspecting the log files, it turned out that the users used a lot of backspace keystrokes to erase typing errors. This may to a certain extent explain the low KSR. Sometimes the actual KSR based on text length and keys pressed even had a negative value due to user errors. In order to compensate for this negative impact on the KSR, attempts have been made to calculate the optimum KSR, i.e. the KSR without such unnecessary keystrokes that typing errors produce. The number of keystrokes due to user errors has been computed in the following way:

- If a backspace is typed after a keystroke event, i.e. the last character typed by the user is erased; the error key count is incremented by 2.
- If a backspace is typed at the beginning of the text, the error key count is incremented by 1.
- Sequences of backspaces after selection events, i.e. correcting or erasing accepted predictions, are not counted as errors.

The optimum KSR is thus based on the length of the text and the number of actual keystrokes reduced by the number of erroneously typed keys. The optimum KSR may not reflect the correct value of the KSR, but is still more reliable than the actual KSR. The optimum KSR is used in the general description below. For a detailed description see [11] and [12].

6.2.1.1 Dutch speaking users

The use time increases with the runtime of the tests, likewise do the words written within a session. So it may be assumed, that the users got better used to the FASTY system. The optimum KSR reached a maximum week average of some 33%. The writing speed increased slightly during the evaluation phase.

6.2.1.2 French speaking users

The use time increased substantially with the runtime of the test, and likewise did the words written within a session. The optimum KSR reached a maximum week average of some 29%, but the writing speed decreased during the evaluation phase. This may be caused due to the big number of special characters in French in combination with the above generally described bugs. A second reason may be that the users got more tired at the end of the much longer sessions and decreased the average in that way.

6.2.1.3 German speaking users

The statistics for German is divided into three parts, depending on which partner that was responsible for the collection.

The use time of the ELI’s users was quite stable within the runtime of the tests, likewise were the words written within a session. The optimum KSR reached a maximum week average of some 46%. The writing speed increased gradually during the evaluation phase, The use time of fortec's user was quite stable within the runtime of the tests. The produced text per session rate differed in the course of the tests. This may be caused by the fact that only one test user participated at this site and therefore more statistical fluctuations occur. Additionally an injury of the test person within the validation phase handicapped the writing abilities of the user and influenced the daily output in a major way. The optimum KSR reached a maximum week average of some 48%. The writing speed increased in the last week of the evaluation phase rapidly. This is caused by the fact that the user mostly wrote the calibration text in the last week.

The use time of FTB's users increased slightly within the runtime of the tests, likewise did the words written within a session. The optimum KSR reached a maximum week average of some 39%. The writing speed increased slightly during the evaluation phase.

6.2.1.4 Swedish speaking users

The use time of the Swedish users increased slightly within the runtime of the tests. The optimum KSR reached a maximum week average of some 23%. The writing speed increased slightly during the evaluation phase. For a more detailed description of the Swedish statistics, see [14].

6.2.2 Result of the questionnaire

After the end of the test period, the test users of FASTY were asked to fill in a questionnaire about the system's functionality and usability. The questions were related to the users' experience with the FASTY system during the writing tasks. Questioned were the users' personal opinion, experience and impression. It aimed at comparing the users' experiences with the proposed expectations and needs. The FASTY Consortium put the questions together after taking into consideration the international ergonomic norm ISO 9241-10, and after consultation with the test users.

A few weeks earlier a draft questionnaire had been presented to the users, in order to guarantee that the questions were understandable and unambiguous and that the questionnaire met the reliability and validity requirements. They thought that the questions were understandable and relevant. Here are a few questions that they wanted to add to the final questionnaire and which were added in consequence.

- Does the system lack some usable function or feature?
- I think that the system satisfies my need as user well.
- I think that the system has improved my text generation.
- More open questions, where the user can write whatever she/he wants about the system.

