AGILE

Automatic Generation of Instructions in Languages of Eastern Europe

Title **_Evaluation of the intermediate prototype_**

Authors Serge Sharoff,
Donia Scott,
Anthony Hartley
Danail Dochev,
Jiri Hana,
Martin Cmejrek,
Ivana Kruijff-Korbayova,
Geert-Jan Kruijff

Deliverables *WP9: EVAL1-Bu, EVAL1-Cz, EVAL1-Ru*

Status *Final draft*

Availability *Public*

Date *2000-04-17*

INCO COPERNICUS PL961104

Abstract:

This document describes the deliverables EVAL1-Bu, EVAL1-Cz, EVAL1-Ru of work package 9, task 9.1 of the AGILE project. The objective of this workpackage consists in the assessment of the usability of the integrated system and the correctness and appropriateness of its output in three languages at the intermediate prototype stage in order to inform the subsequent refinement of the final prototype interface, generators and stylistic control mechanisms. The deliverable discloses the design of declarative evaluation of the output texts generated by the system and of operational evaluation that tests abilities of the integrated system with respect to creating and editing text specification models. We compare text specification models produced by evaluators to their original verbal specifications and analyse comments made by evaluators with respect to the usability of the intermediate prototype and the quality of texts generated by it. Evaluation results show that the intermediate prototype is efficient for making drafts of user manuals, but several improvements should be achieved in the final prototype.

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

Table of Content

Table of Figures

1. Introduction

The primary goal of AGILE consists of developing a suite of software tools to assist technical writers in the production of user manuals and in the CAD/CAM domain in selected languages of Eastern Europe (Bulgarian, Czech and Russian). This problem is approached by means of multilingual generation from a common semantic representation of the procedural aspects of the task of using such software tools. Multilingual documentation is thus generated directly from the user interface and domain task model, in contrast to the current practice where the initial documentation is produced in one language only and subsequently translated.

The objective for this workpackage consists of the assessment of the usability of the intermediate prototype and the correctness and appropriateness of its output in three languages. The results of this workpackage are intended to inform the subsequent refinement of the final prototype. The work in this package utilises resources developed in other workpackages, in particular, including lexicogrammatical resources, text structuring resources and interface localisation. These resources defined linguistic systems of particular languages, while the evaluation workpackage sets a test for instantiation of such systems, which have been used for generation of texts to accomplish practical purposes of experts involved in production of multilingual documentation.

The goal of the workpackage is an evaluation of:

1. the usability of the system for the tasks of production of user manuals;
2. abilities of the system for controlling multilingual generation in Bulgarian, Czech and Russian;
3. re-usability of specification models for different tasks, when an existing specification is extended or used as a template for another task;
4. robustness of interface operations;
5. the quality of the output texts in Bulgarian, Czech and Russian.

The Intermediate Prototype is capable for generation of instructional texts for complex procedures in two styles: personal and impersonal[1]. Examples of such texts are given in Appendix 5.

The system was evaluated by information technology (IT) specialists and by language specialists (with experience in writing or translating manuals). Some evaluators combined both types of experience, some were either IT or language specialists.

The criteria for evaluation of the Intermediate Prototype of AGILE are roughly grouped into two types:

1. specification of the text content should be effective, so that the user's efforts on specification should be commensurable with user's goals;

2. texts produced by the generation system should be sufficiently good in all target languages to be, at least, a draft of high-quality manuals.

The GIST and DRAFTER projects developed in ITRI earlier provided experience for design decisions made in AGILE[2]. Also, the methodologies of their evaluation served as a basis for designing a methodology for evaluating AGILE[3].

[1] Styles available in the Intermediate Prototype are discussed in the TEXS2 deliverable

[2] See the GIST and DRAFTER home pages at http://www.itri.bton.ac.uk/projects.html.

2. Evaluation methodology

The evaluation workpackage included two types of evaluation tasks:

1. operational evaluation that tests abilities of the integrated system with respect to creating and editing text specification models;
2. declarative evaluation of the output texts generated by the system.

Accordingly, two groups of evaluators were selected for each localised version: at the Bulgarian Academy of Sciences, Sofia (Bg), Charles University, Prague (Cz), and Russian Research Institute for Artificial Intelligence, Moscow (Ru).

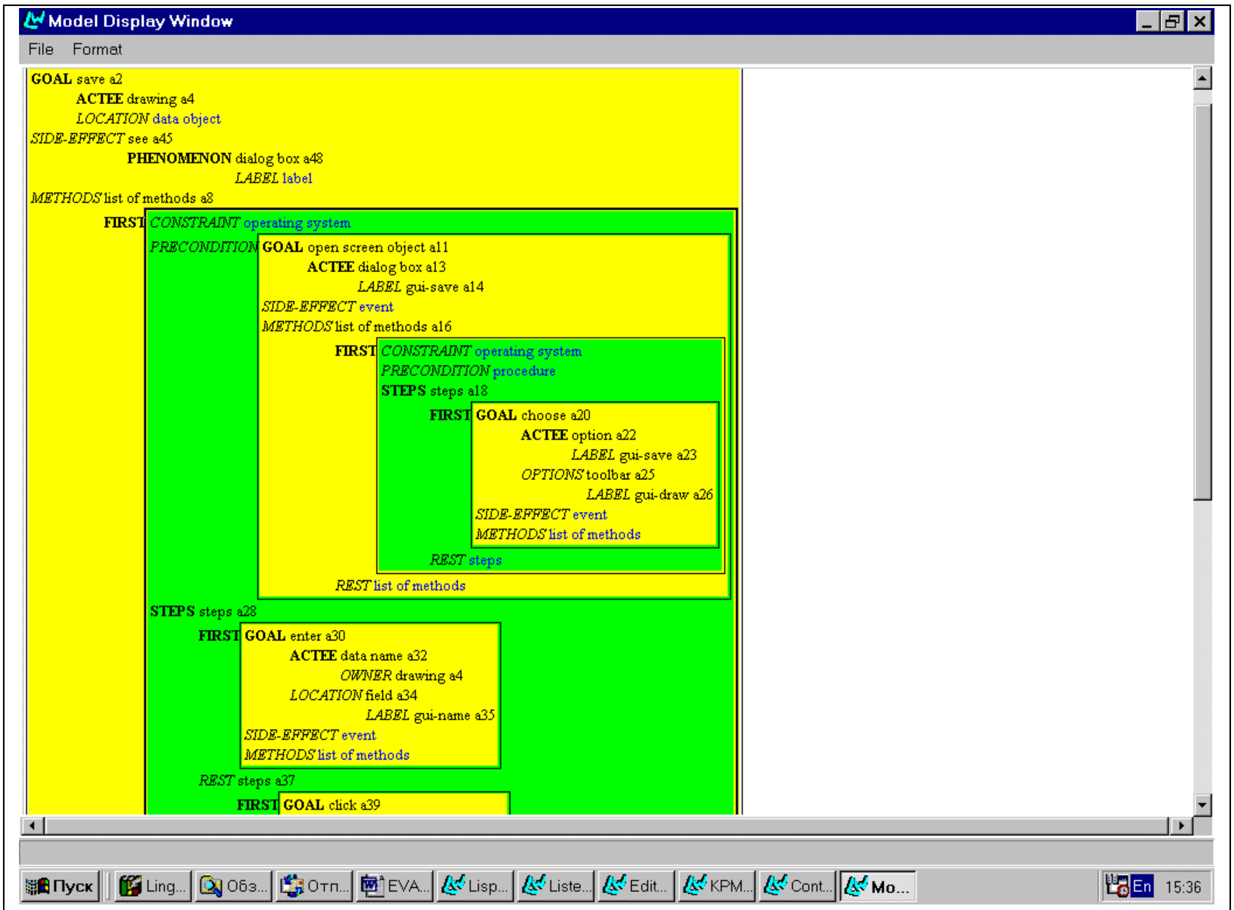
The first group (Group A) was composed of professionals in Information Technology. Their task was to use the system (after training) to specify formal models which correspond to the content of a number of procedural instructions. Subjects in this group were used to evaluate the first four evaluation goals (usability of the system; reusability of specification models; multilingual capabilities; robustness). Groups of evaluators at each site consisted of 3 persons and used a respective localised version of the system (i.e., Bulgarian, Czech and Russian). Each subject worked on the version available in their own native language, and each was experienced in the use of windows-based computer applications. The Bulgarian group of evaluators included an associate professor and a researcher in Computer Science, both with experience in computational linguistics; and a graduate student in information technologies. The Czech group of evaluators included a teacher of mathematics, a graduate student, and a linguistics researcher. The Russian group of evaluators included a university professor with experience in interfaces and knowledge representation; and two graduate students in Information Technology.

The second group (Group B) was composed of persons experienced in writing and/or translating software documentation. Their task was to rate the output of the system against a number of pertinent linguistic parameters. Subjects in this group were used to evaluate the quality of the output texts in Bulgarian, Czech and Russian. The Bulgarian group of evaluators included 3 persons: an associate professor and two programmers experienced in production of software manuals. The Czech group of evaluators included 4 persons: a graduate student, an IT consultant, 2 programmers; all writers of various technical documentation). The Russian group of evaluators included 3 persons: a translator/editor in various IT fields, a translator/technical writer, and a technical writer in the CAD domain.

2.1 Operational evaluation

The operational evaluation included several stages, which were arranged in the identical manner at each site. Firstly, evaluators received training in basic concepts of the AGILE project and the prototype system and acquired knowledge editing skills for using the interface. Then, they performed set tasks, each of which addressed key tasks in the authoring of user manuals (specifically, creating, editing and revising the content of the manuals.). The overall task was to produce, with the aid of AGILE, parts of the User Manual for a fictitious CAD/CAM system. This stage of evaluation also addressed multilinguality through exchange between different language groups of models produced by speakers of a different language.

[3] The evaluation methodology is reported in (Power, Scott, 1997)



```
GOAL save a2
      ACTEE drawing a4
      LOCATION data object
SIDE-EFFECT see a45
             PHENOMENON dialog box a48
                           LABEL label
METHODS list of methods a8
          FIRST CONSTRAINT operating system
                PRECONDITION GOAL open screen object a11
                                   ACTEE dialog box a13
                                          LABEL gui-save a14
                             SIDE-EFFECT event
                             METHODS list of methods a16
                                       FIRST CONSTRAINT operating system
                                             PRECONDITION procedure
                                             STEPS steps a18
                                                   FIRST GOAL choose a20
                                                               ACTEE option a22
                                                                      LABEL gui-save a23
                                                               OPTIONS toolbar a25
                                                                        LABEL gui-draw a26
                                                         SIDE-EFFECT event
                                                         METHODS list of methods
                                                   REST steps
                                       REST list of methods
                STEPS steps a28
                      FIRST GOAL enter a30
                                  ACTEE data name a32
                                         OWNER drawing a4
                                  LOCATION field a34
                                            LABEL gui-name a35
                            SIDE-EFFECT event
                            METHODS list of methods
                      REST steps a37
                            FIRST GOAL click a39
```

Figure 1. The model specification editor

2.1.1 Training evaluators

Since none of the subjects had any previous experience of a system such as AGILE, the evaluation was preceded by an in-depth training -- both in the underlying concepts and in the use of the system itself. To support this we created a Conceptual Tutorial, which introduces basic concepts of authoring documents in AGILE, and a Training Manual, which defines methods for specification of a fragment of a manual in close resemblance to real authoring conditions.

Each site chose Group A evaluators all of whom were professionals in information technology. At the initial meeting, they were briefly introduced into basic concepts of AGILE and received the localised versions of the Conceptual Tutorial to study at home prior to the training session. Evaluators at Bg and Cz sites also received the User Manual and the Training Manual.

The training session started with a discussion of the basic ideas of the project as they are presented in the Conceptual Tutorial, as well as general problems in production of manuals and problems in knowledge editing. The training continued with presentation of the Intermediate Prototype. Czech evaluators spend some time playing with the system and trying to create a-boxes of their own.

The interactive interface of the Knowledge Editor of AGILE allows incremental development of a plan for achieving user's goals and generation of the corresponding text in Bulgarian, Czech and Russian in respective language display windows. Figure 1 presents a formal specification model corresponding to a simple instruction.

The target instructions for the simple specification above are:

(1)

(a) (En) *To save a drawing:*
First choose Save Drawing from the Draw toolbar in order to open the Save As dialogue box.
Enter the name of the drawing in the Name field.
Click on the Save button.
The dialog box appears.

(b) (Bg) *Запазване на чертеж*
Изберете опцията Save от функционалния ред Draw, за да отворите диалоговия прозорец Save As.
Въведете име на чертеж в полето Name.
Натиснете бутона Save.
Виждате диалоговия прозорец

(c) (Cz) *Uložení obrázku*
Vyberte volbu Uložit z nástrojového panelu Kresli pro otevření dialogového panelu Uložit.
Zadejte název obrázku v poli Jméno.
Klikněte na tlačítko Uložit.
Dialogový panel zmizí.

