AGILE

Automatic Generation of Instructions in Languages of Eastern Europe

---

Title ***Evaluation of the final prototype***

Authors Tony Hartley
Donia Scott
Ivana Kruijff-Korbayová
Serge Sharoff
Lena Sokolova
Danail Dochev
Kamenka Staykova
Martin Čmejrek
Jiří Hana
Elke Teich

Deliverable *Eval2-Bu, Eval2-Cz, Eval2-Ru*

Status *Final*

Availability *Public*

Date *12 October 2000*

INCO COPERNICUS PL961104

Abstract:

This document constitutes the deliverables EVAL2-Bu, EVAL2-Cz, EVAL2-Ru of Workpackage 9, Task 9.2 of the AGILE project. The purpose of this workpackage is to assess the usability of the integrated system and the correctness and appropriateness of its output in three languages the at the final prototype stage, paying particular attention to the appropriateness of the different styles available for the user to specify. The deliverable describes the design of the evaluation, which addresses: the *linguistic coverage* of the resources developed within the project; the *usability* of the integrated system for creating and editing text specification models; and two dimensions of text quality—the *acceptability* of the output texts as a first draft of a user manual, and the *grammaticality* of these texts. 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 that 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.

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

1. Introduction 6
2. Evaluation of the usability of the integrated system 8
   2.1 Methodology for the usability evaluation 9
      2.1.1 Training session 9
      2.1.2 Testing session 9
      2.1.3 Filling in questionaires 12
   2.2 Results of the usability evaluation 13
      2.2.1 Analysis of task models produced in Task 4 13
      Total 23
      2.2.2 Rates and comments for the usability evaluation 23
3. Evaluation of the output text quality: acceptability 29
   3.1 Methodology for the acceptability evaluation 29
      3.1.1 Assessing the quality of Overviews 29
      3.1.2 Assessing the quality of Procedures as generated relative to their wordings in the reference texts 30
      3.1.3 Assessing the intrinsic quality of Procedures 31
      3.1.4 Assessing the quality of Quick reference texts 31
   3.2 Results of the acceptability evaluation 32
      3.2.1 Bulgarian 32
      3.2.2 Czech 34
      3.2.3 Russian 37
4. Evaluation of the output text quality: grammaticality 41
   4.1 Methodology 41
      4.1.1 The evaluators and their task 41
      4.1.2 The error categories 41
   4.2 Results 42
      4.2.1 Grammaticality—observed error types 42
      4.2.2 Other errors identified 43
   4.3 Conclusions 44
      4.3.1 Grammatical errors 44
      4.3.2 Other errors 45
      4.3.3 Further remarks 45

5. Linguistic coverage evaluation 46
5.1 Coverage in regions of the lexico-grammars 46
5.1.1 Transitivity 48
5.1.2 Aspect 52
5.1.3 Mood 53
5.1.4 Voice 55
5.1.5 Modality 56
5.1.6 Word order 57
5.1.7 Textual conjunctions 59
5.1.8 Clause complexity 59
5.1.9 Determination 63
5.1.10 Subject Dropping 63
5.1.11 Numerals in nominal group 65
5.1.12 Subject-Verb agreement 66
5.1.13 Nominal group agreement 66
5.2 Coverage of target texts 67
6. Evaluation of exploitation potential 71
7. Conclusions 72
References 73
Appendices 75
Appendix 1: The usability evaluation scenario 75
Appendix 2: The design of usability evaluation questionnaire sheets 77
Appendix 3: Task model specifications for usability evaluation 79
Appendix 4: The design of text quality (acceptability) questionnaire sheets 81
Appendix 5: Target texts for text quality (grammaticality) evaluators 84

Table of Figures

Figure 1 Task 1 specification..... 10
Figure 2 Task 2 specification example..... 11
Figure 3 Task 4 specification example..... 11
Figure 4 Distribution of errors across sites..... 21
Figure 5 Overall distribution of errors..... 22
Figure 6 SPL macros for textual conjunctives..... 59

Table 1 Distribution of texts between partners..... 12
Table 2 GA analysis..... 23
Table 3 Usability evaluation questionnaire rates..... 24
Table 4 Summary of usability evaluation comments..... 28
Table 5 Grammatical error categories and codes..... 42
Table 6 Coverage of phenomena in the coded corpora..... 70

1. Introduction

The primary goal of AGILE is to develop a prototype text generation system that can assist technical writers in the production of user manuals in the CAD/CAM domain and in selected languages of Eastern Europe (Bulgarian, Czech and Russian). This goal is approached by generating multilingual texts from a common semantic representation of the procedural aspects of the software user's tasks. Multilingual documentation is thus generated directly from a domain task model specified through the AGILE interface, in contrast to the current practice of producing the initial documentation is in one language only and subsequently translating it.

The objective of this workpackage is to assess the usability of the final prototype and the correctness and appropriateness of its output, in all three languages and in several text types: full instructional texts, ready-reference (short) instructions, overviews, and functional descriptions (of functions and actions of the software being documented[1]). The work reported here utilises resources developed in other workpackages, in particular, lexicogrammatical resources (WP6.3, 7.3), text structuring resources (WP5.3) and the integrated system (WP 8.2). These resources define a linguistic potential for the project languages, while the evaluation workpackage tests a particular application of this potential to generating texts for the practical benefit of experts responsible for producing multilingual documentation.

The methodology for evaluation of the Final Prototype utilises results of the evaluation of the Intermediate Prototype of AGILE (WP9.1, deliverable EVAL1). The GIST and DRAFTER projects previously developed at ITRI 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]. Within workpackage WP9.2 we conducted an evaluation of three parameters:

1. evaluation of the usability of the system;
2. evaluation of the output text quality with respect to the acceptability and grammaticality of the output texts;
3. evaluation of the linguistic coverage of the lexicogrammatical resources;

The Usability evaluation (Section 2) tests the usability of AGILE to specify the content of multilingual documents. This group of evaluators was composed of professionals in Information Technology. The evaluators started from task model specifications in order to produce the respective task models and to generate multilingual documents from them. The ask model specifications which were provided for this evaluation are given in Appendix 4.

The Text Quality evaluation (Section 3) includes two subtasks:

---

[1] Text types and styles available in the Intermediate and Final Prototypes are discussed in the TEXS2 and TEXS3 deliverables.

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

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

- **acceptability:** how well written the output of AGILE is perceived to be, in terms of its suitability as a final draft of a CAD-CAM manual, and how closely it matches the reference texts;
- **grammaticality:** how grammatically correct the output of AGILE is.

For the first subtask of the Text Quality evaluation, technical writers/translators of software manuals were asked to judge the quality of texts produced in different styles from several task models. For the second subtask of this evaluation, linguists with the respective native language were asked to judge the quality of texts produced in different styles from one of the task models. The texts that were provided for this evaluation type are given in Appendix 5.

The Linguistic Coverage evaluation (Section 5) was conducted within the AGILE team and includes the specification of the linguistic phenomena which are covered by the resources and a comparison of this list with phenomena that really occur in instructional texts (the target corpus of AGILE was coded within WP3.1 deliverable, CORP).

2. Evaluation of the usability of the integrated system

The group of evaluators for the usability evaluation 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. The groups of evaluators consisted of six people at each site and each group used the respective localised version of the system (i.e., Bulgarian, Czech and Russian). All evaluators worked on the version available in their own native language, and all were experienced in the use of Windows-based computer applications. The Bulgarian group included 3 associate professors, one programmer and one student, all experienced in software development and graphical user interfaces, and one research associate with special experience in computational linguistics. The Czech group included four graduates in computer science with rich experience in software development, one graduate in mathematics with experience in translating documentation and in CAD, and one programmer. Three of them had participated in the Intermediate Prototype evaluation. The Russian group of evaluators included an associate professor with experience in interfaces and knowledge representation; and five graduate students in Information Technology. Three persons had participated in the Intermediate Prototype evaluation.