The final questionnaire consisted of 20 closed questions and 5 open questions. The closed questions were constituted by statements, which the users were to agree or disagree with. Each question had four alternative answers for the user to choose among: *strongly disagree*, *rather disagree*, *rather agree*, and *strongly agree*. The questions were further subdivided into four subgroups depending on the main subject of them. These subgroups were:

- Use of prediction and other functions during writing
- Adaptation of the system
- Documentation, online-help, and tutorial
- General impression

The open questions were mainly about user satisfaction and the users were among others asked to give concrete examples of how to improve the system. The general result of the questionnaire is presented below. For a more detailed description, see [12].

One thing to bear in mind when analysing the questionnaire is that it is not clear if the users have answered the questions with the current prototype or the general system in a complete version in mind.

6.2.2.1 Use of predictions or others functions during writing

The users think that it is easy to use the system and that the functions seem to be well integrated. Nevertheless, it seems that the opinions are divided concerning the use the system in every day life.

6.2.2.1 Adaptation of the system

The opinions are divided about the adaptation of the system but most of the users think the personal adaptation is well supported by the system.

A majority of the users thinks that the different functions and settings are clearly arranged in the menus, the vocabulary easy to understand and the support of a technical unnecessary. On the other hand, they think that the system is unnecessarily complex. For the next investigation, we will have to emphasize on theses three points in order to make it sure that the statements are well understood.

6.2.2.3 Documentation, online-help, tutorial

Because of the fact that these documentations were not completely finished at that time of the test only a few persons have answered the corresponding questions. It is therefore difficult to do any analyses of the answers.

6.2.2.4 General impression

The system seems fun to use even though the users do not feel confident in using it. For the training part, they approve that they didn't have to learn a lot of things before using it and that lots of people would learn to use it quickly.
Even if the opinions about the inconsistency of the system are divided, they think that it is not cumbersome to use FASTY.
The users' opinions divided about the fact that FASTY can improve the quality of their text and their text generation rate.

6.2.2.5 Open questions

- A majority thinks that the system satisfies their user needs, but some users detail, that this is only true, if the system would be bug fixed.
- Nearly all users think that improvements should be done in the current system.
- Most of the users think that no function or feature is missing.
- Many of the users did not answer the question, if some of the implemented features are not useful. The rest said that they find all functions useful.
- The additional comments show no special direction or missed topic of the rest of the questionnaire.

6.2.3 Preliminary tests with KEYBOARD

The hardware keyboards were not tested by the corresponding user side, but only at FTB. The test user had some difficulties to operate the keyboard resulting from two main problems:

- The user meant she is principally able to distinguish between weak and strong pressure, but her "strong" pressure was relatively weak during first tests (threshold: 935).
- There are problems to deactivate the repeat rate of the keyboard settings in Windows 98 (and maybe also in other Windows versions), because deactivating the repeat rate (system settings: accessibility options - keyboard - repeat rate) did not really result in a deactivated repeat rate: after two seconds, the letter typed will be repeated.

After first pre-tests the user was instructed to type stronger in order to reach a higher threshold. This was possible. She reached a threshold of 2340. When she was trying to type her name, she was able to press a key strongly but she was not able to release the key fast enough (due to her spastics). This resulted in two letters: a small letter followed by the same capital letter. Usually (without the pressure-sensitive-keyboard) her repeat rate is reduced and there are no problems or difficulties with speed of the key-release-action.

7 Redesign Phase

During the expert tests and the verification phase 262 bugs were discovered. 152 of them were major bugs, which may cause serious troubles using FASTY. Most bugs were caused by the user interface, but some bugs address the installer and the predicted words.

In the following a short summary of the found bugs will be given, a detailed list of bugs can be found in [11].