(d) (Ru) *Сохранение рисунка*
Сначала выберите пункт Save Drawing в панели инструментов Draw, чтобы открыть диалоговое окно Save As.
Введите имя рисунка в поле Name.
Нажмите кнопку Save.
Появится диалоговое окно.

After presentation of the interface, evaluators were instructed on methods for specification of a fragment of a manual in close resemblance to real conditions for authoring of multilingual user manuals. This stage of training was guided by the Training Manual and included two basic phases:

1. modifying a text model specification in order to extend an existing procedure; and
2. building a text model specification in order to author a new procedural content.

During this stage, the evaluators were given the opportunity to familiarise themselves with the interface by following to steps defined in the Training Manual. After completion of the two phases defined by the Training Manual, they continued training to perform actions typical for creating text specification models (also referred to below as A-boxes) and not covered by the Training Manual, for example, reordering a sequence of steps, choosing proper domain concepts, checking correctness of the produced model, and so on. In this activity, the evaluators consulted localised versions of the User Manual of AGILE, as well as on-hand AGILE team members. The duration of the Training session varied across sites from 4-6 hours. The successful completion of the training session means that evaluators are trained with the interface capabilities and are able to test abilities of the Intermediate Prototype.

Save a drawing

First open the *Save As* dialogue box.

Choose the *Save Drawing* option from the *Draw* toolbar.

Enter a name of the drawing in the *Name* field.

Click the *Save* button.

A dialogue box appears.

Figure 2 A text specification example

2.1.2 Testing stage

The testing of the interface was carried out on another day. Unlike the Training session, during which evaluators were guided by the personnel responsible for evaluation at each site, the Testing session involved independent activity of evaluators to accomplish tasks they were assigned. AGILE team persons monitored their actions, but were not allowed to help them.

When the evaluators executed their tasks, they used a semi-formal verbal text specification notation, which was developed to ensure unambiguous understanding of the content to be specified. Verbal text content specifications are shown here framed in a box, whose bold first line expresses the task of a procedure. Steps are expressed by imperative clauses, their side-effects by indicative clauses. Substeps which constitute a method for achieving a complex instruction step are indented. A precondition for a sequence of steps is marked by 'First'. A constraint on the operating system is marked by 'Using' followed by its name. New entities are introduced into instructions using the indefinite article; the definite article marks a previously mentioned entity (except labeled entities). Labels of entities are marked by the italic script. Since evaluators used localised versions of the interface with names of concepts of the Domain Model (DM) from their respective languages, they received both the English original specifications and their translations. A specification for the example (1) is given in Figure 2. The complete set of specifications in English used for evaluation is given in Appendix 3.

The testing session comprised four tasks, each of which had a time limit for its accomplishing (specified in brackets):

Task 1: Load and edit two simple text specification models (20 min)

Task 2: Create two new text specification models (40 min)

Task 3: Create two new bigger text specification models (120 min)

Task 4: Load and edit two bigger text specification models (60 min)

The text specification models included for Task 1 were developed by the AGILE team prior to evaluation and included several simple steps. Their verbal text content specifications were not available to the subjects, who understood the original content by reading the screen presentation. The verbal text content specification for the second model used for Task 1 is given in Figure 3.

Create a style of a multiline

First open-screen-object the *Multiline Styles* dialog box

Click the *Element Properties* button.

Click the *OK* button in the *Multiline Styles* dialogue box. The *Multiline Styles* dialogue box disappears.

Figure 3 Task 1 specification example

Draw a line-arc-polyline

- First, start the *PLINE* command.
- Draw a line segment.
- Switch to the *Arc* mode.
    - Enter the *a* command line.
    - The *Arc mode confirmation* dialog box appears.
- Draw an arc segment.
- Switch to the *Line* mode.
    - Enter the command line *l*.
    - The *Line mode confirmation* dialog box appears.
- Draw a line segment.
- End the polyline.
    - Press the *Return* key.

Figure 4 Task 2 specification example

The task of evaluators was to identify places for corrections and introduce them, for example:

> Add one more step, as the second step for achieving the topmost goal: “Enter the offset of the elements in the *Element Properties* dialogue box”.

Task 2 involved development of two text specification models consisting of several steps with low complexity of embedded submethods. One text specification model was identical for all sites, and one model was specific for each site. The verbal text content specification used by all sites in Task 2 is given in Figure 4.

Task 3 also involved development of one text specification model that was identical for all sites, and one model that was specific to a site. Both models were designed as extensions of models developed at Task 2 (site-specific models for Task 3 also extended models developed at Task 2 by the same site). Evaluators were allowed either to start development from scratch or to extend the existing models developed at Task 2. The verbal text content specification used by all sites in Task 3 is shown in Figure 5.

At the end of Task 3, site-specific models were sent to two other sites. Task 4 involved generation from the received models into native languages of receivers to test the models. The models were also used for alteration of their content according to the specified instructions, for example:

> Change the description in order to specify justification of the multiline between the steps for specifying the properties of the multiline start and end points.

Draw a line-arc-polyline

First start the *Pline* command.

Using the *Windows* operating system: choose the *Polyline* option from the *Polyline flyout* on the *Draw* toolbar.

Using the *DOS* or *UNIX* operating system: choose the *Polyline* option from the *Draw* menu.

Draw a line segment.

Specify the start point of the line segment.

Specify the endpoint of the line segment.

Draw an arc [segment].

First switch to the *Arc* mode.

Enter the command line *a*. The *Arc mode confirmation* dialog box appears.

Choose the *OK* button in the *Arc mode confirmation* dialog box. The *Arc mode confirmation* dialog box disappears.

Specify the endpoint of the arc.

Draw a line segment.

First switch to the *Line* mode.

Enter the command line *l*. The *Line mode confirmation* dialog box appears.

Choose the *OK* button in the *Line mode confirmation* dialog box. The *Line mode confirmation* dialog box appears.

Enter the distance from the endpoint of the arc to the line.

Enter the angle from the endpoint of the arc to the line.

End the polyline.

Press the *Return* key.

Figure 5 Task 3 specification example

This task simulated exchange of models between offices of large company involved in multilingual authoring. Table 1 depicts distribution of texts between partners (specifications of complete texts are given in the Appendix 3).

	Bg	Cz	Ru
Produced	B-5	B-4	B-3
Changed	B-3, B-4	B-3, B-5	B-4, B-5

Table 1 Distribution of texts between partners

2.1.3 Filling questionaire

When the evaluators performed their training and testing tasks, they were asked to comment on their experience with the system. The comments were written as short notices and answer to questionnaire (the questionnaire is included in Appendix 2).

In filling the questionnaire, the evaluators provided objective information about conditions in which they conducted evaluation, including the computer configuration for evaluation and time spent on accomplishing each task. The subjective judgments of evaluators rated the usability of the Intermediate Prototype using the following scale:

Score	Explanation
A	Excellent
B	Good
C	Poor
D	Useless

The parameters by which the usability of the system was rated were the following:

Ref	Question
1	Intuitiveness of operations with the interface (menus, dialog boxes, etc)
2	Abilities of the interface for creating and modifying a text content.
3	Abilities of the interface for presentation of a text content.
4	Intuitiveness of content specification notions (goal, step, precondition, etc)
5	Intuitiveness of CADCAM notions (click, button, multiline, etc)
6	Robustness of interface operations
7	Does the generated output express the content you specified?
8	Adequacy of the user documentation

Each item in the list received a mark from the set specified above. Also, evaluators wrote free-text comments about the usability of the system with respect to these parameters. Their comments are summarised in Section 3.

2.2 Declarative evaluation

Four texts were prepared for the declarative evaluation. Three were generated by AGILE, two of these were in the personal style, a one was in the non-personal style. The fourth was a procedural instruction taken directly from the AUTOCAD manual. Group B evaluators were language professionals. They evaluated texts in their own native language and told to assume that the texts present a human-authored draft for inclusion into a manual. The identity of the texts, i.e. which text was taken from another source and which should be judged, was not revealed. The pretext for Group B evaluators was evaluation of writing qualities of a newly appointed translator/technical writer. This ensured that no bias pro or against computer-produced texts could be exhibited by evaluators.

Group B evaluation consisted of two phases. During the first phase, texts were evaluated as a whole using the following scale:

Score	Explanation
A	Excellent
B	Good
C	Poor
D	Useless

The text quality was rated with respect to the following parameters: how good do you consider the respective text to be in:

- organising the information to help the reader execute the instructions?

- wording the information to help the reader understand the instructions?
- meeting the standard for inclusion in a high-quality manual?

The second phase involved evaluation of individual parts of each text, so that each sentence was judged using the following scale:

Score	**Explanation**
A	The text is entirely satisfactory.
B	Minor revision is desirable.
C	Minor revision is necessary.
D	Major revision is necessary.

3. Results of evaluation

3.1 Analysis of Tasks completed by Group A evaluators

All Group A evaluators completed their tasks assigned for the Testing Session, however, several of them did not fit into the planned time limits. Table 2 depicts duration of task time allocation (in minutes) per site per evaluator. The cases, when extra time was required for an evaluator to complete a task, are shown in italic.

	Bg			Cz			Ru			Average
Task 1	20	20	30	20	20	20	20	20	20	21.11
Task 2	*70*	40	*120*	35	40	30	*50*	35	*45*	51.67
Task 3	100	120	100	95	*120*	80	110	90	90	100.56
Task 4	30	27	30	55	60	60	25	25	25	37.44

Table 2 Task time allocation

Three evaluators (two Bg, one Ru) failed to complete Task 2. All of them made similar errors: expanding the list of methods instead of expanding the list of steps during development of text specification models. One Bg and one Ru evaluator detected their error, when they did Task 2, and tried to correct it. The Ru evaluator succeeded, so that the produced A-box represented correctly the structure of the verbal text content specification for Task 2; the Bg evaluator did partial correction of the error. However, error correction significantly increased time spent on the task. For another Bg evaluator, the error propagated to the respective Task 3 A-box as well. One Czech evaluator failed to create the second A-box of Task 3 in time.

We performed an in-depth analysis of A-boxes produced by each evaluator within Task 3 (cf. the description of Task 3 in Section 2.1.2). The Task 3 A-boxes were compared against the verbal specifications received by evaluators and errors were counted.

All errors possible in development of A-boxes can be classified into the following classes:

A. Concept-name errors, e.g., use of SPECIFY instead DEFINE, or CHOOSE instead of PRESS, or PRESS instead of ENTER, or SEE instead of NOT-SEE, or LINE instead of LINE-SEGMENT, or GUI-OK instead of GUI-ENTER, etc.

B. Co-reference errors

1. a new concept instance is created instead of (re)using an identifier-copy, thus no co-reference information is preserved between two expressions of the same instance;
2. an identifier-copy is created instead of a new concept instance, thus an erroneous co-reference is introduced between two instances.

C. Configuration errors

i. some content is missing, which means that the given content is not modelled (or reflected) anywhere in the a-box at all; for example, a side-effect is omitted, or a (list of) step(s) is omitted, a constraint is omitted, etc.; hence, there are the following subclasses:

1. missing precondition
2. missing constraint
3. missing side-effect
4. missing procedure ("step")
5. missing method ("alternative")
6. missing filler of an optional slot which was specified in the verbal content specification

ii. some content is modelled incorrectly, i.e. by a wrong configurational concept or a wrong constellation of configurational concepts:

1. a side-effect is modelled as a (separate) PROCEDURE (i.e. the GOAL of a procedure) rather than by filling a PROCEDURE's SIDE-EFFECT slot
2. a side-effect is modelled as the SIDE-EFFECT of the wrong PROCEDURE, for example the higher-up one instead of the last procedure within a list of substeps
3. a precondition is modelled as a PROCEDURE (in a METHOD's SUBSTEPS PROCEDURE-LIST) rather than as a METHOD's PRECONDITION
4. a `subsequent' step is modelled as a (possibly embedded) METHOD rather than as a PROCEDURE (in the `current' PROCEDURE-LIST)
5. a step is modelled as a PRECONDITION
6. an alternative is modelled as a PROCEDURE in a PROCEDURE-LIST rather than as a separate METHOD in a (higher-up) METHOD-LIST
7. other cases.

3.1.1 Comparison of Task 3 A-boxes

The following tables show number and distribution of each type of error for each of the two A-boxes and overall distribution of errors per site.