The usability evaluation included several stages, which were arranged in an identical manner at each site. Firstly, evaluators received training in basic concepts of the AGILE project and the final prototype system and acquired knowledge editing skills for using the interface. They were also introduced to the terminology important for performing evaluation task:

- ***task model specification***—a description in 'pseudo-English' of the intended content of a model, used to guide the subjects' model-building activities.
- ***task model***[4]—this typically represents a single task for users of the CAD-CAM manual, such as 'To define a style of a multiline'.
- ***task model set***[5]—this consists of a set of task models and typically represents a sub-section of a manual, such as a chapter.

Evaluators then 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 a user manual for a fictitious CAD/CAM system. This stage of evaluation also addressed multilinguality through exchange between different language groups of task models produced by speakers of a different language.

After the testing session, evaluators filled in a questionnaire, in which they summarised their comments derived from their activity and scored the abilities of the system with respect to relevant parameters. Additionally, the task models produced by them were analysed by members of the AGILE team with respect to conformance to their specifications (each site analysed the task models produced by its evaluators).

---

[4] formerly called 'A-box'

[5] formerly called 'mega A-box'

2.1 Methodology for the usability evaluation

2.1.1 Training session

The evaluation proper was preceded by 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.

At the initial meeting, each evaluator was briefly introduced to basic concepts of AGILE and received the localised versions of the Conceptual Tutorial to study at home prior to the training session.

The training session took place on September 5, 2000. It 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 Final Prototype. After presentation of the interface, evaluators were instructed in methods for specifying a fragment of a manual in close resemblance to real conditions for authoring multilingual user manuals. This stage of training was guided by the Training Manual and included two basic phases:

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

During this stage, the evaluators were given the opportunity to familiarise themselves with the interface by following the steps defined in the Training Manual, in particular, they were trained to use the following functions of the interface:

- Cut, Copy and Delete
- 'Paste as a Copy' and 'Paste as a reference'
- Generate Marked and Generate All
- Save Model and Save Generated Text
- Setting Parameters for Text Style and Text Type
- Font and size changes
- Mark and Unmark
- Hide and Unhide

After completion of the two phases defined by the Training Manual, they continued training to familiarise themselves with the interface and domain concepts. 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 to 6 hours. The successful completion of the training session meant that evaluators were trained in the interface capabilities and able to test the capabilities of the Final Prototype.

2.1.2 Testing session

The testing of the interface was carried out on September 8, 2000. Unlike the Training session, during which evaluators were guided by the AGILE team at each site, the Testing session involved independent activity of evaluators to accomplish tasks they were assigned. AGILE team members monitored their actions, but were not allowed to help them.

When the evaluators executed their tasks, they used semi-formal task model specifications in a notation, which was developed to ensure unambiguous understanding of

the content to be specified. An example specification is given in Figure 1. Task model specifications are shown framed in a box, whose first line expresses the task of a procedure. Steps are expressed by imperative clauses, their side-effects by indicative clauses. A sequence of substeps which constitute a method for achieving a complex instruction step is indented. Alternative methods are marked by M followed by the method number. 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 bold face. Labels of entities are marked in italics. For non-bold, non-labelled entities, the reader should check for their previous occurrence in the specification. Since evaluators used localised versions of the interface with names of concepts of the Domain Model (DM) from their respective languages, they received both translations of the English original specifications. The complete set of specifications in English used for evaluation is given in Appendix 3.

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

Task 1: Edit a simple task model (10 min)

Task 2: Create a simple task model (20 min)

Task 3: Compose a task model set (20 min)

Task 4: Create a complex task model (120 min)

Task 5: Compose a complex task model set (30 min)

The task models included for Tasks 1, 3, 4 and 5 were developed by the AGILE team prior to evaluation. Their specifications were not available to the subjects, who understood the original content by reading the screen presentation. The initial specification for the task model used for Task 1 (MiniTask-1a) is given in Figure 1.

To draw **a polyline**
  Specify **the first point** of the polyline.
  Specify **the endpoint** of the polyline.

Figure 1 Task 1 specification

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

Specify a precondition: “start-command the *Pline* command.”

Change the last step to mean: “Specify the endpoint of **a segment** of the polyline.”

Task 2 involved building a task model consisting of several steps with low complexity of embedded sub-methods. The task model was specific for each site; an example of a specification (MiniSpec-2, BAS specific) is given in Figure 2.

Draw **a line-arc-polyline**
First, start-tool 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 *l* command line. The *Line mode confirmation* dialog box appears.
Draw **a line segment.**
End the polyline.
Press the *Return* key.

Figure 2 Task 2 specification example

Task 3 involved constructing a task model set from four simple task models, and generating from it output in all text types and in all 3 languages.

Task 4 involved creating two complex task models: one model that was identical for all sites, and one model that was specific to a site. Both models were designed as extensions of the ideal solution for specifications used for Tasks 1 (MiniSpec-1) and 2 (respective site-specific models MiniSpec-2, 3 or 4). The task model specification MaxiSpec-4 used by RRIAI in Task 4 is shown in Figure 3. At the end of Task 4, best results of site-specific models (one per site) were sent to two other sites.

Define **a style** of **a multiline** in **a drawing**
First, open-screen-object the *Multiline Style* dialog box. The *Multiline Style* dialog box appears.
M1. Using the *Windows* operating system: choose the *Multiline Style* option from the *Multiline Style* flyout on the
Data toolbar.
M2. Using the *DOS* or *UNIX* operating system: choose the *Multiline Style* option from the *Data* menu.
Specify **the properties** of the multiline.
Click the *Multiline Properties* button. The *Multiline Properties* dialog box appears.
Specify **the offset** of **the elements** of the multiline.
Display **vertices** of the multiline.
Choose the *Display Joints* option.
In the *Caps* heading, specify the properties of **the startpoint** of the multiline.
In the *Caps* heading, specify the properties of **the endpoint** of the multiline.
Display **background colour** of the drawing.
In the *FIll* heading, choose the On option.
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** in the *Description* field.
Add the style to the drawing.
Click the *Add* button.
Click the *Save* button.
Click the *OK* button in the *Multiline Style* dialog box. The *Multiline Style* dialog box disappears.

Figure 3 Task 4 specification example

Task 5 involved composition of a task model set from four complex task models, and generation of output in all text types in the respective native language of the site. This task simulated the exchange of models between offices of an international company involved in

multilingual authoring. Table 1 depicts the distribution of texts between partners (specifications of complete texts are given in the Appendix 3).

	Bg	Cz	Ru
Produced	MaxiSpec-2	MaxiSpec-3	MaxiSpec-4
Received	MaxiSpec-3, MaxiSpec-4	MaxiSpec-2, MaxiSpec-4	MaxiSpec-2, MaxiSpec-3

Table 1 Distribution of texts between partners

2.1.3 Filling in questionaires

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 answers to questionnaire (the questionnaire design is included in Appendix 2).

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

Score	Explanation
4	Excellent
3	Good
2	Poor
1	Useless

Figure 4 Subjective judgment scale

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

Figure 5 Usability parameters

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 2.2.2.

2.2 Results of the usability evaluation

2.2.1 Analysis of task models produced in Task 4

2.2.1.1 Analysis of the content

We performed an in-depth analysis of the task models produced by each evaluator within Task 4 (cf. the description of Task 4 in Section 2.1.2). The Task 4 task models were compared against the verbal specifications received by evaluators and errors were counted.

All errors possible in development of task models can be classified into the following groups:

A. Concept-name errors, e.g., use of SPECIFY instead of 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

i. 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;

ii. 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 task model 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.

The tables in the following sub-sections show the number and distribution of each type of error for each of the two task models and overall distribution of errors per site.

2.2.1.2 Analysis using the Generation Accuracy metric

2.2.1.2.1 Basics of the Generation Accuracy metric

Since the procedure for checking the accuracy of task models in terms of their formal content is extremely time-consuming, we also sought to capture their accuracy in terms of the difference between texts generated from them and texts generated from ideal solutions.

