AGILE

Automatic Generation of Instructions in Languages of Eastern Europe

---

Title ***Design specification for the interface to the intermediate demonstrator***

Authors Stancho Stankov
Richard Power
Tony Hartley

Deliverable *INTF1*
Status *Final*
Availability *Public*
Date *July 1999*

INCO COPERNICUS PL961104

Abstract:

This document contains the partial results of task 1.1 (Interface to intermediate prototype). Specifically, it constitutes the deliverable INTF1 (Design specification for the interface to the intermediate prototype). The document describes the *user interface* for AGILE. This interface is designed for use by technical authors, and it has two main functions. First of all, it provides a knowledge editing tool through which the author can model the content of the generated texts. Secondly, it provides a means of viewing the output texts that the system generates from the current model. Subsidiary functions allow the user to save and load both models and texts, and to vary some presentational parameters such as font size.

More information on AGILE is available on the project web page and from the project coordinators:

URL:	http://www.itri.brighton.ac.uk/projects/agile
email:	agile-coord@itri.bton.ac.uk
telephone:	+44-1273-642900
fax:	+44-1273-642908

Contents

List of Figures

1 Introduction

The AGILE system will assist technical authors in the task of producing documentation for CAD/CAM packages in three Eastern European languages — Bulgarian, Czech and Russian. The author must first define the desired content of the text to be generated. This is achieved by means of a graphical knowledge editing tool; from this formal model of the meaning, the system automatically generates versions in any of the three languages (along with a version in English for demonstration purposes).

We describe in this report the *user interface* for AGILE. This interface is intended for use by technical authors, and it has two main functions. First of all, it has to provide a knowledge editing tool through which the author can model the content of the generated texts. Secondly, it must provide a means of viewing the output texts that the system generates from the current model. Subsidiary functions allow the user to save and load both models and texts, and to vary some presentational parameters such as font size.

The task of knowledge editing depends in an obvious way on the nature of the underlying knowledge representation. As explained in the AGILE deliverable MODL1 (Power, 1998), we follow LOOM and other languages of the KL-ONE family by making a basic distinction between *terminological* and *assertional* knowledge. The terminology, or 'T-box', defines the concepts from which a set of specific assertions (the 'A-box') can be configured. The task of defining the T-box is the responsibility of the *developers* of AGILE, not the users. It is the developers who specify that a procedure comprises a goal and a method, or that a button is a type of interface object and a possible candidate for a clicking operation. From the user's point of view, the T-box is fixed; the purpose of the knowledge editing tool is to exploit the conceptual resources of the T-box in order to build an A-box. In AGILE, an A-box is a set of assertions modelling a procedure for performing some task in a CAD/CAM application.

Knowledge editing in AGILE is a process of progressively extending a model until it is potentially complete. The knowledge editing tool must ensure that the model built by the user conforms to the conceptual definitions in the T-box, and hence that it lies within the scope of the language generators. In order to enforce this compliance, the user is required to begin from a procedure entity for which the goal and method properties are undefined. By clicking on labels signalling these undefined properties, the user obtains a list of suitable values: for instance, a goal can be an action of type *create*, *draw*, *open*, etc. Making a choice from this list introduces a new entity of the specified type. Through the concept definitions in the T-box, further properties will be associated with the new entity — for instance, a *draw* action will have an 'actee' (the entity that is drawn) which must be some kind of *line*.

A special requirement on the AGILE user interface is that it should be *localised* for the three Eastern European languages (as well as for English). In practice, this means that a user speaking any *one* of the supported languages should be able to produce texts in *all* of them. Localisation extends to all dialogue boxes, menu titles and menu options as well as to the labels for concepts and relations that are presented in the knowledge editing tool.

2 Interface Specification

In order to serve the two main tasks (building an A-box model of the text that is to be generated, and showing the generated text in the relevant language) the system has to support two main types of window: the **Model Display Window** and the **Text Display**

Window. The system must be able to display simultaneously a single Model Display Window and as many Text Display windows as there are targe languages for the generated texts. The system user has to be able to switch from any one window to any other. Before beginning a working session, the user must be able to select the interface language for all screen objects.

