AGILE

Automatic Generation of Instructions in Languages of Eastern Europe

---

Title ***Integrated intermediate prototype for all three languages (Bulgarian, Czech, Russian)***

Author Stancho Stankov

Deliverable *PROT1*
Status *Final*
Availability *Restricted*
Date *July 1999*

INCO COPERNICUS PL961104

Abstract:

This document contains the partial results of Task 8.1 (Integration of intermediate prototype), insofar as it provides a discursive overview of the implemented integration. This document and the program code jointly constitute the deliverable PROT1 (Integrated intermediate prototype for all three languages).

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

Contents iii
List of Figures iv
1 Introduction 1
2 Architecture 1
2.1 Interface Module 2
2.2 Text Structuring Module 2
2.3 Tactical Generator Module 3
2.4 Domain Model 3
3 Implementation 3
4 Communication 4
5 Conclusion 5
References 5

List of Figures

Figure 1: General view of integrated system 2
Figure 2: List of programs included in this directory 4

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 a formal model of 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).

The achievement of this end-goal supposes a number of closely-linked sub-tasks. These are performed or supported by various modules programmed within different AGILE work packages. Certain processes utilize resource modules previously generated outside of the project. The purpose of integration is to combine all of these modules and resources into a single functioning system.

In this report we describe the *implementation of integration* for the AGILE intermediate prototype.

2 Architecture

The basic modules and resources constituting the AGILE system are (Figure 1):

- Interface Module
- Text Structuring Module
- Tactical Generator Module (KPML)
- Domain Model (a resource module)

The resource modules used by the Tactical Generator are:

- Upper Model
- Grammars (Bulgarian, Czech and Russian)
- Lexicons (Bulgarian, Czech and Russian)
- Morphological Components (external for Bulgarian and Czech)

Finally, there is the Integration Module itself.