- Many minor bugs address not observed general User Interface Design Guidelines like e.g. not or wrong aligned controls, uncommon self designed dialogs (were standard dialogs are usual), missing labels, strange tab orders, …
- Some bug reports deal with the wrong usage of the serial port (used by the pressure sensitive keyboard)
- Some bugs concern errors in the key code handling like:
  - Keys used for the selection are passed to the destination application, too
  - Keys that may be used for selection (but not used in the current configuration) do not work any more
  - Keys used for selection are not available with modifier keys (e.g. SHIFT, ALT, CTRL)
  - Key strokes get totally lost
  - Key strokes pass sometimes previous key strokes
  - Mouse events are interpreted as keystrokes
  - Problems with caps lock or sticky keys
  - Problems with special characters such as ö, ß, é, û, å, ç, ij and key combinations with ALT and CTRL
- Some bugs describe problems when writing in different applications, child windows or controls.
- Some bugs show problems with the graphical output, such as partly empty prediction window, cut predictions, missing predictions, invisible prediction window, etc.
- Some reports say, that users have problems with the unalterable usage of keys or key combinations.
- Some bugs were caused due to the fact that the user dictionary was not saved, or language parameters were not passed to the language component or the speech output.
- Some bugs describe buttons that do not work at all, or do something not intentional.
- Some bugs are based on not checking existing things.
- Some problems arise, because program parts were not developed along the agreed specification or guidelines.
- Some problems are caused by self-designed dialogs that don't work in different environments or don't work at all.
- Many bugs were reported with abbreviation and dictionary handling. Internal tests show, that

the functions of the Language Part seem to work and that the errors occur using these functions or during dictionary handling within the UI.

- Some bugs address obviously missing features.
- Some bugs were caused by not checking available resources or incorrect handling of error results coming from sub-modules.
- Some bugs show incompatibilities of FASTY with standard products.
- Some bugs show a smart punctuation function that causes problems and even crashes the computer.
- Many users report a slow system in general; very slow dictionary handling and loading, and slow prediction generation.

These bugs were to be taken care of during the redesign phase in order to provide a new, better prototype for the validation phase. The important list of the technical problems was however the subject of a much longer revision than foreseen, so the redesign phase had to be prolonged until December 2003.

During the verification phase, it was also noticed that the general design was inconsistent. For the PT2, a redesign according to general design guidelines was done.

In the PT2, the following was planned: all keys are caught, passed through the FASTY system and generated newly. Some problems concerning delays which has to be ensured for a proper key generation has been noticed too. In the PT2, a new functionality for key generation avoids the delays and the maybe problematic usage of special key codes for the differentiation of own generated keystrokes and external keystrokes. Additionally the shift related problems are solved with that. Most of the display problems are caused due to the special way of creating the prediction list (using rich edit controls). This bunch of bugs is removed in the PT2 by a new way of creating the prediction list.

During the verification, it was noticed that there was an insufficient initialisation or missing drivers, which led to unpredictable behaviour**.** During the implementation of PT2, an error handling avoided problems with not correct working FASTY systems (e.g. not selected main drivers).

The Adjustment Tool has to be open for all kinds of drivers. Therefore it was impossible to integrate all possible settings into the Adjustment Tool. Thus, the Adjustment Tool selects the used drivers and the configuration of driver-specific options is only induced and the driver itself displays a setting dialog. The storage of the settings had to be done by the driver as well. This was not the case for PT1 but has been done in the PT2.

During the verification phase, keys that are reserved as hot keys (numpad) or double functions (F12) could not be used for other purposes. In the PT2, a new way of catching the keys is used to avoid dead keys. Additionally the keys used for FASTY internally (e.g. prediction selection) are selectable. In this way every user can select the keys he or she can give up for normal writing. Additionally, keyboards with extra keys (e.g. Internet keyboards) can be used too (but most likely the drivers of the keyboard must not be installed). The prediction window was taking the input focus in PT1. In the PT2, the prediction window rejects the focus.

The contents of the zip archive containing PT2 are listed below.

- **Dict** holds the customised, user specific, dictionaries. The customised dictionaries are delivered as separate archives.
- **Driver** holds the DLLs of various device drivers (e.g. pressure-sensitive keyboard, speech output etc.)
- **Lang** holds the DLLs for customising the language of the GUI
- **LC** contains the language resources for the different languages (for the language codes see [D7.1]
- **User** contains the user-specific setup and profiles
- **Log** holds the log files generated during the user tests in the verification phase.