Generation Accuracy is a simple metric suggested in the literature on machine translation (Alshawi et al., 1998), based on the string edit distance between some reference text and the output of the system. Use has also been made of generation accuracy in evaluating NLG output (Bangalore and Rambow, 2000; Bangalore, Rambow and Whittaker, 2000), who extend the Alshawi *et al.* metric to account for some of the special properties of the text generation task. Text generation involves ordering word tokens; thus, corrections may involve re-ordering tokens. In order not to penalise a misplaced constituent twice—as both a deletion and an insertion—the generation accuracy metric treats the deletion (D) of a token from one location and its insertion (I) at another location as a single movement (M). The remaining deletions, insertions, and substitutions (S) are counted separately. Generation accuracy is given by the following equation, where R is the number of (word) tokens in the reference text.

$$GenerationAccuracy = \left(1 - \frac{M + I + D + S}{R}\right)$$

For Bangalore and his colleagues, the reference text is the desired text; it is given *a priori* by a corpus representing the target output of the system (i.e., the result of applying the corrections to the generated texts). The generation accuracy of a string from the actual output of the system is computed on the basis of the number of movements, substitutions, deletions and insertions required to edit the string into the desired form. An utterance needing no corrections would thus yield a rating of 1, and one needing as many corrections as words in the reference text would yield a rating of 0. For example, a generated utterance such as:

```
She open the file clicking in the button.
```

with the desired, target form:

```
She opens the file by clicking on the button.
```

and thus, in the AGILE grammaticality evaluation (Section 4), annotated as:

```
She open/s AG/MO the file by CO clicking ~~in~~/on PR the button.
```

would lead to a generation accuracy rating of:

$$1 - \frac{M(0) + I(1) + D(0) + S(2)}{R(9)} = 0.667$$

This metric, as given, raises a number of issues for us.

Errors vs. edits: In our scheme, it is possible that a single word token manifests more than one error, e.g. a verb may be in the wrong aspect (AS) and its inflection incorrect in both number (NU) and person (PE). Yet the approach taken by Bangalore *et al*. would treat this token as needing a single corrective editing operation (substitution). In their sense, *accuracy* is a crude measure of corrective effort.

Granularity: Our classification system allows for errors *below* the level of a word; therefore, we must, in principle, count both word roots and their inflections to arrive at R. However, if one were strictly following Bangalore *et al.,* an utterance comprising a single word containing an error such a MO (the inflection of the noun or verb is absent) would be judged as needing one correction (substitution) and yield an accuracy rating of 0. On the other hand, if the more fine-grained definition of R were applied, *the root+inflection unit* would be judged as needing one correction (insertion), which would yield an accuracy rating of 0.5.

Multiple ways of counting moves: Counting the number of moves is not necessarily straightforward. Let us suppose, for example, the following scenario for a generated utterance and its correction:

Variant 1: $B_1\, B_2\, B_3\, A_1\, A_2\, A_3\, A_4$

Variant 2: $A_1\, A_2\, A_3\, A_4\, B_1\, B_2\, B_3$

Transforming either of these utterances into the other can be accomplished by either three moves (i.e., moving the Bs) or four (i.e., moving the As). The first of these would yield a generation accuracy value of 0.571 and the second 0.429. Alternatively, one might even want to consider the case where all the Bs (or all the As) were moved collectively. Since we are not so much concerned with how judges actually make this transformation but rather with the result of the transformation, it is not at all clear what the correct way of counting should be.

What is the Reference?: By definition, the reference text from which R is calculated should be a text that is constant between judges/judgments—a *gold standard*, so to speak. In the case of MT, this is the target text, which is given by some test corpus; the actual output text may vary between languages or between translation systems. One then counts the number of edits needed to go from the target text to the actual text. In the case of NLG, however, there is no pre-existing target text to act as reference. If we take *target* texts to mean the *desired* text, we might wish to consider the correction (i.e., the text that would result from applying the judge's corrections) as the reference. However, this is very likely to vary between judges. On the other hand, what is certainly constant is the *generated* text; we would then count the number of edits needed to go from the actual text to the target text. One might consider this to be of no consequence. However, the two approaches can lead to different accuracy values. Take the earlier example of `click button to open file`: as we saw, when R is calculated from the desired text, the result is a value of 0.667. If, however, R is calculated from the generated text, the value will be 0.625. In both cases, one would have to normalise over judges to account for between-subject variation; but in principle this will not make the difference go away.

Moreover, if one compares the texts generated from two task models, a task model M1 created by an evaluator, and the "ideal" task model M2, there can be the following additional

complication. The difference between M1 and M2 can give rise to different text plans and, as a result, differences in the generated texts which stem not from a difference in the modelled content, but from the fact that the text planner made different decisions, because of a difference in complexity between M1 and M2. The results obtained by the GA metric can therefore be skewed.

2.2.1.2.2 Methodology for measurement

In the case where the GA metric is applied to the accuracy of task models, the situation of what is considered as a token for the R value is more complex. One possibility for measuring the number of tokens is to treat a proposition as a token, i.e. 'Save the drawing into a file by choosing Save in the File menu' contains two tokens. This is the easiest way to calculate tokens. However, a proposition can have possibly several errors (thus, decreasing the GA value significantly below zero). Another possibility for measuring R is to calculate the number of words. The disadvantage of this approach is that the metric's value will differ significantly from language to language (different collocations, articles, grammatical constructions result in a different word count). Also, while this number reflects grammatical differences (since such errors are mostly expressed in words), it does not reflect the structural complexity of the output text. So, in this case, structural errors, for example, when a precondition is expressed as the first step of a method, do influence the total number of errors, but their absence does not increase the number of correct cases.

The measure we chose was based on the number of concepts and filled slots in a task model. This value is easy to count and it corresponds well to both the conceptual content and the structural complexity of a task model. Since the exact count is also an effort-consuming procedure, the number can be approximated by the output produced by the 'print-abox' function.

For counting errors, we compared Full instructions (with explicit side-effects), which are generated from task models produced by evaluators, against Full instructions generated from task models, which ideally conform to the original specification. Since we analyse texts, not separate sentences as assumed in GA, the D, I, S and M errors may be applicable to constituents within sentences (for example, prepositional phrases) or to complete sentences. In the first case, an omission of a prepositional phrase (for example, *from the X toolbar* in *Choose Y from the X toolbar*) is also considered as a D error.

In our classification of errors, the following types can influence the GA counts:

A. Concept-name errors: this should cause the Substitution error of the GA

B. Co-reference errors: this can cause the Move error of the GA, if the error is reflected in the word order in a sentence (not necessarily for all languages)

C. Configuration errors

i. some content is missing: this should cause the Deletion error of the GA

ii. some content is modelled incorrectly: this can cause the Substitution or Move errors of the GA (not necessarily).

2.2.1.3 Comparison for each site

2.2.1.3.1 Bulgarian

	MaxiSpec1			MaxiSpec2			MaxiSpec1, 2		
Errors	# of errors	**relative %**	**deviation**	# of errors	**relative %**	**deviation**	**# of errors**	**relative %**	**deviation**
A	0	0.00%	0.00%	2	22.22%	77.78%	2	13.33%	86.67%
B1	4	66.67%	66.67%	0	0.00%	0.00%	4	26.67%	26.67%
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%	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	1	16.67%	83.33%	0	0.00%	0.00%	1	6.67%	43.33%
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	1	16.67%	16.67%	5	55.56%	55.56%	6	40.00%	60.00%
Cii4	0	0.00%	0.00%	2	22.22%	77.78%	2	13.33%	86.67%
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	6			9			15		

The errors may be classified in two groups:

- Errors of classes A (2 - use of different label names), Ci6 (1 - use of different slot filler) and Cii4 (2 - a `subsequent' step is modelled as a method rather than as a procedure). All these errors are not constant for the same subjects and hence may be caused by a temporary lack of attention.
- Errors of classes B1 (4 - missing references to introduced objects) and Cii3 (6 - a precondition modelled as a procedure). They may be due to shortcomings in the training of the subjects.