2.1 General design approach

The AGILE system will run on a PC in the Harlequin LispWorks 4.0.1 environment. The interface is to be implemented in CLIM and will have the general look and feel of a Windows application.

Thus all windows will have a title bar and a menu bar with pull-down menus arranged, where applicable, in the standard order. They will also have boxes allowing them to be minimised, resized and closed. Dialogue boxes will similarly conform to users' expectations of a Windows interface.

2.2 Choosing the working language

On starting up the system, the user is asked to select one language from Bulgarian, Czech English and Russian as the language for all the linguistic content of the interface — window titles, menus, dialogue boxes and A-box model labels. This language is set for the duration of the working session.

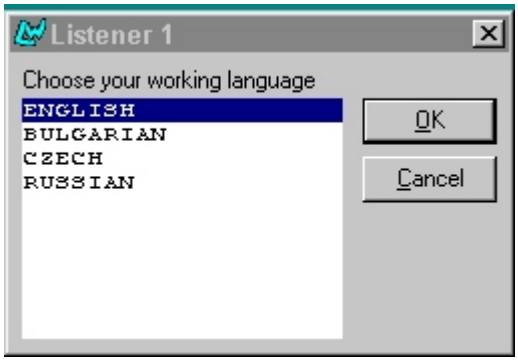

Figure 1: Choose Working Language dialogue box

2.3 Description of the Model Display Window.

This window (Figure 2) is opened automatically when the system is started. It serves the task of building and editing the A-box model. The system does not allow simultaneous work on more than one A-box, i.e. there is only one window of this type and it is for the current A-box.

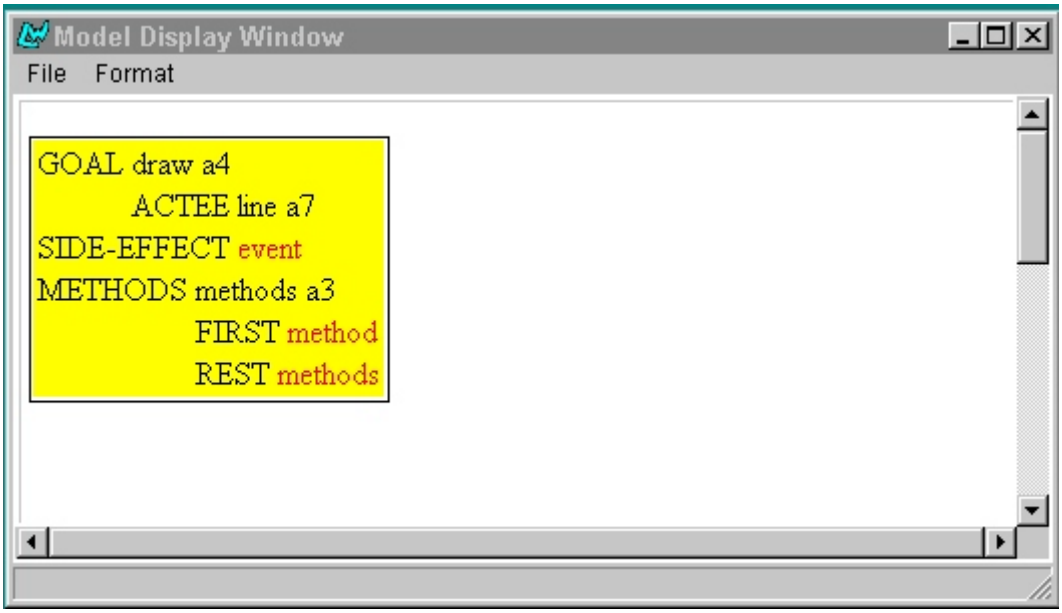

Figure 2: General view of the Model Display Window

2.3.1 Description of the menus

The Menu bar contains the following pull-down menus: **File** and **Format.**

2.3.1.1 Description of the File menu

The File menu contains the following items: **New…**, **Open…**, **Save**, **Save As…**, **Generate…** and **Exit**.