8 User Validation Phase

8.2 User Test Procedure

The user validation phase began in January 2004 and ended the last week in February 2004. The introduction phase is not included in this six weeks period. Unfortunately the Prototype 2 (PT2) was not in the agreed state at the beginning of the tests. Therefore the consortium agreed on the same procedure as with PT1, i.e. to release an update as soon as there was a new version. The user test began with PT2 version 2.00.00 and ended with the version 2.00.03, so three updates were delivered to the users during the validation phase. The updates were delivered as rather small, automated patch packages.

The test included the usage of FASTY for communication jobs in the daily life of the users. Each Friday, a report was sent to the technical team concerning possible bugs, the users' behaviour when using the system, software that does not work with FASTY, etc.

The evaluation is partly based on a questionnaire and partly on log files. The questionnaires and logged data could be submitted by the participants through the user partners or by using a poll tool ‘Polly’ that has been developed for WP10 and that allows sending the data easily via

Internet to a server that has been installed by and at fortec. There was no calibration during this test phase.

8.2.1 Result of the log files

The logged data was collected and calculated in the same way as in the user verification phase. Basically the log files' format follows the specification, that has been described in [13]. All relevant events, such as keystrokes made by the user, selection events, changes in parameter setting, etc., were logged.

The total number of log files is given in the table below, fielded by language.

	Number of logs	Average chars per log	Average number of words per log
Dutch	7	2270	287
French	39	150	19
German	292	826	90
Swedish	61	267	40
Total	399	699	79

Table 2: Total number of logs.

The optimum KSR is used in the general description below. For a detailed description see [15] and [16].

	Average opt. KSR
Dutch	40,32
French	34,5
German	33,34
Swedish	19,21
Total	31,84

Table 3: Optimum KSR

8.2.1.1 Dutch speaking users

Only 7 Dutch logs were collected during 3 weeks. The average optimum KSR was 40,32%. The optimum KSR reached a maximum week average of 52,22%. The use time first decreased and then it increased within the runtime of the test. The average number of words per session increased during the test period. There was a huge step between week 6 and week 7. The writing speed did gradually increase during the evaluation phase, which may indicate that the users improved their text generation by using FASTY.

8.2.1.2 French speaking users

39 French log files were collected during 5 test weeks. The average optimum KSR was 38.59%. The optimum KSR reached a maximum week average of 71,8%. The use time decreased substantially with the runtime of the tests. The average number of words per session was rather stable. The very long session time during the first two weeks may be explained by the fact that introduction and initial testing took place during this time. The writing speed did increase during the evaluation phase. Week 1 the users wrote 0,006 words/minute and the last week they wrote nearly 3 words/minute.

8.2.1.3 German speaking users

292 German log files were collected during 8 weeks. The average optimum KSR was 32.3%. The optimum KSR reached a maximum week average of some 56,84%. The use time slightly increased within the runtime of the tests. The number words written within a session was rather stable. The writing speed was also quite stable during the evaluation phase, except from the first week, when the average typed words per minute was nearly 6. This may be explained by the fact that introduction took place the first week.

8.2.1.4 Swedish speaking users

61 Swedish log files were collected during a period of seven weeks. The average optimum KSR was 19.21%. The optimum KSR reached a maximum week average of 45%. The use time decreased slightly within the runtime of the tests. Due to introduction phase the first week of use showed a rather low number of words per minute. The number of words per session was rather stable. The huge step towards higher writing speed in week 5 can be explained by a much shorter use time.

8.2.2 Result of the questionnaire

After the end of the test period, the test users of FASTY were asked to fill in a questionnaire about the system's functionality and usability. The questions were related to the users' experience with the FASTY system during the writing tasks. Questioned were the users' personal opinion, experience and impression. Each question had six alternative answers for the user to choose among, see figure 9 below. The questionnaire had been extended with a few more questions since the last test period, and there were also some preliminary questions concerning demographic data, computer skills, and purpose of use. The users’ opinions differed rather much. It is therefore be difficult to draw any clear conclusion out of the result, but generally the result is alarmingly poor.