	MaxiSpec1						MaxiSpec2						MaxiSpec1, 2					
	I	D	S	M	GA	E	I	D	S	M	GA	E	I	D	S	M	GA	E
Subject1	0	0	0	0	1,00	0	0	0	2	0	0,98	2	0	0	2	0	0,99	2
Subject2	0	0	0	0	1,00	0	0	0	0	0	1,00	0	0	0	0	0	1,00	0
Subject3	0	4	1	0	0,90	5	0	0	3	0	0,97	3	0	4	4	0	0,95	8
Subject4	0	0	1	0	0.98	1	0	0	0	1	0,99	1	0	0	1	1	0,99	2
Subject5	0	0	0	0	1,00	0	0	0	2	0	0,98	2	0	0	2	0	0,99	2
Subject6	0	0	0	0	1,00	0	0	0	0	1	0,99	1	0	0	0	0	0,99	1

Notes:

- errors of classes A (use of different label names) and Ci6 (use of different slot filler) are substitution errors;
- errors of class Cii4 (a `subsequent' step is modelled as a method rather than as a procedure) - Move error (chunks of texts are 'moved' in another semantic paragraph);
- errors of class B1 (missing references to introduced objects) - Deletion error (missed definite article in the generated BG text);
- errors of classCii3 (a precondition modelled as a procedure) - Substitution error

2.2.1.3.2 Czech

	MaxiSpec1			MaxiSpec4			MaxiSpec1, 4		
Errors	# of errors	**relative %**	**deviation**	# of errors	**relative %**	**deviation**	**# of errors**	**relative %**	**deviation**
A	2	40.00%	60.00%	0	0.00%	0.00%	2	10.00%	23.33%
B1	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
B2	0	0.00%	0.00%	1	6,67%	26.67%	1	5.00%	28.33%
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	6,67%	18,33%	1	5.00%	9,29%
Ci5	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Ci6	2	40.00%	40.00%	12	80.00%	80.00%	14	70.00%	30.00%
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%

	MaxiSpec1			MaxiSpec4			MaxiSpec1, 4		
Errors	# of errors	**relative %**	**deviation**	# of errors	**relative %**	**deviation**	**# of errors**	**relative %**	**deviation**
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%	1	6,67%	18,33%	1	5.00%	9,29%
Cii-other	1	20.00%	20.00%	0	0.00%	0.00%	1	5.00%	9,29%
Total errors	5			15			20		

- The Czech evaluators produced 17 errors, most of them could be interpreted as a loss of concentration. The most frequent error was missing location "command line" in sentence "Enter the endp command line", which was translated into Czech as "Na(on) příkazovém(command) řádku(line) zadejte(enter) *kon*(*endp*)". It was caused by the fact that the name of procedure in Czech means "Enter the command line" and no location specification seems to be needed.
- A common error was varying between "first point" and "start point", which is almost identical in Czech.
- A high number of errors of Subject 5 in MaxiSpec3 was caused by loss of concentration – he omitted one line from the text specification.

	MaxiSpec1						MaxiSpec3						MaxiSpec1, 3					
	I	D	S	M	GA	E	I	D	S	M	GA	E	I	D	S	M	GA	E
Subject1	0	0	0	0	1,00	0	0	2	0	0	0,98	2	0	2	0	0	0,99	2
Subject2	0	0	0	0	1,00	0	0	2	0	0	0,98	2	0	2	0	0	0,99	2
Subject3	0	0	0	0	1,00	0	1	2	0	0	0,97	3	1	2	0	0	0,99	3
Subject4	0	0	1	0	0,98	1	0	2	0	0	0,98	2	0	2	1	0	0,99	3
Subject5	0	1	2	0	0,94	4	0	3	0	0	0,97	4	0	4	2	0	0,97	8
Subject6	0	0	0	0	1,00	0	0	2	0	0	0,98	2	0	2	0	0	0,99	2

2.2.1.3.3 Russian

Analysis of the content combined for all evaluators:

	MaxiSpec1			MaxiSpec4			MaxiSpec1, 4		
Errors	# of errors	**relative %**	**deviation**	# of errors	**relative %**	**deviation**	**# of errors**	**relative %**	**deviation**
A	4	40.00%	60.00%	3	10.71%	10.71%	7	18.42%	21.58%
B1	6	60.00%	60.00%	23	82.14%	82.14%	29	76.32%	36.32%
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%	2	7.14%	17.86%	2	5.26%	19.74%
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	0	0.00%	0.00%	0	0.00%	0.00%	0	0.00%	0.00%
Total errors	10			28			38		

One evaluator paid no attention to co-reference (in spite of the training procedure): he introduced new instances each time, when a concept name appears in the specification. Other evaluators also made co-reference errors relatively often. Thus, B1 errors occupy more than 75% of the total error count.

Error types and GA values for each evaluator and task, as well as number of Task model errors from the content comparison (E column):

	MaxiSpec1						MaxiSpec4						MaxiSpec1, 4					
	I	D	S	M	GSA	E	I	D	S	M	GSA	E	I	D	S	M	GSA	E
Subject1	0	0	0	0	1.0000	1	0	1	0	0	0.9938	7	0	1	0	0	0.9958	8
Subject2	0	0	1	0	0.9870	2	0	1	1	0	0.9877	2	0	1	2	0	0.9874	4
Subject3	0	0	0	0	1.0000	0	0	1	0	0	0.9938	1	0	1	0	0	0.9958	1
Subject4	0	0	2	0	0.9740	6	0	0	1	0	0.9938	12	0	0	3	0	0.9874	18
Subject5	0	0	0	0	1.0000	1	0	0	0	0	1.0000	0	0	0	0	0	1.0000	1
Subject6	0	0	1	0	0.9870	1	0	0	0	0	1.0000	2	0	0	1	0	0.9958	3

2.2.1.4 Joint comparison for all sites

No configuration errors of the following types:

(C.i.1) missing precondition,

(C.i.2) missing constraint,

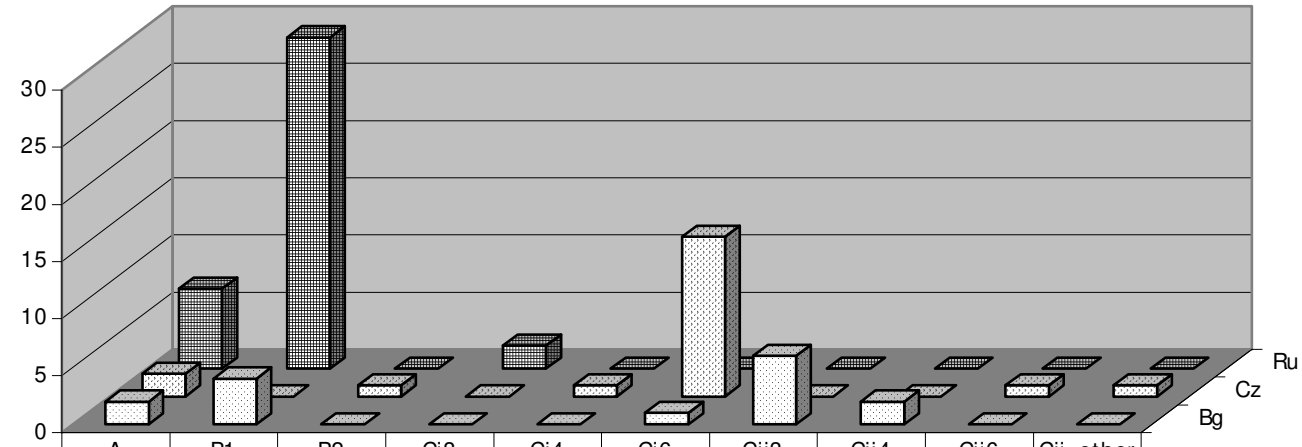
(C.i.5) missing method,

(C.ii.1) a side-effect is modelled as a (separate) procedure,

(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 Task model across all sites. They are omitted from the joint tables below.