- The **New…** option starts work on a new A-box. The appearance of the workspace when **New…** is selected is described in Section 2.3.2. If **New…** is selected while the window is displaying an A-box, a dialogue box appears (Figure 3). By answering 'Yes' the user can save the old A-box (optional). After that, the system resets the workspace of the Model Display Window and clears the Text Display window(s) so that the work on a new A-box can start.

Figure 3: **New…** option dialogue box

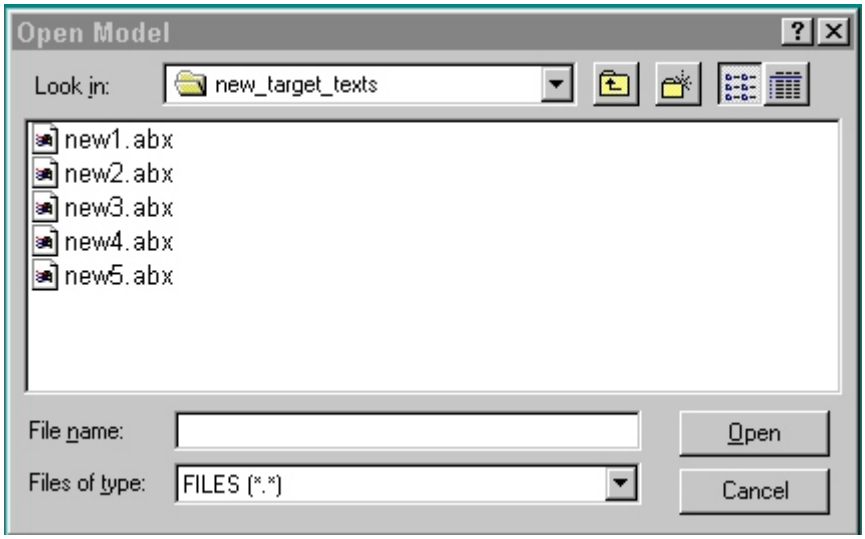

Figure 4: **Open…** option dialogue box

- The **Open…** option opens a dialogue box for specifying an A-box file previously created and saved with the **Save** or **Save As…** options. It offers the user the possibility of browsing through devices and directories (Figure 4). Any A-box currently displayed is first saved or discarded (Figure 3) before the required A-box is loaded.

- The **Save** option saves in a file the A-box the user is working on. As with **Open…**, the user can browse through the directory structure. The dialogue for naming the file occurs only the first time. Then, every time the user saves the file it is saved under the name it has already been given. Saving a file does not cause the contents of the Model Display Window or Text Display window(s) to change and it is possible to resume working from the state in which the **Save** option was executed. Files containing A-box definitions have the extension **.abx**.
- The **Save As…** option behaves in the same way as **Save** except that it asks every time for a file name; once the **Save As…** command is executed, the new name becomes the current file name. In this way, the user can save a single A-box under different names or save variants of the current A-box in separate .abx files.
- The **Generate…** option starts the process of generating an output text from the current A-box. It causes the Choose Generation Options dialogue box to appear (Section 2.4). The result is displayed in the Text Display Window window(s) for the chosen target language(s).
- The **Exit** option leaves the AGILE system, closing all windows. Prior to that, a dialogue box (Figure 3) offers to save the current A-box to a file (see **Save** option). The Model Display and Text Display windows now close and the user return to the OS level — Win95/98/NT.

At this stage, there is no planned facility for printing the graphical representation of the model. Saved .abx files can be printed from any text processor.

2.3.1.2 Description of the Format menu

The **Format** Menu contains the following two submenus: **Font** and **Size.**

- The **Font** option offers the following options: Serif and Courier. After a font is chosen, all text in the workspace of the Model Display Window (i.e. labels on the graphical representation) changes to the chosen font. The font in the dialogue boxes does not change — it coincides with the system font of the OS — Windows 95/98/NT.
- The **Size** option offers the following options: 6, 8, 10, 12, 14, 16 and 18. After choosing a size the system reacts in the same way as in the **Font** option — all text in the workspace of the Model Display Window changs to the chosen size. The size in the dialogue boxes does not change, but coincides with the system size of the OS — Windows 95/98/NT.