	Abox1			Abox2			Aboxes 1 and 2		
Errors	# of errors	**relative %**	**deviation**	# of errors	**relative %**	**deviation**	**# of errors**	**relative %**	**deviation**
A	6	22.22%	6.84%	2	7.69%	7.69%	8	15.09%	7.13%
B1	15	55.56%	17.09%	18	69.23%	30.77%	33	62.26%	15.51%
B2	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci1	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci2	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci3	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci4	0	0.00%	0.00%	1	3.85%	3.85%	1	1.89%	1.96%
Ci5	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci6	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii1	4	14.81%	15.95%	0	0.00%	0.00%	4	7.55%	7.84%
Cii2	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii3	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii4	1	3.70%	3.99%	2	7.69%	7.69%	3	5.66%	5.66%
Cii5	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii6	1	3.70%	3.99%	3	11.54%	11.54%	4	7.55%	7.55%
Cii-other	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Total errors	27			26			53		

Table 3 Summary for Bulgarian evaluators

The results of Bulgarian evaluators can be summarized as follows:

1. The number and distribution of errors are remarkably similar for each of the two A-boxes.
2. All evaluators made small number of a single concept-name error, e.g., use of CHOOSE instead of PRESS (error type A).
3. In nearly all cases the evaluators have not used co-references – they have used every time a new concept instance, instead of re-using a copy of an already described concept instance (error type B1).
4. One evaluator missed a procedure step in the second text specification model (error type Ci4).
5. One evaluator described side effects as separate procedure steps in the first text specification model (error type Cii1). The same evaluator described correctly the side effects in the second text specification model.

6. Two evaluators have made 3 configuration errors in all for both texts, describing a subsequent step as a METHOD rather than as a PROCEDURE in the current list of procedural steps (error type Cii4).
7. Two evaluators have made 4 configuration errors in all for both texts, describing next alternative as a PROCEDURE in the current list of procedural steps rather than as a separate embedded METHOD (error type Cii6).

It seems that the last two configuration errors are due more to inconvenient operation with the interface rather than to unclearness of the concepts of PROCEDURE and METHOD, as these errors have not been made in all possible cases in the texts.

CU	**Abox1**			**Abox2**			**Aboxes 1 and 2**		
Errors	# of errors	**relative %**	**Deviation**	# of errors	**relative %**	**deviation**	**# of errors**	**relative %**	**deviation**
A	8	44.44%	22.22%	4	28.57%	28.57%	12	37.50%	6.94%
B1	9	50.00%	16.67%	6	42.86%	42.86%	15	46.88%	13.54%
B2	0	0.00%	0.00%	4	28.57%	28.57%	4	12.50%	12.50%
Ci1	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci2	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci3	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci4	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci5	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci6	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii1	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii2	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii3	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii4	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii5	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii6	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii-other	1	5.56%	6.94%	0	0.00%	0.00%	1	3.13%	11.16%
Total errors	18			14			32		

Table 4 Summary for Czech evaluators

The results of Czech evaluators can be summarized as follows:

1. There were four concepts that caused together 25 occurrences of A type errors. However for three concepts, the errors were caused by wrong translation of Czech text specifications for evaluators. If we do not count these errors, the result is 2 errors (2 evaluators, each 1 error) caused by using concepts polyline instead of line-arc-polyline.
2. There were 19 co-indexation errors for all evaluators (B1 and B2 type errors). Evaluators did mostly the same errors. Majority of the errors of this category were caused by missing

co-indexation of items. Only about 20 % of co-indexation errors were caused by co-indexing items that should not be. These errors were present only in the second A-box.

3. There was one uncategorized error – one evaluator modeled expression "distance from A to B" as "distance from B to A" (C-other type error). Which from logical point of view has the same meaning, but the generated text is, of course, different.
4. Four B2 type errors were made in the second A-box. There were two bits of identical text in its specification: "Choose the Multiline Styles option", and the two evaluators decided to model this by co-indexation (probably due to wrong insufficient training, they did not know that user actions should not be co-indexed). This gives a rather high relative percentage, because of the low number of other errors.
5. One evaluator failed to create second A-box in time.

If we do not count errors caused by errors in translation of text specifications, most of the errors occurred in missing co-indexation. The evaluators had no problem with structure of A-boxes, with understanding of DM concepts, etc. Their performance in development of A-boxes was also roughly similar.

	Abox1			**Abox2**			**Aboxes 1 and 2**		
Errors	# of errors	**relative %**	**deviation**	# of errors	**relative %**	**deviation**	**# of errors**	**relative %**	**deviation**
A	3	20.00%	22.86%	6	23.08%	23.08%	9	21.95%	21.95%
B1	12	80.00%	22.86%	15	57.69%	42.31%	27	65.85%	34.15%
B2	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci1	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci2	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci3	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci4	0	0.00%	0.00%	2	7.69%	7.69%	2	4.88%	4.88%
Ci5	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci6	0	0.00%	0.00%	2	7.69%	7.69%	2	4.88%	4.88%
Cii1	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii2	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii3	0	0.00%	0.00%	1	3.85%	8.65%	1	2.44%	5.25%
Cii4	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii5	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii6	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Cii-other	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Total errors	15			26			41		

Table 5 Summary for Russian evaluators

The results of Russian evaluators can be summarized as follows:

1. The most of cases of errors are caused by improper co-indexation (27 errors counting to 66%). One evaluator made errors of only this kind. Another evaluator had 92% of such errors. (B1 type error)
2. One evaluator did a lot of A type errors (probably, she was less comfortable with the set of DM concepts). Another reason is some infelicities in translation of concept names into Russian.
3. Other errors (C.i.4: missed steps in a procedure, C.i.6: missed fillers of an optional slot, C.ii.3: a step instead of a precondition) are specific to evaluators. Probably, they were caused by lack of attention on the side of evaluators.

3.1.2 Joint comparison for all sites

No configuration errors of the following types:

(C.i.1) missing precondition,

(C.i.2) missing constraint,

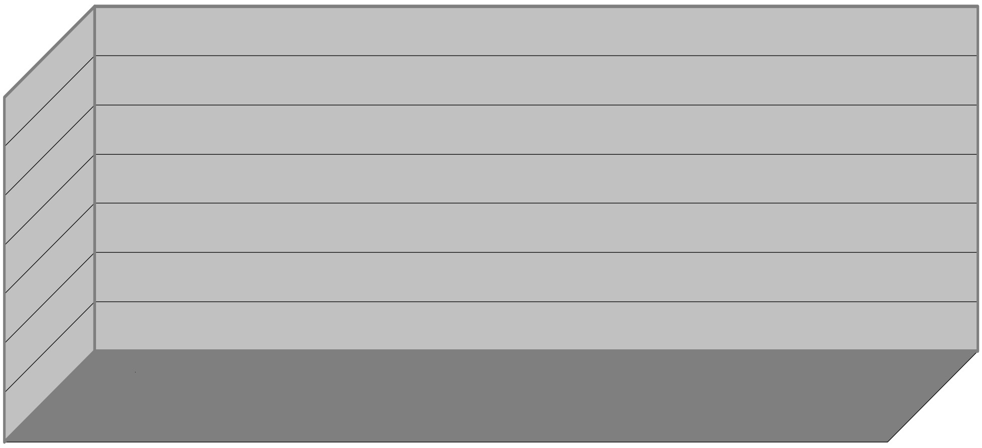
(C.i.3) missing side-effect,

(C.i.5) missing method,

(C.ii.2) a side-effect is modelled as the side-effect of a wrong procedure,

(C.ii.5) a step is modelled as a precondition

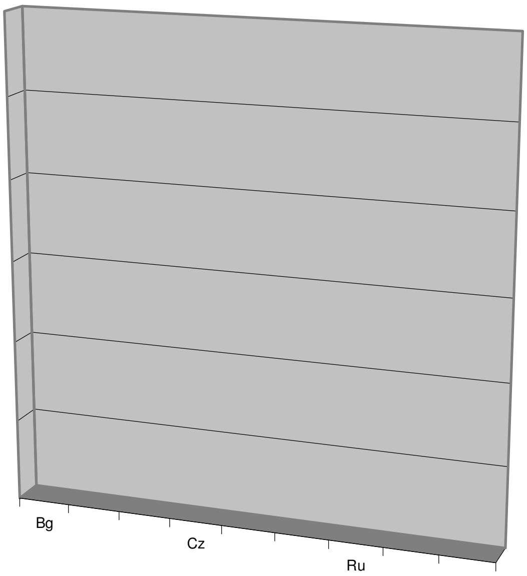
were found in any A-box across all sites. They are omitted from the joint tables below.



tween English and target languages. Another reason for this error concerns lack of documentation on the DM concepts (cf. comment (1), Section 3.2.2, and a comment in Section 3.2.8).

The results of analysis of Task 3 A-boxes also provide data for comparing time spent for each evaluator for the Task 3 and number of errors in produced A-boxes. For each evaluator, Table 6 shows the amount of time spent on development of Task 3 A-boxes, the computer configuration (the processor, its clock rate and memory) and number of errors done by this evaluator in the two A-boxes of Task 3.

The comparison shows that the computer configuration does not affect significantly the amount of time spent on development of relatively complex A-boxes (neither processor speed nor memory amount are important, at least, within the range of hardware used for evaluation). Also, more time spent on development does not necessarily mean A-boxes with a smaller error rate. However, this is related to the fact that the most of errors pertain to the B1 type, so they were not detected and judged as errors by the most of evaluators and they did not spend additional time on improvement of their A-boxes.



Score value	Explanation	Numerical value
A	Excellent	1
B	Good	2
C	Poor	3
D	Useless	4

Ref	Question	Marks for each site			
		Bg	Cz	Ru	Avg
1	Intuitiveness of operations with the interface (menus, dialog boxes, etc)	ABC	BBB	BBB	2
2	Abilities of the interface for creating and modifying a text content.	BCC	BCC	BBB	2.44
3	Abilities of the interface for presentation of a text content.	AAC	BBC	BBB	2
4	Intuitiveness of content specification notions (goal, step, precondition, etc)	ABC	ABB	ABB	1.78
5	Intuitiveness of CADCAM notions (click, button, multiline, etc)	AAB	ABB	BBB	1.67
6	Robustness of interface operations	ABB	ACC	CCC	2.33
7	Does the generated output express the content you specified?	ABC	BBB	AAB	1.78
8	Adequacy of the user documentation	ABC	BBC	AAA	1.78

Table 7 Group A questionnaire rates

The evaluators from Group A provided a lot of valuable comments with respect to capabilities of the AGILE interface for specifying a text content. The comments received from evaluators are structured according to items in the questionnaire. For each comments, we give number of evaluators that raised it and site(s) at which it was raised.

3.2.1 Intuitiveness of operations with the interface

(1) Interface of the system should follow the standards of MS Windows applications:

(a) standard labels, for example, OK instead of ‘Click here to accept values’. [one comment, Ru]

(b) It is hard to control the application from the keyboard. There are no keyboard shortcuts. [two comments, Ru]

(c) Mouse behaviour is not standard: left and right mouse buttons produce the same action. [one comment, Cz]

(d) The current file name is not displayed on the title bar of the AGILE window. It should be there, so that the user knows what file is loaded. [two comments, Bg and Ru]

(e) The current character size should be checked in the ‘Format | Size’ menu, so that one would not have to guess or experiment. (The same applies to the typeface). [one comment, Bg]

(f) The .abx extension is not added automatically to the file name, when the model is saved. [two comments, Bg and Ru]

(2) The Generation Options dialog box requires several improvements: 1) it appears in the upper left corner, not in the center or the current focus of attention of the user; 2) but-

tons are located in non-standard positions and have non-standard labels; 3) there is no default filling of options and there is no the "Fill default" button. [one comment, Ru]

(3) The system works slowly, for example, when it shows choice lists, generates, updates fonts, etc. When the interface is busy, it should show "signs of life" (it would be good to display an indicator "I'm doing X" or a progress indicator "X% is completed".) [three comments, Bg, Cz and Ru]

3.2.2 Abilities of the interface for creating and modifying a text content

(1) Some Domain Model notions can't be understood by a user without a hint, for example, the difference between Open and Open-Screen-Object. There is a discrepancy between concept names and words that express them in text. For example, a natural expression of mouse clicking in Russian as *нажать* (*press*) should not be selected for this action, since the concept НАЖАТЬ is related to keyboard keys only. Instead, a user should choose *щелкнуть* (*click*), which is a DM concept that is rendered as *нажать* (*press*) by the Russian grammar. The same issues occur in Czech, for example, it is hard to know which item in the list is the correct one for *stiskout tlačítko, vybrat tlačítko*. For these reasons, it would be good to assign a longer description with examples and an icon (good for gui-labels) to every T-box term. The icon could be in the menu in front of the name. The comment could be shown as a tooltip when the item is pointed at by the mouse, and could be displayed on the status bar when the item is in focus. The description and examples could be shown in a popup window when F1 is pressed. [two comments, Cz and Ru]

(2) Building an A-box requires a lot of clicking (scrolling, selecting from menus, etc.) (an evaluator expressed his experience as "I got a tennis elbow"). The whole structure is large and complicated, it is hard to find anything if you want to co-index something. [two comments, Cz and Ru]