	A	B1	B2	Ci3	Ci4	Ci6	Cii3	Cii4	Cii6	Cii- other
Bg	2	4	0	0	0	1	6	2	0	0
Cz	2	0	1	0	1	14	0	0	1	1
Ru	7	29	0	2	0	0	0	0	0	0

Figure 6 Distribution of errors across sites

Figure 6 suggests that all errors are mostly site- and/or evaluator-specific. One Russian evaluator always used a new instance of a concept instead of a reference to an existing instance. This skyrocketed the number of B1 errors for Russian (he contributed to more than a half of errors of this type across all other evaluators). This error can be remedied by possibilities of the interface for keeping a list of existing instances. The high number of C.ii.3 errors for Bulgarian may be due to shortcomings in the training process of the evaluators. C.i.6 errors for Czech site are caused by missing location "command line" in sentence "Enter the endp command line", which was translated into Czech as "Na(on) příkazovém(command) řádku(line) zadejte(enter) *kon*(*endp*)". It was caused by the fact that

the name of procedure in Czech means “Enter the command line” and no location specification seems to be needed.

The only error that is shared by all sites is of the A type, which is caused by an error in entering the concept name. This is related to the fact that the manual selection of concepts from the list of their names is error-prone and can be altered because of evaluator-specific interpretation of concept names. Also, there is a difference between the name of a concept in the list and its linguistic realisation in natural language expressions.

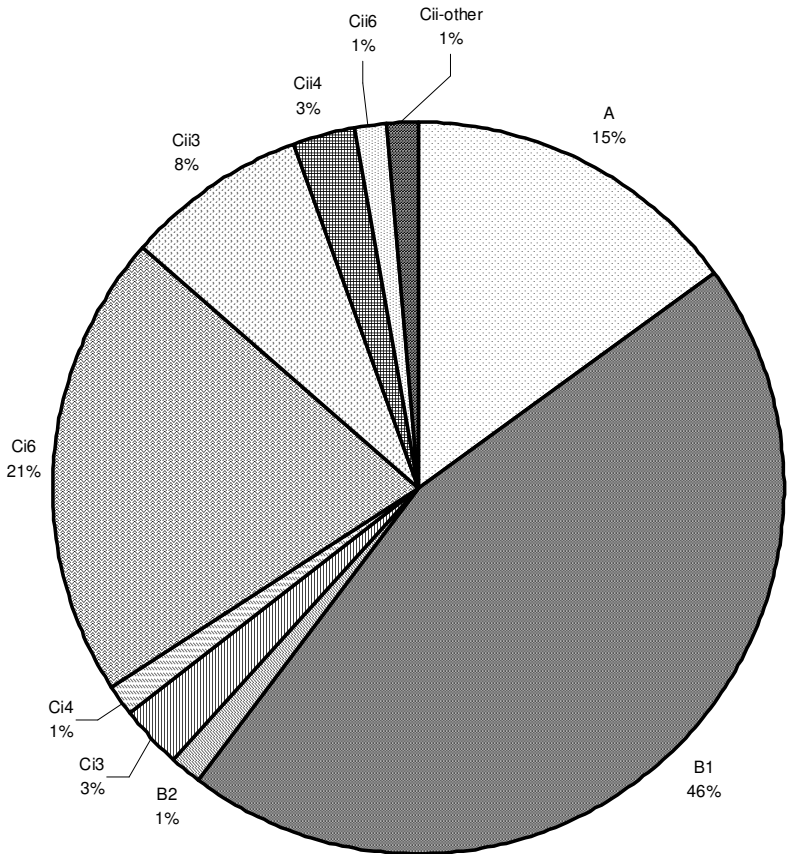


Figure 7 Overall distribution of errors

As for the overall error rate depicted in Figure 7, almost half of errors (47%) pertain to improper co-indexation. The knowledge editor of AGILE lacks tools for tracking interface objects that are introduced in the text, so the burden of proper co-referencing is put on the user. However, the user prefers to introduce a new object instead of looking for its previous appearances, because navigation and searching for an object introduced earlier in a complex Task model is not easy. Although this facility had been specially highlighted during the training, some evaluators did not use it.

There is also a high proportion of concept selection errors (16%). The most of them are caused by improper translation of English concepts and differences in meanings of terms between English and target languages. Another reason for this error concerns lack of documentation on the DM concepts. Some optional slots were not filled by evaluators in spite of the specification. This caused a significant amount of C.i.6 errors (19%), however, as Figure 6 suggests, this error is specific to Czech.

Table 2, which also contains the total number of errors from the formal content analysis. The measure of Task model accuracy using the GA metric is cost-effective, since it is based on comparison of differences between two texts in natural language (texts are produced respectively from an ideal solution and a task model under consideration). Analysis of the formal content of task models is much more effort-consuming. It can reveal additional errors, since GA analysis does not reflect some types of errors, most notably, B1 for Czech and Russian. So the greater number of B1 errors adds no contribution to the GA value, and most errors captured by GA correspond to alteration of concept names and missed content (forgotten steps or their optional slots).

	Total count by all evaluators					
	I	D	S	M	GA	E
Bulgarian	0	5	9	2	0.9882	15
Czech	1	14	3	0	0.9867	20
Russian	0	3	6	0	0.9937	38
Total	**1**	**22**	**18**	**2**	**0.9896**	**73**

Table 2 GA analysis

2.2.2 Rates and comments for the usability evaluation

Ref	Question	Average marks for each site			
		Bg	Cz	Ru	Total
1	Intuitiveness of operations with the interface (menus, dialog boxes, etc)	3.17	2.83	3.67	3.22
2	Abilities of the interface for creating and modifying a text content.	2.83	2.33	3.67	2.94
3	Abilities of the interface for presentation of a text content.	2.83	3.17	3.00	3.00
4	Intuitiveness of content specification notions (goal, step, precondition, etc)	2.79	3.67	3.33	3.26
5	Adequacy of CADCAM notions (click, button, multiline, etc)	3.67	3.00	3.00	3.22
6	Robustness of interface operations	2.67	2.83	3.00	2.83
7	Does the generated output express the content you specified?	3.29	3.00	3.33	3.21
8	Adequacy of the user documentation	2.83	2.66	3.67	3.05

Table 3 depicts distribution of rates given for items in the usability evaluation questionnaire (an average value of rates for each site and in total).

Ref	Question	Average marks for each site			
		Bg	Cz	Ru	Total
1	Intuitiveness of operations with the interface (menus, dialog boxes, etc)	3.17	2.83	3.67	3.22
2	Abilities of the interface for creating and modifying a text content.	2.83	2.33	3.67	2.94
3	Abilities of the interface for presentation of a text content.	2.83	3.17	3.00	3.00
4	Intuitiveness of content specification notions (goal, step, precondition, etc)	2.79	3.67	3.33	3.26
5	Adequacy of CADCAM notions (click, button, multiline, etc)	3.67	3.00	3.00	3.22
6	Robustness of interface operations	2.67	2.83	3.00	2.83
7	Does the generated output express the content you specified?	3.29	3.00	3.33	3.21
8	Adequacy of the user documentation	2.83	2.66	3.67	3.05

Table 3 Usability evaluation questionnaire rates

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

2.2.2.1 Intuitiveness of operations with the interface

The procedure for construction of a model set is non-intuitive. The "Cancel" button means a positive value: to accept the chosen task models. (Bg, Ru - 2)

There is also no way to remove an erroneously chosen task model from the set. (Ru)

The interface for model composition should be improved – it is necessary to mark such models that have been already loaded. (Bg-2)

It would be also desirable to have a possibility to load several models at once. (Bg-2, Ru)

Should not the order be Delete—Cut—Copy (seeing that cutting is a combination of copying and deleting) rather than Cut—Delete—Copy? (Bg)

When opening a "list of methods" or "list of procedures" to open these lists (to create the first method or procedure), it is necessary to click twice – though it is clear that one click should suffice. (Cz-2)

The functions HIDE and UNHIDE are very worthy, although both functions take too much time. (Cz)

User interface has no keyboard support, which makes it very difficult and time consuming to use.(Cz, Ru)

More flexible and configurable user interface and built-in help would be expected. (Cz)

2.2.2.2 Abilities of the interface for creating and modifying text content

There is no "Undo" option. (Bg - 3, Ru -2, Cz-3)