Figure 9: Screenshot of the Poll Tool

8.2.2.1 Demographic Data

The questions asked in the personal part of the questionnaire concerned demographic data about the users, such as gender, age, level of education, etc.

The distribution of gender in the test was rather uniform. 18 women and 15 men took part in the test. There is unfortunately no information concerning the distribution across the languages available.

Most of the users were rather young. Only two of the users were over 50 years. Most of the users had secondary school, CSE-level as highest level of education, which can be explained by the fact that most of the users were younger than 20 years. Only one of the users had university degree.

Most of the users had some kind of motor disability. Some users had linguistic disabilities, such as dyslexia. The most common disabilities were tetra-spasticity and spasticity. For a detailed description, please see [15] or [16].

8.2.2.2 Computer Skills and The Purpose of Use

Most of the users claimed to be advanced computer users. Seven users claimed to be beginners and only three claimed to be experts. This means that the experience of computers among the users were rather uniform.

Most of the users use computers between 0-15 hours per day. Most of the users that claim to use computers more than 15 hours per day use the computer at work or in school. Generally, the users use their computers for a rather long time each day.

Most of the users did not have any special devices. They mainly use standard keyboard and standard mouse.

The users use their computer for many kinds of purposes, but the computers are mostly used for writing and Internet tasks. Only 5 users use computers for programming. The users also use many different application programs. MS Word is the most common program.

8.2.2.3 Use of predictions or others functions during writing

About half the users did think that the system is easy to use. They did not think that the functions seem to be well integrated and most of users did not want to use FASTY for their everyday writing tasks.

8.2.2.4 Adaptation of the system

The opinions were divided about the adaptation of the system but most of the users did not think that the personal adaptation is well supported by the system.
A majority of the users did not think that the different functions and settings are clearly arranged in the menus or that the vocabulary easy to understand. Most of the users claimed to need technical in order to use the system. On the other hand, they did not think that the system is unnecessarily complex.

8.2.2.5 Documentation, online-help, tutorial

The opinions concerning documentation differed very much. They did think that the explanations in the menu and the manual were easy to understand. They did however not think that the manual describes features/problems that they needed/faced.

8.2.2.6 General impression

Most of the users did think that it was fun to use FASTY. The opinions concerning confidence in using the system and inconsistency were very divided. The users did not have to learn many new things to be able to use the system and they did think that most people would learn to use FASTY very quickly. There was no clear decision if the system was cumbersome or not. A majority of the users did however not think that the system increased the effort in writing nor improved the quality of their texts. They did not think that FASTY improved the text generation rate. Most of the users did also think that FASTY needs improvement in order to fit their requirements and needs.

8.2.2.7 Further remarks

The additional comments show no special direction or missed topic of the rest of the questionnaire. So there is nothing special here to mention. For a detailed description, see [15].

8.3 Tests of the Pressure Sensitive Keyboard

The Pressure Sensitive Keyboard (PSK) is a standard PC keyboard with a pressure sensitive element under each key inside the keyboard case. It allows evaluation of pressure on the keys and uses the pressure information for additional functions. At the present time the PSK exists only for a standard PC keyboard.

Two programs were available for the PSK test: FASTY and PSKTest. PSKTest is a program to control the writing of lower-case/upper-case letters by the amount of applied pressure. It has been developed to test the PSK before the necessary implementations for FASTY had been done. The keyboard has not been tested by users, but only by the project partners themselves. The shift function works rather well, but may be unintentionally activated. Because of sluggish keys, typing also can get very strenuous and needs a lot of efforts. For at detailed description of the test result, please see [15].

9 Technological Implementation

9.1 Technological implementation plan

The Technological Implementation Plan summarises the results that have and will be achieved in the project. A midterm draft plan for FASTY is available.