(3) It should be possible to enter the concept names using the keyboard. (One could enter several first letters of a concept to bring the complete name, in the case, if concept names are long, for example, precondition). [one comment, Ru]

(4) Sometimes, a selection list for some concept types contains concepts which are impossible in a given context; for example, *style* is possible only for lines (or other quite specific objects), but the interface offers all configured objects as its OWNER. [two comments, both Ru]

(5) It would be better to replace some specification parts using one action instead of deleting the existing content and inserting a new one. [one comment, Ru]

(6) There is no possibility to add a new label, for example, for another menu option (in specification tasks, for example, there was no concept for the label 'File'). [one comment, Ru]

(7) There is a possibility to develop by mistake a specification structure using Methods instead of Steps. There is no possibility to correct the wrong structure in an easy way. [one comment, Ru]

(8) Paste, as the least frequent choice, should be added to the end of the list of concepts. [one comment, Ru]

(9) It is not very convenient to work without UNDO-command.(A non-advanced user can loose all the text after doing CUT -operation and loosing a content of buffer). [four comments, all sites]

(10) Vertical scrolling moves not smoothly and sometimes it jumps not according to scroll bar. [one comment, Ru]

(11) A Lisp-like presentation of lists is unsatisfactory - it is not intuitive and is hard to read and navigate. They proposed list items to be equally indented and numbered. [four comments, Bg and Ru]

(12) A-boxes are very large; therefore it is hard to manage them. Users proposed adding possibility to collapse and expand nested A-boxes and possibility to switch between displaying A-box and some text assigned to it ("Click on the OK button" instead of the corresponding A-box). [three comments, Bg and Cz]

(13) Too much redundant information is visible on the screen (all these labels: Goal, Methods, Side-effect, etc.). The important information is hard to pick out. [one comment, Bg]

(14) A new step can be easily added only as the last step of a method. Several click and paste operations are required for inserting a step in the middle of a sequence. It'd be better to insert steps in the middle by a single click. [one comment, Ru]

(15) Within the A-box editor, it is better to have an intelligent alignment of the structure being edited into the center of the window. Now the focus of a complex A-box tends to move to the bottom right corner of the window. [three comments, Bg and Ru]

(16) The color schema is convenient, but there is no possibility to alter it. [one comment, Ru]

(17) It is better to display Actee always in the first position of action slots. [two comments, Ru]

(18) Long names of specification concepts cause wide structures that do not fit into screen. [two comments, Bg and Ru]

(19) The line being edited should be highlighted. [one comment, Ru]

(20) Currently, labels for digits in selection lists are ordered using the alphabetic order, but it would be better to order them by value. [one comment, Ru]

(21) The labels for concepts in the specification should not use an unknown abbreviation “Gui”, but be prefixed with an icon, for example, ⌦. [one comment, Ru]

3.2.3 Abilities of the interface for presentation of a text content

(1) There is no possibility to have a connection between sentences of the generated text and the respective A-box configuration (a hypertext-like switch). [two comments, Cz and Ru]

(2) It would be nice to show changes between the last two generations. [two comments, Cz and Ru]

(3) The process of generation does not bring the output text window in the front of the screen, so a user can desperately wait for generation results. [three comments, Bg and Ru]

3.2.4 Adequacy of content specification notions

(1) Abilities of the system with respect to content specification are restricted in certain respects; for example, it lacks the possibility to express conditional actions: if something happens, do something. It would be also good to have a possibility to define logical expressions with operations of conjunction, disjunction, negation. [two comments, Ru]

(2) It is also desirable to separate slot names from slot fillers by a colon or another punctuation mark, to exclude readings like "rest steps" or "следующий шаги". [one comment, Ru]

(3) It is not evident how to distinguish a precondition from the first step in the generated text, which makes it difficult to judge whether the model is being correctly specified. [one comment, Ru]

3.2.5 Adequacy of CADCAM notions

(1) Choice lists should correspond better to respective notions, thus they should be much shorter, for example, the name of an operating system should not be selected from a list of all proper names. [two comments, Ru]

(2) One evaluator was concerned with the limited size of the existing inventory of domain model notions. She predicted problems, when the vocabulary is of a real size for a problem domain, say, 3000 entries. [one comment, Ru]

(3) It is not possible to express relation "under" (e.g. Under heading). [one comment, Cz]

3.2.6 Robustness of interface operations [Errors, implementation]

(1) Windows sometimes invokes General Protection Fault, and then the AGILE system needs to be loaded from scratch. It would be more convenient to work with autosaving. [one comment, Ru]

(2) Sometimes menu choices cause error messages: 'malformed property list'. [one comment, Ru]

(3) When you try to move the focus of the window during the process of generation (which is very slow) the content of the window disappears and never appears again. Even after the next "Generate" The window with output text displays only the first line. You must re-open the Output window. [one comment, Ru]

(4) The «gui-file» concept, which is required in specifications, is missing in the Domain Model. [two comments, Cz and Ru]

(5) The generation of the texts in three languages is the slowest process of the system. [two comments, Bg and Cz]

3.2.7 Does the generated output express the content you specified?

The remarks from Group A concerned quality of the generated texts with respect to the content of specification models they developed. The remarks are structured according to language.

3.2.7.1 *Bulgarian*

(1) The use of definite articles is not correct in all necessary cases, especially in non-personal style.

(2) Side effects are formulated as active sentences, which is unsatisfactory especially in the non-personal style.

3.2.7.2 *Czech*

(1) Problems with generating of some prepositions, “na souboru” (should be “do souboru” - to the file).

(2) Sometimes discrepancy between GUI names in menu and in the text (dialogové okno vs. dialogový panel).

3.2.7.3 *Russian*

(3) Some logical connections are lost in the output text, for example: “Введите команду endp, чтобы указать начальную точку дуги. Выберите точку.” It is not evident that it is the same point referred in these two sentences.

(4) In the output Russian text (the non-personal style): the phrase “нажимается клавиша” does not look logical, because this buttons are not pressed by themselves, but by somebody.

(5) The generated output is even better than the content presentation. The system improves slang expressions typical for software developers.

3.2.8 Adequacy of the user documentation

The user documentation (this includes User’s Manual, Conceptual Tutorial and Training Manual) covers the system adequately, however, it lacks an explanation of generation options (side-effect implicit, explicit) [one comment, Cz]. It should also contain information about concepts provided by the T-box, including words or collocations, by which concepts are expressed in the target languages [one comment, Ru].

3.2.9 Summary of comments

Table 8 shows a summary of comments to the topics of the questionnaire. In total, evaluators made 69 comments, 47 of them addressed different issues. The evaluators were mostly concerned with differences of the interface features from the standards of MS Windows applications and with problems of creating and modifying a text content.

In general, the rating is positive, in spite of the difficulties the subjects may have experienced. This suggests that the difficulty was within tolerable limits.

Topic of evaluation	Different issues	Total number	Average rate
Intuitiveness of operations with the interface (menus, dialog boxes, etc)	9	13	2
Abilities of the interface for creating and modifying a text content.	22	32	2.44
Abilities of the interface for presentation of a text content.	3	7	2
Intuitiveness of content specification notions (goal, step, precondition, etc)	3	4	1.78
Adequacy of CADCAM notions (click, button, multiline, etc)	3	4	1.67
Robustness of interface operations	5	7	2.33
Adequacy of the user documentation	2	2	1.78
Total number of comments	47	69	

Table 8 Summary of Group A comments

3.3 Group B comments

Comments from Group B evaluators were not as numerous as from Group A. They are structured according to language.

3.3.1 Bulgarian

The comments of the Bulgarian group B evaluators may be classified in the following groups:

(1) Remarks about the meaning and clearness of some sentences:

- no need to use "first" when describing the first step of a procedure; [two comments]
- no clear definition of the necessary steps in the sentence "Repeat these steps to define another element"; [one comment]
- not enough information for some procedural steps (e.g. for the steps "Enter distance", "Enter angle" there are not descriptions of the necessary actions to achieve these goals). [one comment]

In fact evaluators' remarks in this group criticise not only the automatically generated texts, but also the Text 1, extracted from the Autocad User's Manual.

One of the evaluators defends more laconic texts in procedure descriptions and strongly argues against the repetition of the object type together with the object name in subsequent procedure steps. He thinks that after the first sentence, when the name of an object is described with the object type, only the object name has to be used in the next procedural step descriptions (e.g. using "the button OK" only where it is mentioned first and then using only "OK").

(2) Remarks about Bulgarian verbalisation of some notions:

- The action applied on a button may be "press", but not "choose" or "point-to". In fact all these wordings follow not very appropriate trials to verbalise the notion "Click" in Bulgarian. [two comments]

- In the descriptions of side effects a dialogue box has to be "closed", but not "hidden", considering the context of the applied actions. [one comment]

(3) Remarks about some flaws in Bulgarian resources:

- inclusion of articles is not processed properly in all necessary cases, especially in impersonal style; [two comments]
- a morphological feature (vocalisation) is not considered (using another word-form of some prepositions when the next word begins with definite consonant - e.g. "във" instead of "в" before words, beginning with 'v' or 'f'); [one comment]
- not appropriate formulation of the side-effects as active sentences. [one comment]

The problems noted above are general comments about the text quality, no text-specific problems were identified by the Bulgarian evaluators.

3.3.2 Czech

The evaluators were generally satisfied with the text. They were really surprised after having learned (after the evaluation) that the texts were automatically generated.

One evaluator said that it is hard to understand the text without having the application. In general, evaluators had problems with understanding AutoCAD notions (e.g. multiline – multičára). However, they stated this about all texts, including the nongenerated text.

One evaluator said it would be good to use more fonts, e.g. for names of buttons, dialogs, menu items, etc.

Evaluators decreased rating especially for two reasons:

- strange names for user interface elements, terms, etc. (e.g. multičára, element, Zobraz klouby, etc.) – however, they are official Czech AutoCad terms.
- impersonal style (Text 4)

3.3.2.1 Text 1 (real AutoCAD text)

One evaluator was wondering if there is any difference between used term “element” (element) and more common Czech word “prvek” (element).

3.3.2.2 Text 2

In 2.6 and 2.7, there is an instructions to specify a point, one evaluator thinks, it should be explained how (e.g. by mouse?). Real AutoCad manual expresses it in the same manner.

One evaluator did not understand what “úhel úsečky od koncového bodu oblouku” (angle of line-segment from end point of an arc) means.

3.3.2.3 Text 3

Different formatting of two similar sections.

AutoCad expression ‘Zobraz klouby’ (Show joints, 3.7) seemed strange to one evaluator.

3.3.2.4 Text 4 (Impersonal style)

Evaluators did not like impersonal style. However, they meant they prefer using personal style (imperative), not that they do not like our way of impersonal style.

3.3.3 Russian

The Russian Group B evaluators rated the texts as having good grammar, but they noted some infelicities in their structure and word choice, including:

(1) It is not quite clear what format should be used for entering distance and angle. [two comments]

(2) Repeating of labels of dialog boxes, like “Arc mode confirmation” and “Line mode confirmation” seems redundant and makes understanding more difficult. Using pronouns would be better. [two comments]

(3) It is hard to rate an instruction without trying out by oneself and comparing the wording with the result obtained.

3.3.3.1 Text 1 (real AutoCAD text)

There are several comments about the unclear status of the term “element” which should be expressed with a concrete meaning as for example, “line element”. What is meant by an element (for a line, style, etc) should be explicitly defined. Also, when a new element is added, it is better to explicitly say that the element is new. [two comments]

The word “смещение” needs a specification “with reference to what?”. A conjugated arc is probably meant in Step 2, but this is not expressed explicitly. [one comment]

Sequences of steps must be clearly indicated. The user shouldn’t execute Step 1, but Steps 2-5 in Text 1. [two comments]

3.3.3.2 Text 2

The structure of the whole text doesn’t seem fully consistent. For example there are two instructions for drawing a line segment, and they are entirely different (p.1 and p.3). In p.3 it is not quite clear what format should be used for entering distance and angle. [one comment]

3.3.3.3 Text 3

The term “рисунок” occurs for the first and only time at the end of the text and it is not clear which drawing is meant. [one comment]

The expression “чтобы добавить стиль мультилинии в рисунок” is mysterious, because of request to unify objects different by their nature - style and drawing: why should the user add style to it? [two comments]

It is preferable to express “в поле Caps” as “в секции Caps” (*поле* in Russian can contain just one element of data). [one comment]

3.3.3.4 Text 4

The text in the non-personal style does not fit expectations of the evaluators about the nature of instructions as a sequence of steps [three comments]. Sentence 1.13 has two moot points: 1) what is meant by «внутренняя точка», 2) what actions one is to perform to indicate it. [one comment]

3.3.4 Summary

In general, the evaluators were satisfied with the quality of texts presented for evaluation. They had no prior knowledge that some of the texts were generated by a computer. They were positively surprised to learn of the origin of the texts. Text 1 for evaluation was not a text produced by Agile, but extracted from the Autocad User's Manual. However, evaluators raised a lot of concerns about its quality too. The evaluators also noted that it is hard to judge the quality of an instructional text without knowing the application. By this reason, an instructional text should be accompanied with a general introduction into the application.