When the user hides a given procedure it would be better more information of the hidden block to be visible – for example except "procedure a??" on the screen to be shown also the beginning of what the user had enter [in bold or in different colour]. This would definitely

improve user's orientation when he works with more complicated structures. (Bg -2, Ru, Cz)

Most of editing operations lead for changing the graphical appearance of the current model on the screen. The lack of the cursor pointing the place in the model where the last updates have been made also makes the model editing/creating process more difficult. (Bg - 2)

Sometimes when the user wants to generate part of a task model on which he is currently working, the system discovers errors without showing in some way their placing in the structure – this cause troubles in the process of finding and correcting these errors. (Bg, Ru)

The user often could hardly conform to the references, which he has already entered. It would be better to have a list with the already entered objects, from which to choose. No more than two or three embedding levels would be enough. (Bg - 2, Ru)

The solution to mark such portions of information that have to be inserted is very useful. (Bg)

The system must have the ability to copy parts from other task models. (Bg, Ru - 2, Cz-2)

I propose to add "auto-hide" of an element of the instructions plan when it is finished (the following element is begun to fill) (Ru)

I would minimise the quantity of steps from the STEPS element of the model to the GOAL element (now you do three clicking on by the mouce button for this). (Ru)

Instructions for the user are highly formal. I think you could get the model text automatically from the specifications written in controlled NL. (Ru)

It is not possible to "override" a value already specified – user has to delete the old value first and then specify the new one. (Cz-2)

If user want to use an already generated model and replace e.g. "polyline" (which is there on several places – co-referenced) for e.g. "arc", it is not possible. User has to delete the first occurrence of polyline, insert arc, and then delete each subsequent reference to the polyline and insert a new reference to the arc. (Cz)

If copying an object (as copy) the "inner" reference are not preserved – instead of the co-referenced objects new ones are introduced. (Cz-2)

If user wants to use a copy of an existing "structured" object to create a new "structured" object which differs from the former one in just one value, which is not "atomic" (not a "leaf"), he has to cut its "subsumed" values, change this value, and then paste the "sub-values" back – e.g. when user wants to change "start point" to "endpoint" (of a polyline). (Cz-2)

2.2.2.3 Capabilities of the interface for presenting text content

This ability very tightly depends on the created ontology of the problem domain. Such an ontology is represented only implicitly and cannot be augmented by the user at any level, which is my opinion is too strong restriction. (Bg)

The significance of the conventions regarding new lines, indentation etc., some of which clearly indicate a difference in the meaning whilst others do not, is quite opaque. (Bg)

The Bulgarian content specification labels still take up a lot of space on the screen (being long words or two- to three-word phrases, and written in all capitals, as if it weren't enough that they're in boldface). This makes the model even harder to find one's way in than it would be if the boxes had more compact labels. On the bright side, the representation of lists as sequences is a major improvement over the head–tail representation in the previous version, as is the fact that one is no longer swept back to the upper left corner of the screen after every editing operation. (Bg)

In the precondition the inner levels should be shown to the right from their current position. (Bg)

I liked the form of the instruction content presentation – by levels – and in any place you can insert a new elements. This is methodologically comfortable and meets the programming habit. (Ru)

There were not the proper and unambiguous translation of all words (e.g. Křivka = polyline/pline), perhaps it would be better to insert text content only in English language interface. (Cz-2)

2.2.2.4 Adequacy of content specification notions

Everywhere in the AGILE-system there are tags "gui-", which impede from finding a given tag faster. (Bg)

Some user actions are not Goals, for example, "click the button". (Ru)

2.2.2.5 Adequacy of CAD/CAM notions

The question is not clearly formulated – it is not clear to me why I need to evaluated a degree of intuitiveness of a given problem domain. (Bg)

"first point" is presented by two notions – "first" and "point" but the "end point" is presented by one notion. (Ru)

There is a problem in such concept as "these steps": it is clear that it refers to something in previous text, but there is no possibility to refer to those steps explicitly. (Ru)

CADCAM notions presenting actions are too language specific. You must guess the appropriate name of a concept, because there are near synonyms as, for example, 'specify-component' and 'define', which mean very different things in the system. (Ru - 3)

2.2.2.6 Robustness of interface operations [Errors, implementation]

The program is very slow, and often stops communicating for a long time without saying what it is doing. (Bg - 4, Cz)

It would be better a "progress bar" to be integrated in the system just to give information about the readiness of a given task. (Bg - 3, Ru - 3)

It should be mentioned that the interface is significantly different from the "traditional" interface of host environment (Windows) and the user should spend a lot of training time for adaptation. Accomplishing of editing process is significantly more sophisticated than the process of creating a model (Bg - 2)

It is possible to create an incorrect model by deleting an object, which has already made references to, and after that by adding the same (as type) object. As a result of such

operation a new object will be created and the references to the old, nonexisting object will still present in the model. (Bg)

The error-messages do not help to detect the given error/mistake at all. (Bg)

Items in some menus are not ordered correctly according to Czech alphabet. (Cz)

The system has much improved since the last evaluation. This version is quite robust and the graphical presentation of models is much more intelligible.(Cz)

The scroll-bars behaves sometimes a bit crazy, the position in current window is often set to unexpected part of the window, it happens also when switching between windows. (Cz-2)

Save as does not allow usage of names with more dots, e.g. om_task4.1.abx was saved as om_task4.abx. (Cz)

The program is very slow and gets slower and slower with the growth of Task model. (Cz)

2.2.2.7 Does the generated output express the content you specified?

2.2.2.7.1 Bulgarian

The syntax and punctuation of some constructions in the Bulgarian text generated are still erroneous. (The commas before the conjunction *i* 'and' should be removed; when three or more clauses are conjoined, *i* should only appear before the last one.) There is one case of untranslated words in the Bulgarian text (*USE*, which appears for Bulgarian 2nd person present tense verb form).

In cases of more sophisticated procedures the generated text needs serious post-processing efforts to remove unnecessary repetitions.

The word for Czech" is translated in Bulgarian as "чехски". I think "чешки" is more suitable.

2.2.2.7.2 Czech

Evaluators often pointed out the poor formatting of the resulting text.

Co-reference does not seem to have any impact on the output text – no contextual pronouns or (partial) ellipsis or things like that happening.

2.2.2.7.3 Russian

A subheading must be presented differently from a method, for example, "чтобы нарисовать дугу…; "Рисование дуги";

I also suppose that formatting and numeration of steps is enough, i.e. words "наконец", "также" are not needed. (Ru - 2)

Numeration of a list having only one step is funny.

2.2.2.8 Summary of comments

Table 4 shows a summary of comments to the topics of the questionnaire. In total, evaluators made 87 comments, 45 of them addressed different issues. The evaluators were mostly concerned with problems of creating and modifying a text content and robustness of system operations.

In general, the rating is positive, in spite of the difficulties the subjects may have experienced with the interface. 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)	16	9	3.22
Abilities of the interface for creating and modifying a text content.	35	14	2.94
Abilities of the interface for presentation of a text content.	7	6	3.00
Intuitiveness of content specification notions (goal, step, precondition, etc)	2	2	3.26
Adequacy of CADCAM notions (click, button, multiline, etc)	6	4	3.22
Robustness of interface operations	21	10	2.83
Total number of comments	87	45	

Table 4 Summary of usability evaluation comments

3. Evaluation of the output text quality: acceptability

3.1 Methodology for the acceptability evaluation

The group of evaluators for the acceptability evaluation 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 as their native language. The Bulgarian group of evaluators included 6 persons, IT specialists: 4 associate professors, 1 research associate and 1 programmer, all experienced in production and/or translation of software manuals in different areas. The Czech group of evaluators included 6 persons: 4 programmers experienced in production of documentation and 2 translators of technical documentation. The Russian group of evaluators included 6 persons: a translator/editor in various IT fields, a technical writer in the CAD domain, 2 associate professors with experience in documentation writing, a technical manager and a technical designer having experience in translation and writing manuals.