By the middle of the third year of the three-year plan, the following was achieved:

- The prototype of a User Ability Assessment Tool (UAAT) has been developed for testing purposes.
- The prospective users have been contacted and the user panel has been founded.
- A first rapid prototype has been developed, in order to give the potential users a view of the ideas.
- A FASTY web site has been created (available in English, German, French, Dutch and Swedish) and can be accessed under http://www.fortec.tuwien.ac.at/ fasty .
- A FASTY folder has been created (available in English, German, French, Dutch and Swedish).
- The first edited annual report was published and is additionally available online as HTML-document.
- A prototype of a pressure sensitive keyboard (FASTY Box) and pressure sensitive single sensors are available for testing and gaining first data.
- The internal structure of the FASTY system was developed together with the interfaces between the modules and their basic functionality.
- The first version of a user simulation tool is available.
- A Text Collection Tool has been developed which helped to collect and anonymise user text.
- A huge amount of user text (from disabled users) and general text has been collected for linguistic analysis.
- A detailed definition of the Language Component was developed.
- A prototype of the language components has been developed and language resources for all supported languages have been created.
- A SAPI Interface for speech output has been developed.
- A first prototype (PT1) was developed, which contains the language component, the user interface and an Adjustment Tool.
- A test-phase with 20 users using PT1 has been finished.

- A prototype two (PT2) based on the results from the user-tests has been developed.
- A second test-phase with a huge number of users has been started and is still running, results influences the further development in a direct way.

10 Dissemination and PR Activities

During the third project year, the FASTY partners have continued to carry out several activities for the dissemination of the expected results of the project and the benefits of the future use of its results.

10.1 Project presentation on the web

There is a presentation of the FASTY project on the web at the following site:
http://www.fortec.tuwien.ac.at/fasty
The presentation is available in English and further, for all FASTY languages. During the third project year an upgrade has been undertaken. Fig. 7 shows a screenshot of the current English version.