2.3.2 Description of the workspace

The workspace of the Model Display Window contains a graphical representation of the current A-box. After choosing the **New…** option, the user first sees a red label called 'start' (or the equivalent label in the current working language). Red labels represent locations in the model where a new entity may be added. To add a new entity, the user opens a pop-up menu on the label, using the left-hand mouse button, and chooses a concept from a list of options. These options are computed from the T-box, and presented in the current working language. In the case of the start label, the choice may be constrained to the single concept *procedure*, since every A-box in AGILE consists of some kind of procedure entity. However, as the model is progressively extended, there will typically be many options at each choice point.

For the most part, the current state of the model is presented through a feature-structure notation: labels in lower case represent entity types, and labels in upper case represent properties. Thus the text **GOAL draw** would mean that an entity of type *draw* was the value of the *goal* property on a procedure. However, since objects of type *procedure* and *method* play a fundamental role in modelling instructions, they are presented through a special graphical device: yellow/orange rectangles present procedures, and blue/violet rectangles present methods. This notation is convenient because in complex instructions, procedures and methods are nested inside one another. Such nested structures can be visualised more easily if they are presented through nested rectangles.

2.3.2.1 Workspace editing options

As well as adding entities at the red labels, the user can open a pop-up menu on any box representing an entity that has already been defined. For instance, if a feature structure corresponding to the sentence `draw a multiline' has been built as the goal of a procedure, the user can open a menu on this feature structure and choose from two options, **Cut** and **Copy**.

- **Cut** removes the selected fragment of the A-box to a buffer, where it remains available for pasting into a new context. The box disappears and is replaced by a red label, which can be used to define a new goal.
- **Copy** copies the selected fragment to a buffer, so that it can be pasted into other contexts. It is likely, for instance, that the multiline in the goal will be mentioned in some of the steps.

After a **Cut** or **Copy**, **Paste** is presented as the first item on the pop-up menu that is invoked by clicking on a label in the model.

2.4 Setting the text generation parameters

When the user chooses the **File/Generate** option in the Model Display Window (Section 2.3.1.1), the Choose Generation Options dialogue box appears (Figure 5). The user is asked to specify the language(s) in which the output is to be generated, and to set parameters which will affect the wording of the generated texts[1]. These settings apply to all output languages.

Figure 5: Choose Generation Options dialogue box

- The **Choose languages** check boxes allow the user to specify one or more text output languages (Bulgarian, Czech, English and Russian).
- The **Choose style** check boxes allow the user to specify whether the output text is to be generated in style A **Personal** only, in style B **Impersonal** only, or in both styles. When texts are generated in both styles, the two versions are displayed in separate panes of the Text Display Window (Section 2.5.2).
- The **Choose side-effects option** radio buttons require the user to specify either that any side-effects should be explicitly mentioned in the text or that they should be left implicit.

When the **File/Generate** option is next selected, the dialogue box displays the current settings. The user is free to modify them. If a new language is specified, an additional Text Display Window appears.

2.5 Description of the Text Display Window

This window (Figure 6) presents texts generated from the A-box description in the Model Display Window. For every language chosen by the user there is a separate Text Display Window. Each Text Display Window consists of a unique Title bar, a Menu bar and a workspace.

[1] The styles and variations selected for the Intermediate Prototype are defined in AGILE deliverable LSPEC2-Bu/Cz/Ru.