In the process of the evaluation, they performed four tasks, including

1. assessing the quality of Overviews,
2. assessing the quality of Procedures as generated relative to their wordings in the reference texts,
3. assessing the quality of Procedures and
4. assessing the quality of Quick reference texts (abbreviated procedural instructions).

The evaluation sheets received by evaluators are given in Appendix 4. The text quality was judged within range of Excellent to Terrible, which received the following numerical values:
4 for Excellent, 3 for Good, 2 for Poor, and 1 for Terrible

Overviews and Quick reference texts were also judged with respect to acceptability of the level of detail, using the following binary values:
2 for Satisfactory, 1 for Unsatisfactory,
If Unsatisfactory, 2 for ‘Too much detail’, 1 for ‘Too little detail’.

Each site calculated mean results over scores given by the six respective evaluators for each text and text type.

3.1.1 Assessing the quality of Overviews

All subjects received two texts in their language. One text was generated in the object-oriented overview style and another one in the process-oriented overview style from the same task model set corresponding to the MaxiModel-1, 2, 3, and 4, henceforth referred as the MaxiSet. This is the same set as used in the usability evaluation.

Each subject made four judgments of each text as a whole with respect to:

i. how well-organised it is
ii. how well-written it is
iii. how useful it is as a draft
iv. how appropriate the level of detail is

The order of presentation of the four texts was randomised across subjects. Total number of data points: 72 (4 judgments * 18 subjects).

3.1.2 Assessing the quality of Procedures as generated relative to their wordings in the reference texts

Under ***reference text*** we mean an instructional text from the AutoCAD manual—Bg, Cz and Ru versions—from which have been deleted the non-procedural parts and other expressions not supported by the Domain Model (because deemed to be beyond the scope of the project). Under ***discrepancy type*** we mean a category, such as 'word order', representing token discrepancies in the wordings of pairs of generated and reference texts. A reference text represents a benchmark against which the Procedures produced by the AGILE generator can be judged. The quality of the reference text is deemed validated by virtue of its publication in the AutoCAD manual. We set out to establish the perceived impact on quality—negative or positive—of any discrepancies in wording between generated and reference texts.

The set of discrepancy types for each of the project languages includes 10 types in total, though some discrepancies are language-dependent. The first five types are in all three languages, the other five types are specific for Czech and Bulgarian. Discrepancies between the reference texts and generated texts fall into three categories: restrictions of the conceptual representation (DM), of the text structuring module (TSM) and individual lexicogrammars (LG). The types are given in Figure 8:

1. Order of locations. In the reference texts, the locatives are *often* placed at the beginning of a sentence, while in the generated texts they are *always* placed after the predicate (TSM).
2. Nominal group realisation. This type reflects the problems of a reference to an instance, which copies the entire content, for example, "style of multiline" is repeated in generated texts, while the reference texts refer to it as "style". (TSM)
3. No reference to plan elements. Thus expressions of the reference texts like "Repeat steps 1 through 4" are not modelled in the generated texts. Also, the TSM does not output "using one of the following methods". (DM)
4. Aggregation into clause complex. They are sometimes different from the original instructions; for example, 'Press Return to complete the line, or enter c for Close to connect' is rendered as 'Complete the line' with two itemised subemthods; (TSM)
5. Sequence marking. Another set of HTML tags is used to mark text elements in comparison to the reference texts. (TSM)
6. Purpose dependent ordering. (Bg and Cz specific, LG).
7. Difference in expression of locatives, for example, distance and angle with respect to the line. (Bg and Cz specific, LG)
8. Class noun. Some labels, like OK, are expressed without a class noun, like tlačítko (Cz specific, LG).
9. Definiteness. There are differences in using articles in reference and generated texts. (Bg specific, LG)
10. Concept realisation (Cz specific, for example, the flyout is expressed in the reference texts as Z plovoucího ikonového menu, while the generated text is Z plovoucí nabídky, LG).

Figure 8 Discrepancy types

All subjects saw three examples—either actual or hand-crafted—of each of 10 discrepancy types available in their language. Each example was presented as a pair of

variant wordings A and B in a local, invariant context, for example, for the discrepancy type 4:

> To create a line:
> *(A)*
> *Specify the start point.*
> *Specify the end point.*
> *(B)*
> *Specify the start point, then specify the end point.*
> Save the line.

Each subject made one judgment of each discrepancy pair with respect to whether:

(A) is better than / the same as / worse than (B)

The order of presentation of the pairs is randomised across examples. For the purposes of the analysis below, we assume that A is the generated text, while B is the reference text. Thus, the result compares the percentage of judgment types over all discrepancies for each of respective evaluators and the mean values for each site.

3.1.3 Assessing the intrinsic quality of Procedures

All subjects received four texts generated in their language from the same four task models—MaxiModels 1, 2, 3 and 4. Each task model set gives rise to one text, in personal style, with explicit side-effects. Each subject made three judgments of each text as a whole with respect to:

i. how well-organised it is

ii. how well-written it is

iii. how useful it is as a draft

The order of presentation of the four texts is randomised across subjects. Total number of data points: 216 (4 texts * 3 judgments * 18 subjects)

3.1.4 Assessing the quality of Quick reference texts

All subjects received four texts generated in their language from the same four task models—MaxiModels 1, 2, 3 and 4. Each task model set gives rise to one text, in personal style. Each subject made three judgments of each text as a whole with respect to

i. how well-written it is

ii. how useful it is as a draft

iii. how appropriate the level of detail is

The order of presentation of the four texts is randomised across subjects. Total number of data points: 216 (4 texts * 3 judgments * 18 subjects)

3.2 Results of the acceptability evaluation

3.2.1 Bulgarian

3.2.1.1 Overviews

The mean values of the evaluators' results are shown in the table below.

The text is:	**Text1**	**Text2**	**Text3**	**Text4**	**All texts**
Well organised	2.33	2.33	2.33	2.50	2.375
Well written	1.67	2.00	1.83	2.00	2.00
Useful as a draft	2.17	2.33	2.33	2.50	2.33
Satisfactory detail	1.33	1.50	1.25	1.50	1.40
Too much / too little detail	1.50	1.67	2.00	1.67	1.71

The subjects pointed out to the undesirable number of repetitions of connectives in the clauses. Some subjects discussed that the text described notions of different level of generality in an overview.

3.2.1.2 Discrepancies

The test for discrepancies was made on 15 text examples, covering the following 5 groups of discrepancies between the Bulgarian generated (group A) and reference (group B) texts:

D1. Difference in details of the descriptions (availability/absence of actee' location, owner, type etc.)

D2. Different word order.

D3. Different wordings (due to the definition of domain model notions).

D4. Different aggregation (for time sequences , due to the lack of alternative instruction steps etc.).

D5. Different wordings, specific for Bulgarian resources (lack of articles, different prepositions etc.).

The evaluation results for the 15 text examples, given by the different subjects and the mean values are shown in the table below.

Subject	A is better	B is better	A and B are equal
1	53.3 %	33.3 %	13.3 %
2	33.3 %	53.3 %	13.3 %
3	53.3 %	20.0 %	26.7 %
4	60 %	26.7 %	13.3 %
5	20 %	60 %	20 %
6	53.3 %	33.3 %	13.3 %
Mean values	45.5 %	37.8 %	16.7 %

Thus the generated texts were evaluated slightly better than the original reference texts.

Some subjects pointed out that the limited size of the examples makes hard the evaluation because of the lacking context. One of the subject proposed to rate the examples also in terms 'A and B are equally good' and 'A and B are equally bad'.

3.2.1.3 Full instructions

The mean values of the evaluators' results are shown in the table below.

The text is:	Text1	Text2	Text3	Text4	All texts
Well organised	3.17	3.33	3.17	3.50	3.29
Well written	2.83	3.33	2.67	3.00	2.96
Useful as a draft	3.00	3.17	3.00	3.00	3.04

3.2.1.4 Short instructions

The mean values of the evaluators' results are shown in the table below.