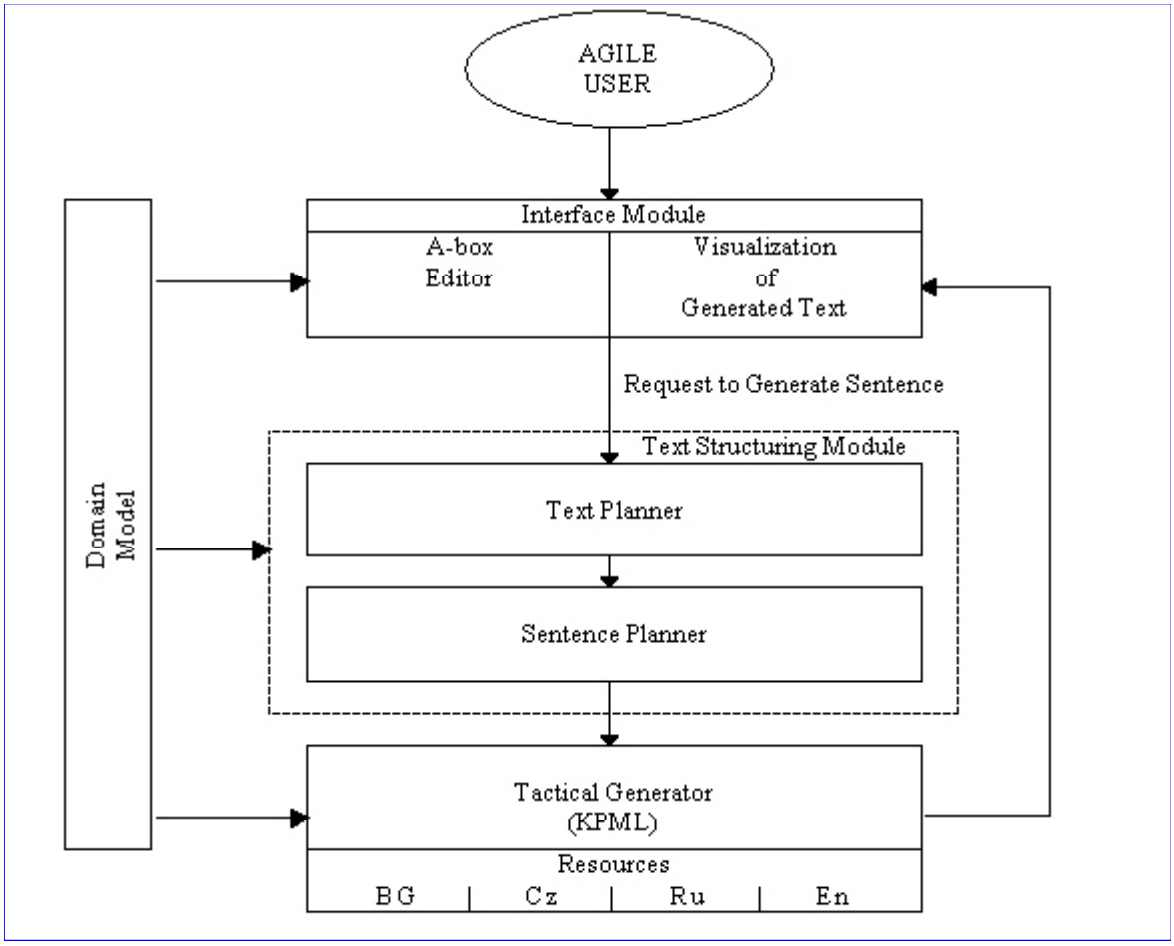

Figure 1: General view of integrated system

2.1 Interface Module

The 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, by building an A-box (See section 2.4.), 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 Interface module is implemented in Lisp and CLIM and described in deliverable [INTF1].

2.2 Text Structuring Module

The Text Structuring Module includes a *Text Planner* component that creates a text plan for realizing a given A-box, and a *Sentence Planner* component that creates sentence plans for individual sentences on the basis of the text plan. The A-box that the Text Planner takes as its input comes from the user interface. The SPLs provided by the Sentence Planner serve as input to a language-specific tactical generator.

The Text Structuring module is implemented in Lisp and is described in deliverables [TEXM1] and [TEXM2].

2.3 Tactical Generator Module

The Tactical Generator Module is the KPML (Komet-Penman Multilingual) system used in "black box" operation mode. The KPML development environment is a system for developing and maintaining large-scale sets of multilingual systemic-functional linguistic descriptions and for using such resources for text generation. The basic units manipulated by the system are grammatical systems, choosers, inquiries, lexical units, punctuation rules, and examples.

All the resources necessary for KPML to work and generate texts in Bulgarian, Czech and Russian are developed and described in deliverables [LEXN2] and [IMPL2].

The KPML system is implemented in Lisp and is described in document [GMD-304].

2.4 Domain Model

A domain model is a set of concepts for representing knowledge in a specific domain — in this case, the domain of CAD/CAM applications. This set of concepts is sometimes called a terminology, or "T-box". Using the concepts in the T-box, a model of the content of a specific text can be constructed: such a model is called an "A-box". To generate texts with the AGILE system, the author builds an A-box, using the interface developed in WP1, then calls the generator, which consults the A-box as it plans the general structure and the detailed wording of the text. To allow the interface and the generator to edit and consult the A-box, we have developed a domain model API (Application Programmer's Interface). The API also allows the definition of the T-box, which includes a set of concepts specific to the CAD/CAM domain; these concepts are linked to an "Upper Model"' (also part of the T-box), which represents the abstract semantic concepts that are expressed by the syntactic features of natural languages.

The Domain module is implemented in Lisp and CLIM and described in deliverable [MODL1].

3 Implementation

All modules are implemented in Lisp and CLIM. The AGILE Consortium is licensed by Harlequin for Lisp and CLIM - LispWorks (LW) versions 4 and 4.1.

Because of intractable problems with the support of cyrillic fonts in LW version 4.1, the integration of the AGILE intermediate prototype is implemented in LW version 4.0.1. The standard LW 4.0.1 version does not support cyrillic fonts either. However, for the needs of the AGILE project, Harlequin have been able to develop four "patch" files which provide support for cyrillic fonts and for use of characters from code pages other than code page 1252 (Latin I) while working with application frames. The patches must be loaded after starting LW. Without them the system does not support the Czech character set either. Harlequin are attempting to find a solution for LW 4.1.

In order to run the AGILE system, you first load LW and then start the program `a-integrate.lisp`. This program loads and configures all AGILE system modules, then starts the interface module to enable the user to start working with the AGILE system. Two conditions have to be met if the integrated AGILE system is to work properly:

1. All the programs which implement the Domain Module, Interface Module and Integrate Module must be in one and the same directory. The path to the directory is set as a value of two variables:

   `if::lpoint` by editing the program `a-integrate.lisp`

   `dm::*path*` by editing the program `agile2.lisp`

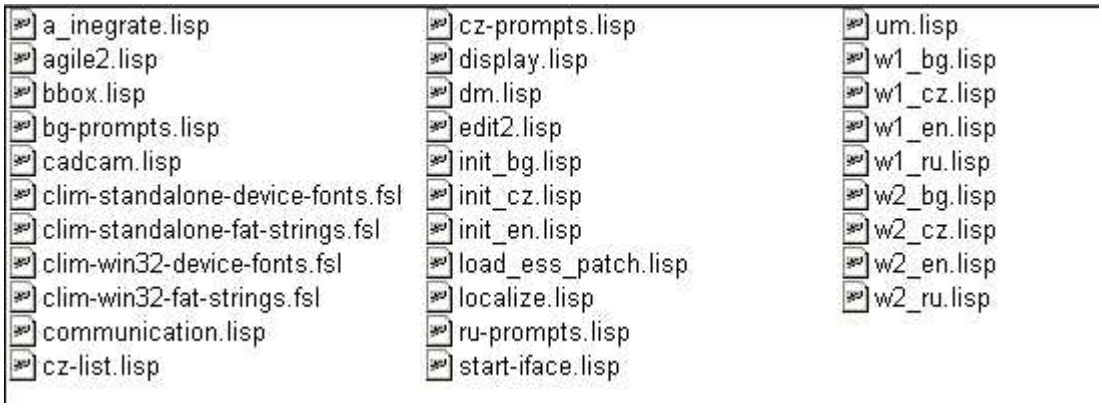

Figure 2: List of programs included in this directory

2. KPML and the resources for the Bulgarian, Czech, Russian and English languages have to be installed according to the instructions given at the KPML web site:

   `ftp://ftp.darmstadt.gmd.de/pub/komet/KPML-2.0/`

   The Test Structuring Module must be installed in the KPML subdirectory according to the instructions given in deliverable [TEXM2].

   The path to KPML must be set as value of the variable:

   `if::lpkpml` by editing the program `a-integrate.lisp`

The integration of the system pays special attention to the initialization of KPML. This must be such that KPML functions without the window interface, in "black box" mode. The initialization of the necessary variables and the loading of the appropriate language resources can be seen in the file `bbox.lisp`.

4 Communication

There are two types of communications in the system: communications between module and resources, and communications between non-resource modules.

The organisation of the first type of communications (Module<—>Resources) has been undertaken as part of the implementation of the module itself, and is not a matter for the present integration task. For example, the communications between A-box editor and Domain Model, between Domain Model and Text Structuring Module and between KPML and Language Resources have all been previously resolved in this manner. A description of these communications can be found in the deliverable relating to the relevant module.

Some of the second type of communications (Module<—>Module) have been covered in the implementation of the modules themselves; they are described in the relevant deliverables.

There remain three cases of communications between modules which are handled by the Integration Module:

- communication between the Interface Module and the Test Structuring Module through the structure `'langrecord`
- communication between the Text Structuring Module and the Tactical Generator (KPML) through the use of the funtion `'say` implemented in KPML `(kpml::say <spls form> :language <language identificator>)`
- communication between Tactical Generator (KPML) and Interface Module to visualize the generated text. The text generated by Tactical Generator is assigned to the variables `'text-a_en, 'text-b_en, 'text-a_bg` etc.

The implementation of the communications described above can be seen at Lisp level in the file `communication.lisp`, in the interface programs directory.

5 Conclusion

The integration of all the modules constituting the AGILE intermediate prototype has been described. The intermediate prototype can operate in Bulgarian, Czech, English and Russian.

References

[GMD-304] J. Bateman, *KPML Development Environment*, GMD-Studien Nr. 304, December 1996

[IMPL2] E. Andonova, J. Bateman, S. Hansen, J. Hana, I. Kruijff-Korbayová, G-J. Kruijff, K. Staykova, E. Sokolova, E. Teich, *Implementation of grammatical resources for the initial demonstrator*, AGILE deliverable 7.2, July 1999

[INTF1] St. Stankov, R. Power, T. Hartley, *Design specification for interface to demonstrator.* AGILE deliverable 1.1, July 1999

[LEXN2] H. Skoumalová, I. Kruijff-Korbayová, J. Hana, M. Malkovsky, M. Boldasov, D. Dochev, *Lexical and morphological resources for the final prototype*, AGILE deliverable 4.3, July 1999

[MODL1] R. Power, *Preliminary model of the CAD/CAM domain.* AGILE deliverable 2.1, June 1998

[TEXM1] J. Bateman, A. Hartley, I. Kruijff-Korbayová, D. Dochev, N. Gromova, J. Hana, S. Sharoff, L. Sokolova, *Generation of simple text structures in Bulgarian, Czech and Russian*. AGILE deliverable 5.1, June 1998. (Deliverable comprises TEXS1-Cz, TEXS1-Bu, TEXS1-Ru and TEXM1.)

[TEXM2] G-J. Kruijff, I. Kruijff-Korbayová, J. Bateman, *Text Structuring Module for the Intermediate Prototype* AGILE deliverable 5.2, July 1999