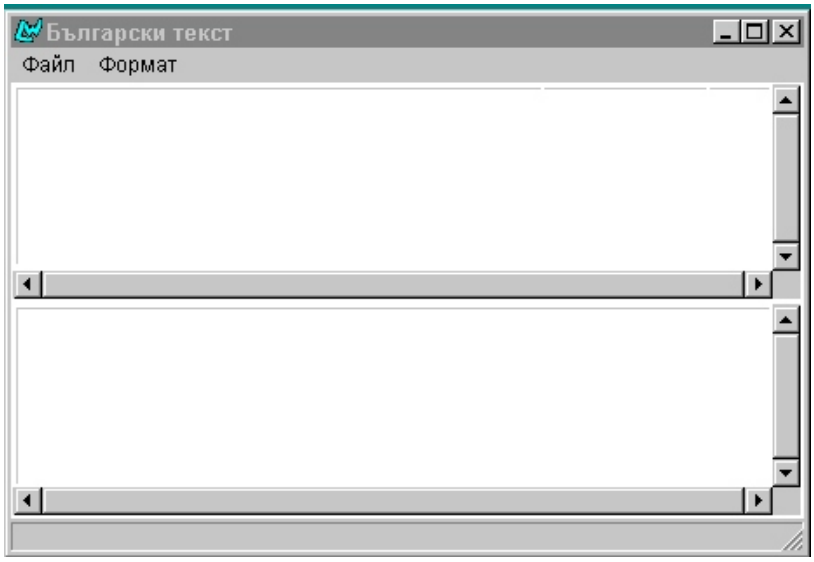

Figure 6: General view of a Text Display Window

2.5.1 Description of the menus

The Menu bar contains the following pull- down menus: **File** and **Format**.

2.5.1.1 Description of the File menu

The **File** menu contains the following options: **Save** and **Print**.

- The **Save** option saves the generated text. The file name is asked only the first time. Then, whenever the user saves the file, it is saved under its existing name. Saved files have the extension **.txt**.
- The **Print** option prints the contents of the active Text Display Window to the active printer of the OS — Windows 95/98/NT.

2.5.1.2 Description of the Format menu

The **Format** menu contains the options **Font** and **Size**, and is the same as for the Model Display Window (Section 2.3.1.2). Changing the font or size affects all the Text Display windows currently open and will apply to any Text Display windows opened subsequently.

2.5.2 Description of the workspace

The workspace of an individual Text Display Window contains the generated text of the CAD/CAM instruction in one of the chosen target languages.

When the generation of texts in two styles has been specified, the workspace is divided in two equal horizontal panes, for the simultaneous display of styles A and B respectively. Each pane is identified by its own title bar and has both vertical and horizontal scroll bars.

3 Localisation of the interface

The interface program is implemented in Harlequin LispWorks 4.0.1 and CLIM. Thus, it will work under different OSs (UNIX, WINDOWS 95/98/NT) and will be compatible with the other components of the AGILE system, which also use the Harlequin LispWorks environment. All the suggestions of the GLOSSASOFT project (Methods and Guidelines

for Interlinguality in Software Construction) will be followed. The interface language can be set at the beginning of a working session to either Bulgarian, Czech, English or Russian. Measures will be taken to ensure that new interface languages can be added with ease. In keeping with the GLOSSASOFT guidelines, the resources subject to localisation will be separated from the program code. The appropriate resource texts in the required language can thus be selected. We are considering an implementation that reduces the need for translation during localisation by deriving the labels and menu items in the Model Display Window automatically from the linguistic resources of the AGILE system.

4 Conclusion

A user interface for the AGILE Intermediate Prototype has been proposed. It has two main functionalities: a knowledge editing tool for defining the content of the texts to be generated (the Model Display Window); and a facility for presenting the generated texts (the Text Display Window). In addition to their main functions, both types of window have options for saving the contents of the workspace (A-box models and generated texts) and for controlling presentational features, such as font and size. The interface is fully localised for Bulgarian, Czech, English and Russian.

References

[MODL1] Richard Power (1998) Preliminary model of the CAD/CAM domain. Deliverable MODL1 of AGILE project PL961104.

The GLOSSASOFT Consortium: The Open University, UK; The National Center for Scientific Research "Demokritos", Greece; Hewlett Packard Hellas, Greece; Claris, Ireland; VTT, Finland; Bull, France (1995) Methods and Guidelines for Interlinguality in Software Construction. Deliverable 11.1 of GLOSSASOFT project LRE61003.