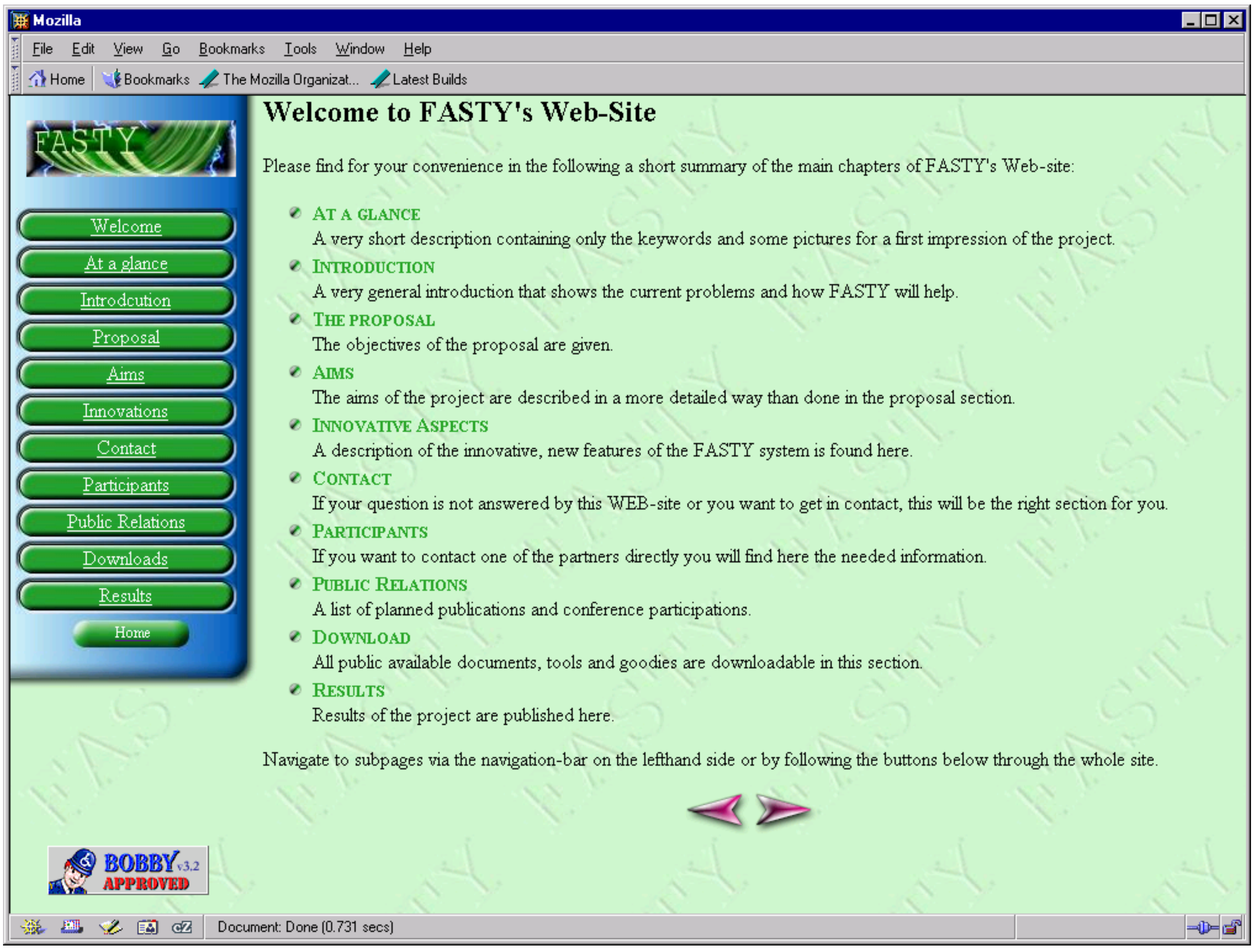

Fig. 7 Screenshot of the current English start-page of the FASTY website

10.2 Project Folder

A colourful Project Folder, with an overview of the objectives and features of the project, has been issued in English as well as in the FASTY languages and distributed at meetings, conferences and workshops. It is available for download at the FASTY website (http://www.fortec.tuwien.ac.at/fasty). During the third project year it has been updated with the latest information to better mirror the current state of the project.

10.3 Project presentations

During the third project year the FASTY partners have participated in several conferences and other events, as detailed below:

- CSUN, Los Angeles, USA, 17th – 22nd March 2003
- 'Der Turmbau zu Babel' Exhibition, Graz Austria 5th April – 5th October 2003
- AAATE, Dublin, Irland, 30th August – 3rd September 2003
- 7. Österreichtag, Vienna, Austria 28th October 2003

The FASTY partners have participated in the following workshops, seminars or courses:

- EACL Workshop Budapest, Hungary, 12th April 2003
- Rehab Fair, Karlsruhe, Germany 14th – 16th May 2003
- Workshop 'Neue Medien' Vienna, Austria 28th October 2003

10.3 Project publications

During the third year the following publications have been produced:

- Matiasek J., Baroni M.: Exploiting Long Distance Collocational Relations in Predictive Typing, in Proceedings of the EACL Workshop on Language Modeling for Text Entry Methods, Budapest, Hungary, pp.1-8, 2003. [available online at http://www.oefai.at/~john/papers/eaclws03.pdf]
- Zagler W.L., Beck C., Seisenbacher G.: FASTY - Faster and easier text generation for disabled people; Presentation: AAATE '03, Dublin; 08-30-2003 - 09-03-2003; in: "Assistive Technology - Shaping the Future", G. Craddock, L. McCormack, R. Reilly, H. Knops (ed.); IOS Press, Volume 11 (2003), 1 58603 373 5; 964 - 968.
- W.L. Zagler, C. Beck: FASTY - Faster Typing for Disabled Persons; Presentation: EMBEC'02 2nd European Medical & Biological Engineering Conference, Vienna; 12-04-2002 - 12-08-2002; in: "Proceedings of the EMBEC'02", (2002), ISBN 3-901351-62-0; 1678 - 1679.
- W.L. Zagler, C.Beck, G. Seisenbacher, et. al.: First user test results with the predictive typing system FASTY; Presentation: ICCHP'04
- Gustavii E., Pettersson E.: A Swedish Grammar for Word Prediction, Master's thesis, 2003. [available online at http://stp.ling.uu.se/~matsd/thesis/arch/2003_gustavii_pettersson.pdf]

- Wester M.: User evaluation of a Word Prediction System, Master's thesis, 2003. [available online at http://stp.ling.uu.se/~matsd/thesis/arch/2003_wester.pdf]

References

[1] Carlberger J. (1998): Design and Implementation of a Probabilistic Word Prediction Program, Royal Institute of Technology (KTH).