Table 9 depicts the distribution of rates for the overall text quality of texts produced by AGILE (rate values are explained in Section 2.2, average rate is calculated according to the same mapping as in Section 3.2). The data can be compared to the rates received by the real AutoCAD manual text in different languages (Table 10). The comparison shows that the real AutoCAD text is judged slightly better with respect to organising the information and meeting the standard for inclusion in a high-quality manual, but AGILE generated texts are judged slightly better with respect to wording the information to help the reader understand the instructions. The general impression is that the texts generated by AGILE are of such quality that they could be included in a commercial manual.

Questions	Marks for each site			
As a professional author, how good do you consider the text to be in…	Bg	Cz	Ru	Avg
… organising the information to help the reader execute the instructions?	9A	6A 2B 4C	A 6B 2C	1.67
… wording the information to help the reader understand the instructions?	5A 4B	3A 4B 5C	2A 2B 5C	2
… meeting the standard for inclusion in a high-quality manual?	6A 3B	3A 4B 4C, D	4B 5C	2.07

Table 9 Group B overall text quality rates for texts generated by AGILE

Questions	Marks for each site			
As a professional author, how good do you consider the text to be in…	Bg	Cz	Ru	Avg
… organising the information to help the reader execute the instructions?	3A	3A B	A B C	1.4
… wording the information to help the reader understand the instructions?	A 2B	2B 2C	3B	2.1
… meeting the standard for inclusion in a high-quality manual?	2A B	A 3B	2B C	1.8

Table 10 Group B overall text quality rates for real AutoCAD texts

4. Further Work

Numerous comments received from the evaluators significantly expanded our understanding of user's goals and expectations about a multilingual drafting tool such as AGILE. The evaluation also put a test on the developed grammars of the Bulgarian, Czech and Russian languages. The computational grammars describe a set of possibilities for expressing communicative needs, while experiments by the evaluators provided a testbed of instantiation of these possibilities. In general, the assessment of both interface abilities and text quality by evaluators was positive, however, all the issues mentioned in Sections 3.2 and 3.2.9 should be addressed in the final prototype.

Suggestions for further improvements have been identified by Group A evaluators fall into four groups:

1. extensions of the Domain Model (DM);
2. improvements in the A-box editor;
3. presentation of the content of A-boxes;
4. general robustness

4.1 Extensions of the Domain Model

Evaluation detected several significant restrictions concerning representation of configurations required by texts in the domain.

At the moment, there is no relation between meanings of concepts and meanings of their slots. Thus, in a configuration of ‘style of X’, X can be any concept, not necessarily having a style. Also, any label can be assigned to any object, for example, the name of an operating system is selected from the list of all available labels. Adding a relation between concepts and their slots provides the possibility to prohibit irrelevant configurations and restrict the selection lists, thus, simplifying the user choice.

The current set of DM concepts does not cover all relations between concepts necessary for the Domain Texts. This includes the possibility to express such characteristics of concepts as ‘distance from X to Y’, ‘offset by Z’, ‘under something’, ‘three points’.

Currently, appearance and disappearance of CAD/CAM objects on the screen is expressed by concepts SEE and NOT-SEE with CAD/CAM objects being phenomena of visual perception of the user. This is not felicitous for the user. A better alternative is to represent such side-effects as events. Another infelicity concerns the order of slot names of some concepts: the Location slot is currently expressed prior to the Actee slot, which is the obligatory slot for any action taking an object.

4.2 Extensions of the A-box editor

Evaluation also detected several significant infelicities in the A-box editor interface (their comments on this topic are summarized in Section 3.2.2).

Currently, it is based on mouse operations exclusively. Evaluators suggested that it should be possible to use the keyboard for a variety of operations. For example, it should be possible to specify concept names by typing them and provide also for automatic completion (a feature typical for modern GUI applications). The keyboard could be also used for issuing commands to the system, for example, for generating or saving an A-box. Another suggestion concerns the possibility to search for concept instances in an A-box. This would help both in understanding the instructions represented by an A-box and in proper co-indexation of concepts (the latter is a source for the majority of errors). Another suggestion for decreasing the amount of co-indexation errors is to have a list of already introduced objects, which can be referenced, when the A-box is developed.

The evaluators asked for additional comments to DM concepts, so that the right concept can be selected up without the trial-and-error cycle (it is hard to decide what a concept means from its name only). A comment could be shown as a tooltip, when a concept is pointed at by the mouse, and could be displayed on the status bar when the item has focus. The description and examples could be also shown in a popup window, when the user presses the Help key.

The evaluators also asked for the better design of the insert and replace operations. Inserting a new step into a list of steps requires several Cut/Paste operations for shifting the list of steps to free the space for the new step. The sequence of such operations is error-prone. Another suggestion is to provide the Undo operation.

Building the hierarchical structure of A-boxes is not an easy procedure, for example, the user should choose 'list of methods: first method: steps: first step: action' for adding a step of a submethod. The evaluators asked for automatic filling of such steps to simplify the task.

4.3 Presentation of the content of A-boxes

Some comments from the evaluators concerned presentation of the content of A-boxes. They like the color scheme for visual guidance that helps to distinguish alternative methods for achieving goals. However, they were dissatisfied with LISP-like structures using FIRST/REST slots. They strongly suggest representation of sequences of steps as numbered lists.

They also suggest more compact displaying of the A-box content, because it is hard to fit into the computer screen. This can be achieved by using shortened slot names (combined with the on-line help for hinting their meanings), hiding unfilled optional slots, collapsing and expanding submethods for achieving steps.

The evaluators noted that it is hard to infer the meaning of an A-box piece from its formal structure, while the text generated from the complete A-box does not allow to identify the target piece. They suggested a possibility to generate from a selected piece of the A-box being edited.

4.4 Robustness and compatibility

There is a group of comments about the general robustness of the AGILE interface and its compliance with standards of the Windows environment. The interface is implemented in Common LISP Interface Manager (CLIM), which allows cross-platform compatibility, but lacks some features typical for Windows applications. Some problems identified below cannot be resolved adequately because of internal restrictions of the CLIM implementation in Windows, but these problems should be taken into account during the development of the Final Prototype. This includes several improvements, for example, keeping the focus of operations on the item being edited, displaying a file name in the header of the editor, etc. (the complete set of problems is given in Sections 3.2.1 and 3.2.6 above). The evaluators also suggested to use standard keyboard shortcuts and the indicator, displaying the percentage of task completion, when the system performs a task that requires significant time.

4.5 Text quality improvements

Apart from a few language-dependent problems caused by restrictions in the lexicogrammars, evaluators were mostly concerned with the lack of variation in the evaluated texts. According to the Technical Annex, the intermediate prototype should be capable of generating a single style of instructional text for complex procedures. However, a style alteration has been added even in the intermediate demonstrator for expressing user actions in the personal style or in the non-personal style (see the TEXS2 deliverable).

The grammatical structures used in the non-personal style are language-dependent and are discussed in the SPEC2 deliverable. A text written exclusively in the non-personal style was judged as less felicitous for all target languages, because it less corresponds to the nature of an instruction for the user: A-boxes describe steps that are accomplished to achieve a goal, while the non-personal style describes abilities of the interface objects.

The style of a text varies according to user's goals even within a single text, so that descriptions and side effects are expressed in the indicative mood, lists of actions are rendered in the personal style, while remarks on other possible actions are in the non-personal style. Other types of control over the generated text are advisable, for example, modal expressions, infinitival commands or specification of user's knowledge as naive or expert to omit evident methods in the latter case (for example, "Save the drawing" vs. "Save the drawing by choosing Save in the File menu"), as discussed in the TEXS3 deliverable.

One more style control option concerns the possibility either to express user's goals with actions required to achieve them (*Specify the start point by entering endp*) or to express steps which are performed by the user and are commented with respect to user's goals (*Enter endp to specify the start point*), as also discussed in TEXS3.

Style control options belong to topics scheduled for the final prototype (according to the Technical Annex, p. 25, the final prototype is defined as "The final prototype will enable flexible text structuring and offer stylistic variation in the generation of complex procedures"). So, we find a coincidence of the nearest project goals and features expected by potential users/evaluators.

5. Conclusions

The results of our evaluation shows that, with training, users are able to write documentation for the CAD/CAM domain in their own language with the aid of AGILE and the quality of output texts is sufficiently good for their inclusion into drafts of high-quality manuals. This is true for all three localised versions and for all the subjects tested. However, the system interface proved to be cumbersome to use and to be unintuitive in places. The results have highlighted some important defects to be rectified in the final prototype. As is usually the case with user-interface, some of these involve small changes to the interface but have a large impact on the user. They are identified in Section 3.3.3.1 above.

Evaluators rated the abilities of the Intermediate Prototype as good, with the exception of the knowledge editing abilities and general robustness of the interface, which were rated slightly below good (in average). Detailed analysis of A-boxes produced by evaluators shows that the most of them have correct structure, but some of them lack proper co-indexation of concept instances.

Sample output texts of AGILE were judged by natural language experts to be of comparable quality to similar texts found in good commercial manuals for the three languages. Evaluators had no knowledge that the evaluated texts were not produced by a human author and were positively surprised to learn that they were computer-generated. The quality of the generated texts was judged as good both in terms of the organisation of information to help the reader and with respect to the possibility of their inclusion into a draft of a high-quality manual.

References

AutoCAD (1995) *AutoCAD Release 13. User's guide,* chapter 2. Autodesk Co.

Halliday, M.A.K. (1978) *Language as social semiotic: the social interpretation of language and meaning.* London: Edward Arnold.

Halliday, M.A.K. (1985) *Introduction to Functional Grammar,* London: Edward Arnold.

Power, R. and Scott, D. (1997) Evaluation of the DRAFTER system, WP5/WP10/WP11 Deliverable, Drafter Project EPSRC J19221, Information Technology Research Institute.

Sparck-Jones, K. and Galliers, J.R. (1996) *Evaluating natural language processing systems.* Berlin: Springer-Verlag.

AGILE deliverables referred in the present deliverable:

[CORP] Anthony Hartley, Danail Dochev, Nevena Gromova, Kamenka Staykova, Alla Bemova, Aleksandr Rosen, Jiri Trojanek, Elena Sokolova. *Tagging and analysis of instructional texts in the software domain.* June 1998. The deliverable for the WP3 of AGILE project PL961104.

[LSPEC2] Geert-Jan M. Kruijff, John Bateman, Alla Bémová, Danail Dochev, Ivana Kruijff-Korbayová, Serge Sharoff, Hana Skoumalová, Lena Sokolova, Kamena Stoikova, Elke Teich, Jirí Trojánek. *Modelling Lexical Resources in KPML for Generating Instructions in Slavonic Languages.* October 1998. The deliverable for the WP4-2 of AGILE project PL961104.

[MODL1] Richard Power, *Preliminary model of the CAD/CAM domain.* June 1998. The deliverable for the WP2-1 of AGILE project PL961104.

[SPEC2] Elena Andonova, John Bateman, Nevena Gromova, Anthony Hartley, Geert-Jan M. Kruijff, Ivana Kruijff-Korbayová, Serge Sharoff, Hana Skoumalová, Lena Sokolova, Kamenka Staykova, Elke Teich, *Formal specification of extended grammar models.* February 1999. The deliverable for the WP6-2 of AGILE project PL961104.

[TEXS2] I. Kruijff-Korbayová, G.J.M. Kruijff, J. Bateman, D. Dochev, N. Gromova, A. Hartley, E. Teich, S. Sharoff, L. Sokolova, and K. Staykova, *Specification of elaborated text structures.* AGILE deliverable 5.2, April 1999. (Deliverable comprises TEXS2-Cz, TEXS2-Bu, TEXS2-Ru)

6. Appendices

1. The Group A evaluation scenario.
2. The design of Group A questionnaire sheets.
3. Specifications of target texts for Group A evaluators.
4. The design of Group B questionnaire sheets.
5. Target texts for Group B evaluators

Appendix 1: The Group A evaluation scenario

The following text was received by Group A evaluators.

Precondition

Approaching this stage assumes that you have been successfully trained with the AGILE interface to complete tasks specified below without any help from the AGILE development team. We solicit your judgements and free-text comments on capabilities of the AGILE interface according to the questionnaire below.

While you are accomplishing the tasks, keep the “think-aloud” protocol mode by writing short remarks for anything that is worth noting. This will help you to fill the questionnaire afterwards. Please, also note the time you spent for completion of each of four tasks.

Task 1: Load and edit two simple text specification models

The directory \Agile\Iface\A-boxes\Task1\ contains two text specification models (also called A-boxes) that have been developed by another member of your team. You have 20 minutes for completion of this task.

1. Load the file “Test1.abx”.
2. Understand the content specified by the model.
3. Add the following step standing for the precondition of the sequence of steps under the topmost goal: “Click the Save icon from the Draw toolbar”.
4. Add one more step, as the last step for achieving the topmost goal: “Click the Save button”.
5. Generate the text and check how it matches your intentions. If there are discrepancies, edit the model and generate the text again to make sure that the text conveys your intended meanings.
6. Save the model as “Test1-<your-name>.abx”.
7. Load the file “Test2.abx”.
8. Understand the content specified by the model. Tip: you may generate its text, if you have troubles.
9. Add the following side effect “The *Element Properties* dialogue box appears.” as the result of the following step “Click the *Element Properties* button.”
10. Add one more step, as the second step for achieving the topmost goal: “Enter the offset of the elements in the *Element Properties* dialogue box”.
11. Generate the text and check how it matches your intentions. If there are discrepancies, edit the model and generate the text again to make sure that the text conveys your intended meanings.

12. Save the model as “Test2-<your-name>.abx”.

Task 2: Create two new text specification models

You have 40 minutes for completion of this task.

1. Create a new model according to the specification below. Tip: if you want to fit more information from the model on your screen, you may downscale the font size.

> **Draw a line-arc-polyline**
>
> First, start the *PLINE* command.
>
> Draw a line segment.
>
> Switch to the *Arc* mode.
>
> Enter the *a* command line.
>
> The *Arc mode confirmation* dialog box appears.
>
> Draw an arc.
>
> Switch to the *Line* mode.
>
> Enter the command line *l*.
>
> The *Line mode confirmation* dialog box appears.
>
> Draw a line segment.
>
> End the polyline.
>
> Press the *Return* key.

2. Save the model as “Test1-<your-name>.abx” in the directory \Agile\Iface\A-boxes\Task2\.
3. Generate the text and check how it matches your intentions. If there are discrepancies, edit the model and generate the text again to make sure that the text conveys your intended meanings.
4. Create a new model.

[The model is site-specific. S-5 for BAS, S-4 for CU, S-3 for RRIAI, see Appendix 3]

5. Save the model as “Testx-<your-name>.abx” in the directory \Agile\Iface\A-boxes\Task2\.
6. Generate the text and check how it matches your intentions. If there are discrepancies, edit the model and generate the text again to make sure that the text conveys your intended meanings.

Task 3: Create two new bigger text specification models

You have 2 hours for completion of this task.

1. Create a new model according to the specification below. The model in \Agile\Iface\A-boxes\Task2\Test1.abx may be used for this task.

Draw a line-arc-polyline

First start the *Pline* command.

Using the *Windows* operating system: choose the *Polyline* option from the *Polyline flyout* on the *Draw* toolbar.

Using the *DOS* or *UNIX* operating system: choose the *Polyline* option from the *Draw* menu.

Draw a line segment.

Choose the start point of the line segment.

Choose the endpoint of the line segment.

Draw an arc.

First switch to the *Arc* mode.

Enter the command line *a*. The *Arc mode confirmation* dialog box appears.

Click the *OK* button in the *Arc mode confirmation* dialog box. The *Arc mode confirmation* dialog box disappears.

Choose the endpoint of the arc.

Draw a line segment.

First switch to the *Line* mode.

Enter the command line *l*. The *Line mode confirmation* dialog box appears.

Click the *OK* button in the *Line mode confirmation* dialog box. The *Line mode confirmation* dialog box appears.

Enter the distance.

2. Save it to the file “Test1-<your-name>.abx” in the directory \Agile\Iface\A-boxes\Task3\.

3. Generate the text and check how it matches your intentions. If there are discrepancies, edit the model and generate the text again to make sure that the text conveys your intended meanings.

4. Create a new model according to the specification below. The model in \Agile\Iface\A-boxes\Task2\Testx-<your-name>.abx may be used for this task.

[The model is site-specific. B-5 for BAS, B-4 for CU, B-3 for RRIAI. , see Appendix 3]

5. Save it to the file “Testx-<your-name>.abx” in the directory \Agile\Iface\A-boxes\Task3\.

6. Generate the text and check how it matches your intentions. If there are discrepancies, edit the model and generate the text again to make sure that the text conveys your intended meanings.

7. The file “Testx-<your-name>.abx” containing the model you created is sent by e-mail to respective Eastern European partners.

Task 4: Load and edit two bigger text specification models

You have 30 minutes for completion of this task.

1. Receive two models from respective Eastern European partners. Save them in the directory \Agile\Iface\A-boxes\Task4\.
2. Load the file “Testx.abx”.
3. Understand the content specified by the model.
4. Generate the text into your mother tongue and check how it matches the content specified by the model. If there are discrepancies, edit the model and generate the text again to make sure that the text conveys your intended meanings.
5. Alter the model according to the specification below.

[changes are site-specific, see Appendix 3]

6. Load the file “Testx.abx”.
7. Understand the content specified by the model.
8. Generate the text into your mother tongue and check how it matches the content specified by the model. If there are discrepancies, edit the model and generate the text again to make sure that the text conveys your intended meanings.
9. Alter the model according to the specification below.

[CHANGES ARE SITE-SPECIFIC, SEE APPENDIX 3]

Now, you are ready for filling the questionnaire.

Appendix 2: The design of Group A questionnaire sheets.

Data on the evaluator of the AGILE interface.

0. **Name and affiliation:**

1. **Expertise with graphical user interfaces:**

2. **Expertise in software development:**

Computer configuration for evaluation (processor and memory):

Time spent on accomplishing:

Task1

Task2

Task3

Task4

How felicitous is the AGILE interface for specification of a text content?

Use the following scale of scores for your judgements:

Score	Explanation
A	Excellent
B	Good
C	Poor
D	Useless

Score the AGILE interface capabilities by circling the letter that corresponds to your judgment,
for example B

Ref	Question	Score			
1	Intuitiveness of operations with the interface (menus, dialog boxes, etc)	**A**	**B**	**C**	**D**
2	Abilities of the interface for creating and modifying a text content.	**A**	**B**	**C**	**D**
3	Abilities of the interface for presentation of a text content.	**A**	**B**	**C**	**D**
4	Intuitiveness of content specification notions (goal, step, precondition, etc)	**A**	**B**	**C**	**D**
5	Intuitiveness of CADCAM notions (click, button, multiline, etc)	**A**	**B**	**C**	**D**
6	Robustness of interface operations	**A**	**B**	**C**	**D**
7	Does the generated output expresses the content you specified?	**A**	**B**	**C**	**D**
8	Adequacy of the user documentation	**A**	**B**	**C**	**D**

Please, add separate comments supporting your scores. Use as much extra sheets as you need.

Appendix 3: Specifications of target texts for Group A evaluators

Task 1: Load and edit two simple A-boxes

The total maximal duration of the task: 20 min.

The directory \Agile\Iface\A-boxes\Task1\ contains two A-boxes. Their text specifications are not presented to the subjects.

1. “Test1.abx”.

To save a drawing

Enter the data-name of the drawing in the Name field.

2. “Test2.abx”.

Create a style of a multiline

First open-screen-object the *Multiline Styles* dialog box

Click the *Element Properties* button.

Click the *OK* button in the *Multiline Styles* dialogue box. The *Multiline Styles* dialogue box disappears.

Task 2: Create two new A-boxes

The total maximal duration of the task: 40 min. The subjects read text specifications.

S-2

Draw a line-arc-polyline

First, start the *PLINE* command.

Draw a line segment.

Switch to the *Arc* mode.

Enter the *a* command line.

The *Arc mode confirmation* dialog box appears.

Draw an arc segment.

Switch to the *Line* mode.

Enter the command line *l*.

The *Line mode confirmation* dialog box appears.

Draw a line segment.

End the polyline.

Press the *Return* key.

S-3 (RRIAI-specific)

Draw an arc

- First start the *ARC* command.
- Specify the start point of the arc.
- Specify the second point of the arc.
- Specify the endpoint of the arc.
- End the arc.
  - Press the *Return* key.
- Save the drawing.
  - Choose the *Save* option from the *File* menu. The *Save* dialogue box appears.
  - Enter a name of the drawing in the *Name* field.
  - Click the *Save* button. The *Save* dialogue box disappears.

S-4 (CU-specific)

Define a style of a multiline in a drawing

- First, choose the *Multiline Style* option from the *Data* menu. The *Multiline Styles* dialog box appears.
- Specify the properties of the multiline.
  - Click the *Multiline Properties* button. The *Multiline Properties* dialog box appears.
  - Click the *OK* button in the *Multiline Properties* dialog box. The *Multiline Properties* dialog box disappears.
- Save the style into a file.

S-5 (BAS-specific)

Define a boundary set in a complex drawing

- First open the screen object *Boundary Hatch* dialog box.
- Specify the properties of the boundary set.
  - Choose the *Advanced* option in the *Boundary* heading.
  - Click the *Make New Boundary Set* button in the *Advanced Options* dialog box.
  - Define the boundary set.
  - Click the *OK* button in the *Advanced Options* dialog box,. The *Advanced Options* dialog box disappears.
- Click the *OK* button. The *Boundary Hatch* dialogue box disappears.

Task 3: Create two new bigger A-boxes

The total maximal duration of the task: 2 hours. The subjects read text specifications.

B-2: common for all sites.

Draw a line-arc-polyline

First start the *Pline* command.

Using the *Windows* operating system: choose the *Polyline* option from the *Polyline flyout* on the *Draw* toolbar.

Using the *DOS* or *UNIX* operating system: choose the *Polyline* option from the *Draw* menu.

Draw a line segment.

Specify the start point of the line segment.

Specify the endpoint of the line segment.

Draw an arc [segment].

First switch to the *Arc* mode.

Enter the command line *a*. The *Arc mode confirmation* dialog box appears.

Choose the *OK* button in the *Arc mode confirmation* dialog box. The *Arc mode confirmation* dialog box disappears.

Specify the endpoint of the arc.

Draw a line segment.

First switch to the *Line* mode.

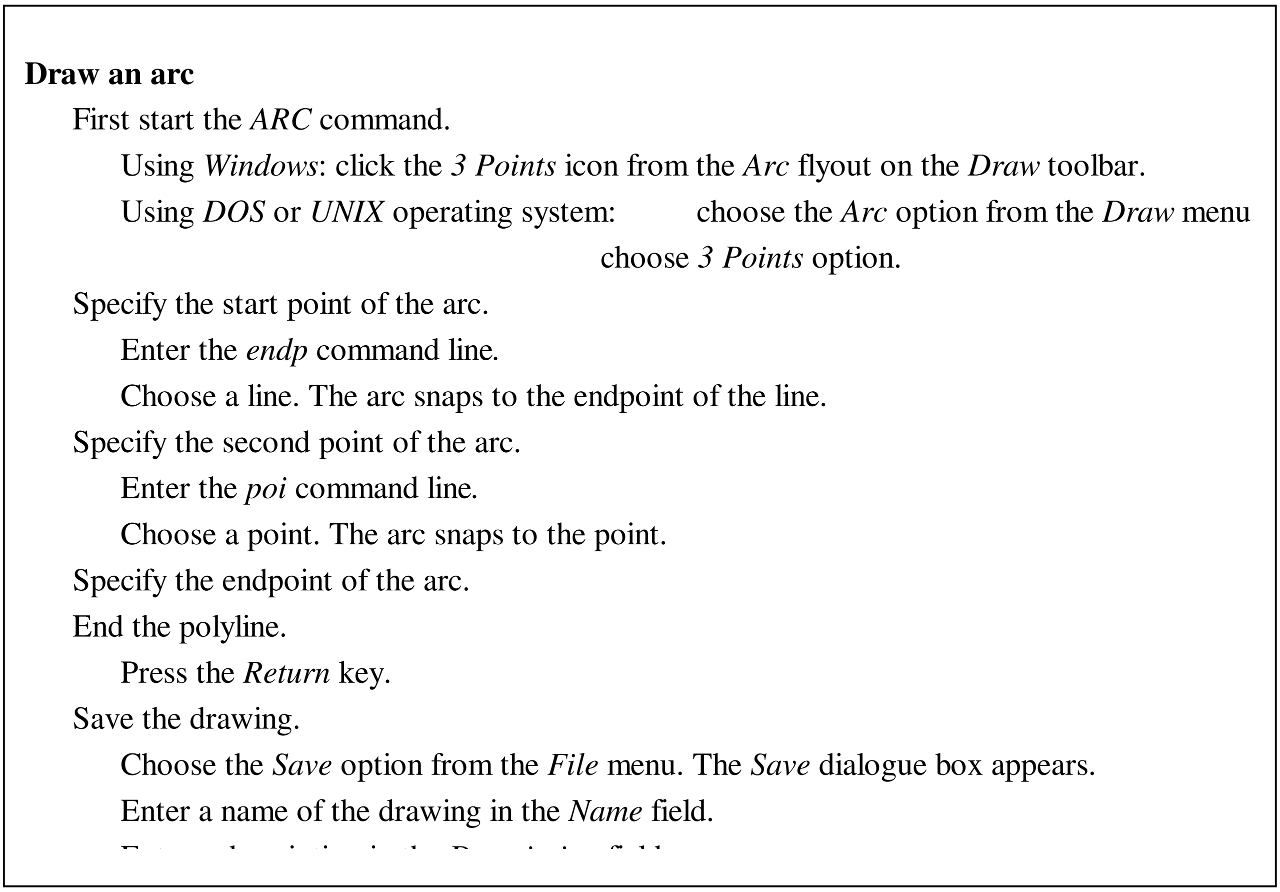
Enter the command line *l*. The *Line mode confirmation* dialog box appears.