[2] Palazuelos-Cagigas S. (2001): Contribution to word prediction in Spanish and its integration in technical aids for people with physical disabilities. PhD Thesis, Universidad Politecnica de Madrid.

[3] D5.4 Overall Specifications and Specifications of Interfaces – FASTY 2002

[4] D7.1 Definition of Language Components – FASTY 2001

[5] Copestake A. (1996): Applying Natural Language Processing Techniques to Speech Prostheses, in Working Notes of the 1996 AAAI Fall Symposium on Developing Assistive Technology for People with Disabilities.pp.225-231.

[6] Rosenfeld R. (1996): A Maximum Entropy Approach to Adaptive Statistical Language Modeling, Computer Speech and Language, 10, pp. 187-228.

[7] Rosenfeld R. (1994): Adaptive Statistical Language Modeling: A Maximum Entropy Approach, Carnegie Mellon University.

[8] Weijnitz P. (1999) Uppsala Chart Parser Light. System Documentation. In: Working Papers in Computational Linguistics & Language Engineering 12. Department of Linguistics, Uppsala University. 24 p.

[9] D4.2 User Abilities, Preferences and Needs – FASTY 2001

[10] D4.3 Report on Ethical Aspects – FASTY 2001

[11] D9.1 Verification Test-Data – FASTY 2003

[12] D4.6 Review Report and Documentation after verification – FASTY 2003

[13] D8.2 - FASTY Deliverable D8.2, Prototype System Report – FASTY 2003

[14] Wester M. (2003): User Evaluation of a Word Prediction System, Department of Linguistics, Uppsala University

[15] D10.2 Validation Test Data – FASTY 2004

[16] D4.8 Review Report and Documentation after validation – FASTY 2004

Appendix

Project Coordinator:

fortec

Research Group for Rehabilitation Technology

Favoritenstrasse 11/029

A-1040 Wien

Contact: Dr. Wolfgang Zagler, G. Seisenbacher, C. Beck

Email: fasty-fortec@fortec.tuwien.ac.at

Tel. +43-1-58801-42911, 42914

www.fortec.tuwien.ac.at

Project Partners:

Austria

Oesterreichisches Forschungsinstitut fuer Artifical Intelligence (ÖFAI)

Contact: Dr. Ernst Buchberger, Prof. Dr. Harald Trost

Email: fasty-ofai@fortec.tuwien.ac.at

Seraphisches Liebeswerk für Tirol und Salzburg,Elisabethinum Axams (ELI)

Contact: Bernhard Frischmann, Stefan Mina

Email: fasty-eli@fortec.tuwien.ac.at

Germany

Forschungsinstitut Technologie-Behindertenhilfe der Evangelischen Stiftung Volmarstein (FTB)

Contact: Dr. Helmut Heck

Email: fasty-ftb@fortec.tuwien.ac.at

IGEL Elektronische Kommunikationshilfen GmbH (IGEL)

Contact: Dipl.-Ing. Holger Neumann

Email: fasty-igel@fortec.tuwien.ac.at

Ingenieurbüro für Kunst und Technik II (IKuT)

Contact: Ing. Jörg-Michael Lindemann

Email: fasty-ikut@fortec.tuwien.ac.at

Belgium

Multitel ABSL (MULT)

Contact: Dr. Ir. Stéphane Deketelaere

Email: fasty-mult@fortec.tuwien.ac.at

Facultés universitaires Notre-Dame de la Paix (FUNDP)

Contact : Geneviève Baxier, Jean Pierre Peters, Bruno Plumat

Email : fasty-fundp@fortec.tuwien.ac.at

Sweden

Uppsala University Department of Linguistics and Philology (UU)

Contact: Prof. Anna Sågvall Hein

Email: fasty-uu@fortec.tuwien.ac.at