Choose the *OK* button in the *Line mode confirmation* dialog box. The *Line mode confirmation* dialog box appears.

Enter the distance from the endpoint of the arc to the line.

Enter the angle from the endpoint of the arc to the line.

B-3 (RRIAI-specific)



Draw an arc

First start the *ARC* command.

Using *Windows*: click the *3 Points* icon from the *Arc* flyout on the *Draw* toolbar.

Using *DOS* or *UNIX* operating system: choose the *Arc* option from the *Draw* menu
choose *3 Points* option.

Specify the start point of the arc.

Enter the *endp* command line.

Choose a line. The arc snaps to the endpoint of the line.

Specify the second point of the arc.

Enter the *poi* command line.

Choose a point. The arc snaps to the point.

Specify the endpoint of the arc.

End the polyline.

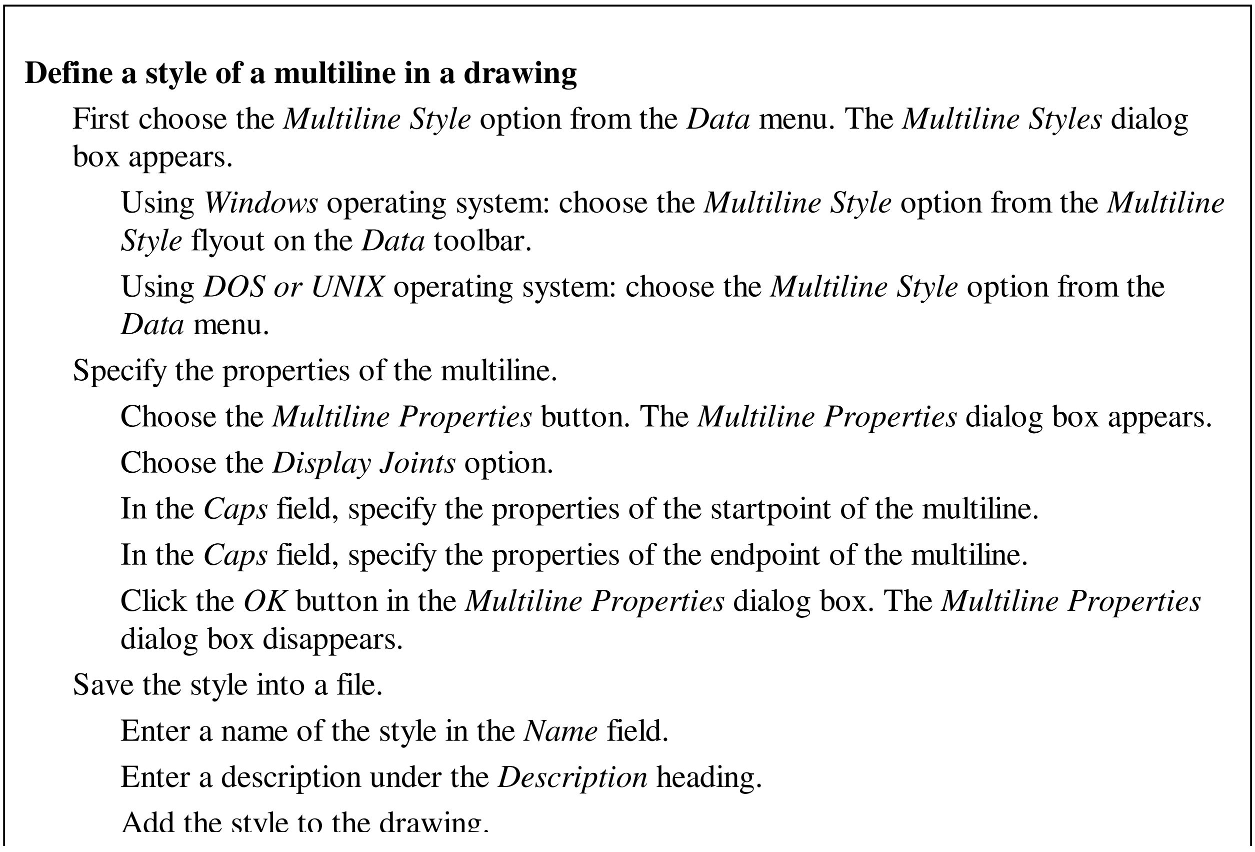
Press the *Return* key.

Save the drawing.

Choose the *Save* option from the *File* menu. The *Save* dialogue box appears.

Enter a name of the drawing in the *Name* field.

B-4 (CU-specific)



Define a style of a multiline in a drawing

First choose the *Multiline Style* option from the *Data* menu. The *Multiline Styles* dialog box appears.

Using *Windows* operating system: choose the *Multiline Style* option from the *Multiline Style* flyout on the *Data* toolbar.

Using *DOS or UNIX* operating system: choose the *Multiline Style* option from the *Data* menu.

Specify the properties of the multiline.

Choose the *Multiline Properties* button. The *Multiline Properties* dialog box appears.

Choose the *Display Joints* option.

In the *Caps* field, specify the properties of the startpoint of the multiline.

In the *Caps* field, specify the properties of the endpoint of the multiline.

Click the *OK* button in the *Multiline Properties* dialog box. The *Multiline Properties* dialog box disappears.

Save the style into a file.

Enter a name of the style in the *Name* field.

Enter a description under the *Description* heading.

Add the style to the drawing.

B-5 (BAS-specific)

Define a boundary set in a complex drawing

First open the screen object *Boundary Hatch* dialog box.

Using *Windows* operating system: choose the *Hatch* option from the *Hatch* flyout on the *Draw* toolbar.

Using *DOS or UNIX* operating system: choose the *Hatch* option from the *Draw* menu.

Specify the properties of the boundary set.

Under the *Boundary* heading choose the *Advanced* option. The *Advanced Options* dialog box appears.

In the *Advanced Options* dialog box, choose the *Make New Boundary Set* button.

Define the boundary set.

Specify the corner points at the *Select Objects* prompt.

Press the *Return* key.

In the *Advanced Options* dialog box, choose the *OK* button. The *Advanced Options* dialog box disappears.

In the *Boundary Hatch* dialog box, choose the *Pick Points* option.

Produced site-specific A-boxes are sent to:

BAS: dochev@iinf.bas.bg

CU: korbay@fairway.ms.mff.cuni.cz

RRIAI: sharoff@aha.ru

Task 4: Load and edit two bigger A-boxes

The total maximal duration of the task: 30 minutes. The subjects correct textual specifications received from other sites.

For Text B-3 (received by BAS and CU from RRIAI):

A/ Insert steps in order to describe the actions necessary to specify the endpoint of the arc (similarly to the specification of the actions for specifying the second point).

B/ Edit the actions described as constraints of the procedure in order to specify the same actions under the constraint ‘Windows’ as the actions, described under the constraint ‘Dos or Unix’.

C/ Generate the text into your mother tongue.

For Text B-4 (received by BAS and RRIAI from CU)

A/ Change the description in order to specify justification of the multiline between the descriptions of the steps for specifying the properties of the multiline start and end points.

B/ Edit the actions described as constraints of the procedure in order to specify the same actions under the constraint ‘Dos or Unix’ as the actions, described under the constraint ‘Windows’.

C/ Generate the text into your mother tongue.

For Text B-5 (received by CU and RRIAI from BAS)

A/ After the text for choosing the ‘Apply’ option insert steps in order to describe the actions necessary to save the boundary set in a file by clicking the button ‘Save’ in the ‘Boundary hatch’ dialogue box.

B/ Edit the actions described as constraints of the procedure in order to specify the same actions under the constraint ‘Windows’ as the actions, described under the constraint ‘Dos or Unix’.

C/ Generate the text into your mother tongue.

Appendix 4: The design of Group B questionnaire sheets

Series A: evaluating the text as a whole

Text

Use the following scale:

Score	Explanation
A	Excellent
B	Good
C	Poor
D	Useless

Score the text as a whole by circling the letter that corresponds to your judgment, for example B

To create a multiline style

First open the Multiline Styles dialog box using one of these methods:

Windows: From the Object Properties toolbar or the Data menu, choose Multiline Style.

DOS and UNIX: From the Data menu, choose Multiline Style.

1. Choose Element Properties to add elements to the style.
2. In the Element Properties dialog box, enter the offset of the multiline element.
3. Select Add to add the element.
4. Choose Color.
   The Select Color dialog box appears. Select the element's color.
5. Choose Linetype.
   The Select Linetype dialog box appears. Select the element's linetype.
6. Repeat these steps to define another element.
7. Choose OK to save the style of the multiline element and to exit the Element Properties dialog box.

Give your rating.

As a professional author, how good do you consider the text to be in…	**Score**
… organising the information to help the reader execute the instructions?	**A B C D**
… wording the information to help the reader understand the instructions?	**A B C D**
… meeting the standard for inclusion in a high-quality manual?	**A B C D**

If you wish to make any further comments, please add them here:

-Break page here, presenting each [scale + text + questions] separately.-

Series B: evaluating individual parts of the text

Text

Use the following scale:

Score	Explanation
A	The text is entirely satiscfactory.
B	Minor revision is desirable.
C	Minor revision is necessary.
D	Major revision is necessary.

Score the individual fragments by circling the letter that corresponds to your judgment, for example B

Ref	Text Fragment	Score			
1.1	**To create a multiline style**	**A**	**B**	**C**	**D**
1.2	First open the Multiline Styles dialog box using one of these methods:	**A**	**B**	**C**	**D**
1.3	**Windows**: From the Object Properties toolbar or the Data menu, choose Multiline Style.	**A**	**B**	**C**	**D**
1.4	**DOS and UNIX**: From the Data menu, choose Multiline Style.	**A**	**B**	**C**	**D**
1.5	1. Choose Element Properties to add elements to the style.	**A**	**B**	**C**	**D**
1.6	2. In the Element Properties dialog box, enter the offset of the multiline element.	**A**	**B**	**C**	**D**
1.7	3. Select Add to add the element.	**A**	**B**	**C**	**D**
1.8	4. Choose Color..	**A**	**B**	**C**	**D**
1.9	The Select Color dialog box appears.	**A**	**B**	**C**	**D**
1.10	Select the element's color	**A**	**B**	**C**	**D**
1.11	5. Choose Linetype..	**A**	**B**	**C**	**D**
1.12	The Select Linetype dialog box appears.	**A**	**B**	**C**	**D**
1.13	Select the element's linetype	**A**	**B**	**C**	**D**
1.14	6. Repeat these steps to define another element.	**A**	**B**	**C**	**D**
1.15	7. Choose OK to save the style of the multiline element	**A**	**B**	**C**	**D**
1.16	and to exit the Element Properties dialog box.	**A**	**B**	**C**	**D**

If you wish to make any further comments, please add them here:

-Break page here, presenting each [scale + text + questions] separately.-

Appendix 5: Target texts for Group B evaluators

Bulgarian

Text 1

Създаване стил на мултилиния

1.Първо отворете диалоговия прозорец *Multiline Styles*, като използвате един от следните методи:

Windows: Изберете опцията *Multiline Style* от функционалния ред *Object Properties* или от

менюто *Data*.

DOS / UNIX: Изберете опцията *Multiline Style* от менюто *Data*.

2.Изберете опцията *Element Properties*, за да прибавите елементи към стила.

Въведете отместването на мултилинията в диалоговия прозорец *Element Properties*.

Изберете опцията *Add*, за да прибавите елемент.

3.Изберете цвят.

Появява се диалоговия прозорец *Select Color*.

Изберете цвят на елемента.

4.Изберете тип на линията.

Появява се диалоговия прозорец *Select Linetype*.

Изберете типа на линията на елемента.

5.Повторете тези стъпки, за да дефинирате друг елемент.

6.Изберете *OK*, за да запазите стила на мултилинията и да скриете диалоговия прозорец *Element Properties*.

Text 2

Чертане на полилиния от отсечки и дъги

1. Стартирайте командата *PLINE*.

Windows: Изберете опцията *Polyline* от плаващото меню *Polyline* в функционалния ред

Draw.

DOS / UNIX: Изберете опцията *Polyline* от менюто *Draw*.

2. Начертайте отсечка.

Задайте начална точка на отсечка.

Задайте крайна точка на отсечка.

3. Начертайте дъга.

Включете режима *Arc*.

Въведете командата *a*.

Виждате диалоговия прозорец *Arc Mode Confirmation*.

Натиснете бутона *OK* в диалоговия прозорец *Arc Mode Confirmation*.

Скривате диалоговия прозорец *Arc Mode Confirmation*.

Задайте крайна точка на дъга.

4. Начертайте отсечка.

Включете режима *Line*.

Въведете командата *l*.

Виждате диалоговия прозорец *Line Mode Confirmation*.

Натиснете бутона *OK*.

Скривате диалоговия прозорец *Line Mode Confirmation*.

Въведете разстояние.

Въведете ъгъл.

5. Натиснете клавиша *Return*, за да завършите полилинията от отсечки и дъги.

Text 3

Дефиниране на стил на мултилиния в чертеж

1.Изберете опцията *Multiline Styles* от менюто *Data*.

Windows: Изберете опцията *Multiline Styles* от плаващото меню *Multiline Styles* в

функционалния ред *Data*.

DOS / UNIX: Изберете опцията *Multiline Styles* от менюто *Data*.

Виждате диалоговия прозорец *Multiline Styles*.

2.Въведете характеристики на мултилиния в диалоговия прозорец *Multiline Styles*.

Изберете опцията *Multiline Properties* от диалоговия прозорец *Multiline Styles*.

Виждате диалоговия прозорец *Multiline Properties*.

Изберете опцията *Display Joints* от диалогов прозорец *Multiline Properties*.

Въведете характеристики на началната точка на мултилинията в полето *Caps*.

Въведете характеристики на крайната точка на мултилинията в полето *Caps*.

Посочете бутона *OK* в диалоговия прозорец *Multiline Properties*.

Скривате диалоговия прозорец *Multiline Properties*.

3.Запазете стила на мултилинията в файл.

Въведете име на стила на мултилинията в полето *Name*.

Въведете описание на стила на мултилинията в полето *Description*.

Изберете бутона *Add* от диалоговия прозорец *Multiline Styles*, за да добавите стил на

мултилинията към чертежа.

Изберете бутона *Save* от диалоговия прозорец *Multiline Styles*, за да запазите стила на мултилинията във файл.

4.Посочете бутона *OK* в диалоговия прозорец *Multiline Styles*.

Скривате диалоговия прозорец *Multiline Styles*.

Text 4

Дефиниране на област за щриховане в сложен чертеж

1.Отваря се диалоговият прозорец *Boundary Hatch*.

Windows: Избира се опция *Hatch* от плаващото меню *Hatch* в функционалния ред *Draw*.

DOS / UNIX: Избира се опция *Hatch* от менюто *Draw*.

2.Задават се характеристики на област за щриховане.

Избира се опция *Advanced* от заглавното поле *Boundary*.

Виждате диалоговия прозорец *Advanced Options*.

Избира се бутон *Make New Boundary Set* от диалоговия прозорец *Advanced Options*.

Задават се ъглови точки в командното поле *Select Objects*, за да се дефинира област за

щриховане.

Посочва се бутон *OK* в диалоговия прозорец *Advanced Options*.

Скривате диалоговия прозорец *Advanced Options*.

Избира се опция *Pick Points* от диалоговия прозорец *Boundary Hatch*.

Задава се вътрешна точка.

Натиска се клавиш *Return*.

Избира се опция *Apply* от диалоговия прозорец *Boundary Hatch*, за да се направи щриховка.

3.Скривате диалоговия прозорец *Boundary Hatch*.

Czech

Text 1

Vytvoření stylu multičáry

Nejdříve otevřete dialogový panel *Styly multičár* jednou z následujících metod:

Windows: Z nástrojového panelu Vlastnosti objektů nebo z menu Data vyberte *Styl mutičáry.*

DOS a UNIX: Z menu Data vyberte *Styl multičáry.*

1. Vyberte *Vlastnosti prvků* pro přidání elementů ke stylu.
2. V dialogovém panelu Vlastnosti prvků zadejte rozměr posunutí multičáry.
3. Vyberte *Přidat* pro přidání elementu.
4. Zvote element.
5. Vyberte *Barva*. Poté zvolte barvu elementu z dialogového panelu *Výběr barvy*.
6. Vyberte *Typ čáry*. Poté zvolte typ čáry daného elementu z dialogového panelu *Výběr typů čar*.
7. Pro vytvoření dalšího elementu tyto kroky opakujte.
8. Vyberte *OK* pro uložení vlastností elementu multičáry a opuštění dialogového panelu *Vlastnosti multičáry.*

Text 2

Kreslení křivky z úseček a oblouků

Spusťte příkaz Křivka.

Windows Vyberte volbu Křivka z plovoucí ikonové nabídky Křivka v nástrojovém panelu Kresli.

DOS a Unix Vyberte volbu Křivka z nabídky Kresli.

1. Nakreslete rovný segment.
    1. Určete počáteční bod rovného segmentu.
    2. Určete koncový bod rovného segmentu.
2. Nakreslete oblouk.
    1. Přepněte režim Oblouk.
        1. Zadejte řetězec o na příkazovém řádku.
           Objeví se dialogový panel Potvrzení režimu kreslení oblouků.
        2. Klikněte na tlačítko OK v dialogovém panelu Potvrzení režimu kreslení oblouků.
           Dialogový panel Potvrzení režimu kreslení oblouků zmizí.
    2. Určete koncový bod oblouku.

3. Nakreslete rovný segment.

    Přepněte režim Úsečka.

    1. Zadejte řetězec e na příkazovém řádku.
       Objeví se dialogový panel Potvrzení režimu kreslení úseček.
    2. Klikněte na tlačítko OK v dialogovém panelu Potvrzení režimu kreslení úseček.
       Dialogový panel Potvrzení režimu kreslení úseček zmizí.

    1. Zadejte vzdálenost úsečky od koncového bodu oblouku.
    2. Zadejte úhel úsečky od koncového bodu oblouku.

4. Stiskněte klávesu Enter pro ukončení křivky z úseček a oblouků.

Text 3

(B4, personal, implicit s.e.)

Definování stylu multičáry

Otevřete dialogový panel Styly multičár.

Windows Vyberte volbu Styly multičár z plovoucí ikonové nabídky Styly multičár v nástrojovém panelu Data.

DOS a Unix Vyberte volbu Styly multičár z nabídky Data.

1. Zadejte vlastnosti multičáry.
    1. Vyberte volbu Vlastnosti multičár z dialogového panelu Styly multičár.
    2. Vyberte volbu Zobraz klouby v dialogovém panelu Vlastnosti multičár.
    3. Zadejte vlastnosti počátečního bodu multičáry v poli Zakončení.
    4. Zadejte vlastnosti koncového bodu multičáry v poli Zakončení.
    5. Klikněte na tlačítko OK v dialogovém panelu Vlastnosti multičár.
2. Styl multičáry uložte v souboru.
    1. Zadejte název stylu multičáry v poli Jméno.
    2. Zadejte popis stylu multičáry v poli Popis.
    3. Klikněte na tlačítko Přidat pro přidání stylu multičáry k obrázku.
    4. Klikněte na tlačítko Uložit pro uložení stylu multičáry v souboru.
3. Klikněte na tlačítko OK v dialogovém panelu Styly multičár.

Text 4

(B5, impersonal explicit s.e.)

Definování hraniční množiny na komplexním výkrese

Otevře se dialogový panel Hraniční šrafování.

Windows Vybere se volba Šrafy z plovoucí ikonové nabídky Šrafy v nástrojovém panelu Kresli.

DOS a Unix Vybere se volba Šrafy z nabídky Kresli.

1. Zadejte vlastnosti multičáry.
    1. Vybere se volba Pokročilé z nadpisu Hranice šrafování.
       Objeví se dialogový panel Pokročilé možnosti.
    2. Vybere se tlačítko Tvořit novou hraniční množinu z dialogového panelu Pokročilé možnosti.
    3. Definuje se hraniční množina.
        1. Určí se rohové body na výzvě Výběr objektů.
        2. Stiskne se klávesa Enter.
    4. Klikne se tlačítko OK v dialogovém panelu Pokročilé možnosti.
       Dialogový panel Pokročilé možnosti zmizí.
    5. Vybere se volba Výběr objektů z dialogového panelu Hraniční šrafování.
    6. Určí se vnitřní bod.
    7. Stiskne se klávesa Enter.
2. Vybere se volba Aplikovat z dialogového panelu Hraniční šrafování pro použití šrafování.
   Dialogový panel Hraniční šrafování zmizí.

Russian

Text 1

Создание стиля мультилинии

Сначала откройте диалоговое окно Multiline Styles:

Windows В панели инструментов Object Properties или в меню Data выберите пункт Multiline Style.

DOS / UNIX В меню Data выберите пункт Multiline Style.

1 Нажмите кнопку Element Properties, чтобы добавить элементы в стиль.

2 В диалоговом окне Element Properties введите смещение первого элемента линии.

3 Нажмите кнопку Add, чтобы добавить этот элемент.

4 Выберите пункт Color. Затем выберите цвет элемента в диалоговом окне Select Color.

5 Выберите пункт Linetype. Затем выберите тип линии элемента в диалоговом окне Select Linetype.

6 Повторите эти шаги, чтобы задать еще один элемент.

7 Нажмите кнопку OK, чтобы сохранить стиль элементов мультилинии и закрыть диалоговое окно Element Properties.

Text 2

Рисование полилинии из прямых и дуг

Запустите команду PLINE.

Windows В палитре Polyline на панели инструментов Draw выберите пункт Polyline.

DOS / Unix В меню Draw выберите пункт Polyline

1. Нарисуйте отрезок.

Укажите начальную точку отрезка.

Укажите конечную точку отрезка.

2. Нарисуйте дугу.

Включите режим Arc.

В командной строке введите команду a.
На экране появится диалоговое окно Arc mode confirmation.

В диалоговом окне Arc mode confirmation нажмите кнопку OK.
Диалоговое окно Arc mode confirmation исчезнет с экрана.

Укажите конечную точку дуги.

3. Нарисуйте отрезок.

Включите режим Line.

В командной строке введите команду l.
На экране появится диалоговое окно Line mode confirmation.

В диалоговом окне Line mode confirmation нажмите кнопку OK.
Диалоговое окно Line mode confirmation исчезнет с экрана.

В командной строке введите расстояние.

В командной строке введите угол.
4. Нажмите клавишу Return, чтобы завершить полилинию из прямых и дуг.

Text 3

Сохранение стиля мультилинии

Откройте диалоговое окно Multiline Styles .

Windows

В палитре Multiline Styles на панели инструментов Data выберите пункт Multiline

Styles .

DOS / Unix

В меню Data выберите пункт Multiline Styles .

На экране появится диалоговое окно Multiline Styles .

Введите свойства мультилинии.

В диалоговом окне Multiline Styles выберите пункт Multiline Properties .
На экране появится диалоговое окно Multiline Properties .

В диалоговом окне Multiline Properties выберите пункт Display joints .

В поле Caps введите свойства начальной точки мультилинии.

В поле Caps введите свойства конечной точки мультилинии.

В диалоговом окне Multiline Properties нажмите кнопку OK.
Диалоговое окно Multiline Properties исчезнет с экрана.

Сохраните стиль мультилинии в файле.

В поле Name введите имя стиля мультилинии.

В поле Description введите описание стиля мультилинии.

Нажмите кнопку Add , чтобы добавить стиль мультилинии в рисунок.

Нажмите кнопку Save , чтобы сохранить стиль мультилинии в файле.

В диалоговом окне Multiline Styles нажмите кнопку OK.
Диалоговое окно Multiline Styles исчезнет с экрана.

Text 4

Определение контура в сложном рисунке

Открывается диалоговое окно Boundary Hatch.

Windows

В палитре Hatch на панели инструментов Draw выбирается пункт Hatch.

DOS / Unix

В меню Draw выбирается пункт Hatch.

Указываются свойства контура.

В разделе Boundary выбирается пункт Advanced.
На экране появится диалоговое окно Advanced Option.

В диалоговом окне Advanced Option выбирается кнопка Make New Boundary Set.

В командной строке Select objects указываются угловые точки, чтобы определить

контур.

В диалоговом окне Advanced Option нажимается кнопка OK.
Диалоговое окно Advanced Option исчезнет с экрана.

В диалоговом окне Boundary Hatch выбирается пункт Pick Points.

Указывается внутренняя точка.

Нажимается клавиша Return.

В диалоговом окне Boundary Hatch выбирается пункт Apply, чтобы применить штриховку.
Диалоговое окно Boundary Hatch исчезнет с экрана.