The text is:	Text1	Text2	Text3	Text4	All texts
Well written	2.67	2.50	2.33	3.33	2.71
Useful as a draft	2.57	2.67	2.83	3.17	2.84
Satisfactory detail	1.50	1.17	1.50	1.67	1.46
Too much / too little detail	1	1	1	1	1

3.2.1.5 General comments

The evaluators' results for the three text types under evaluation are integrated in the table below

The text is:	Overviews	Full instructions	Short instructions
Well organised	2.375	3.29	-
Well written	2.00	2.96	2.71
Useful as a draft	2.33	3.04	2.84
Satisfactory detail	1.40	-	1.46
Too much / too little detail	1.71	-	1

The table shows that Full instructions receive the best evaluation, as well written and useful as a draft for a high quality manual (the scores are 'Good'). The Overviews have obtained the lowest scores (around 'Poor'), while Short instructions have obtained medium scores. The level of details was evaluated as 'Too much detail' for Overviews and 'Too little detail' for Short instructions.

The subjects (especially the more technologically oriented among them) expressed their opinion that it is hard to evaluate the acceptability of a technical instruction text *per se,* without real knowledge of the software system it describes.

3.2.2 Czech

3.2.2.1 Overviews

The mean values of the evaluators' results are shown in the table below.

The text is:	Text1	Text2	Text3	Text4	All texts
Well organised	2.33	2.00	2.00	2.00	2.08
Well written	2.17	2.33	2.00	2.00	2.13
Useful as a draft	2.67	2.67	2.17	2.17	2.42
Satisfactory detail	1.83	1.83	1.33	1.33	1.58
Too much / too little detail	1.0	1.0	2.00	2.00	1.89

The evaluators pointed out that it is difficult to read overviews, because of the very similar appearance of all sentences and only two different connectives. They suggested generating a simple list of actions instead in order to achieve better start point for writing high quality overview.

3.2.2.2 Discrepancies

The test for discrepancies was made on 29 text examples, covering the following 9 groups of discrepancies between the Czech generated and reference texts, each subject had to one randomly chosen example for each category:

- Location ordering in a sentence
- Purpose dependent ordering
- Anaphoric nominal group realisation
- "one of the..."
- Class nouns
- Distance expressions
- Flyout menu
- Aggregation into clause complexes
- Sequence marking

The evaluation results for the 29 text examples, given by the different subjects and the mean values are shown in the table below.

Subject	**Reference is better**	**Generated is better**	**Both are equal**
1	45%	44%	11%
2	11%	67%	22%
3	22%	22%	56%
4	11%	33%	56%
5	33%	45%	22%
6	22%	33%	45%
Mean values	24%	41%	35%

Thus the generated texts were evaluated better than the original reference texts.

3.2.2.3 Full instructions

The mean values of the evaluators' results are shown in the table below.

The text is:	**Text1**	**Text2**	**Text3**	**Text4**	**All texts**
Well organised	3.17	2.50	2.83	2.33	2.71
Well written	3.33	2.83	3.50	2.83	3.12
Useful as a draft	3.00	2.67	2.82	2.67	2.79

The evaluators mentioned that some bullets in a formatting seem extraneous.

3.2.2.4 Short instructions

The mean values of the evaluators' results are shown in the table below.

The text is:	Text1	Text2	Text3	Text4	All texts
Well written	3.66	3.50	3.50	3.33	3.50
Useful as a draft	3.00	3.17	3.00	3.33	3.13
Satisfactory detail	1.33	1.83	1.83	2.00	1.75
Too much / too little detail	1.00	1.00	2.00		1.17

3.2.2.5 General comments

The evaluators' results for the three text types under evaluation are integrated in the table below.

The text is:	Overviews	Full instructions	Short instructions
Well organised	2.08	2.71	-
Well written	2.13	3.12	3.50
Useful as a draft	2.42	2.79	3.13
Satisfactory detail	1.58	-	1.75
Too much / too little detail	1.89	-	1.17

The table shows that Short instructions are on average evaluated as "Good" both from the perspective of how well they are written and how useful they are as a draft for high quality manual. Also Full instructions scores are also around "Good". On the other hand, Overviews were evaluated mostly as "Poor" in all categories. The level of detail was evaluated as 'Too much detail' for Overviews and 'Too little detail' for Short instructions.

The subjects strongly pointed out the ineffectiveness of overviews as they are now and their opinion was that for subsequent editing of a "real" documentation only goals could be reused, while the rest is only padding, which needs to be replaced anyway.

The subjects, who also participated in the evaluation of the Intermediate Prototype, mentioned that the quality of generated texts has improved since the last evaluation.

3.2.3 Russian

3.2.3.1 Overviews

The mean values of the evaluators' results are shown in the table below. The Russian Text quality evaluators rated the text of Overviews as good with respect to organisation and text quality and agreed it is a useful draft for a high-quality manual.

The text is:	**All Texts**
Well organised	3.2
Well written	2.6
Useful as a draft	3.4
Satisfactory detail	1.6
Too much / too little detail	1

The subjects pointed out to the undesirable number of repetitions of connectives in the clauses, for example, “и вы…и вы…, также…, также…также”. They suggested special formatting or aggregation of repeated pieces for the adequate presentation of the content. It was also noted that plural is better in Russian to present Actee, for example, “рисовать отрезки и дуги”.

3.2.3.2 Discrepancies

The test for discrepancies was made on 11 text examples, covering the following 5 groups of discrepancies between the Russian generated (group A) and reference (group B) texts:

1. Order of locations
2. Nominal group realisation.
3. No reference to plan elements.
4. Aggregation into clause complex.
5. Sequence marking.

The evaluation results for the 11 text examples, given by the different subjects and the mean values are shown in the table below.

Subject	A is better	B is better	A and B are equal
1	-	27.3 %	72.7 %
2	-	72.7 %	27.3 %
3	-	72.7 %	27.3 %
4	-	63.7%	37.3 %
5	9.1%	54.5 %	36.4 %
6	-	63.7%	37.3 %
Mean values	1.5%	58.56 %	39.37 %

Thus, the majority of expressions in the original reference texts were evaluated better than those used in the generated texts, though many expressions are of equal quality. The additional context is often required for making better judgements. For example, one of evaluators treated the variant “style of multiline” as being better than “style”, because the context lacks a reference to “multiline”, but in a normal text it would be redundant.

3.2.3.3 Full instructions

The mean values of the evaluators' results are shown in the table below. The Russian Text quality evaluators rated the text of Full instructions as good with respect to organisation and text quality and agreed it is a useful draft for a high-quality manual.

The text is:	Text1	Text2	Text3	Text4	All texts
Well organised	3.17	2.50	3.00	2.83	2.82
Well written	2.83	2.50	2.50	2.83	2.67
Useful as a draft	3.83	3.00	3.33	3.50	3.40

However, they noted some infelicities in formatting, text structure and word choice. As general objections to the Procedural texts the subjects pointed out that it is better to explicitly say that the main text is concerned only one of the cited operational systems (Windows or DOS / UNIX) and not only for DOS / UNIX and detach the constraint from the procedure by a sign. The methods how to get the goal, for example, specify the points of a segment, start-point and so was not described. Numbered list and not numbered list have the same indentation that is not right. Pairs Goal – Steps must be realised differently, “Рисование отрезка. Перейдите в режим …”

Also they had objections to the content of some texts:

Text1:

A sublist consisting from one point is meaningless. It will be better to use simply two steps. The concrete steps to repeat are not defined. “Сегмент” doesn’t used in the sense. Must be “участок линии” or “отрезок линии”. It seems from the text that a polyline always has only two segments, while this is not true.

Text2:

The difference between steps 1 and 3 as well as 2 and 4 are not clear. Some terminological objections: if “second point” is used, then “third point” is necessary, otherwise “end point” – “middle point”.

Text3:

“Start the arc command” and the precondition are at the same level. It would be better to use the connective “Для чего…”. Unneeded inversion of the subject position: “Привяжется дуга к точке”. It is not clear where is the line from in the first step and where the first point of the arc. The word “потом” is superficial and bad style.

Text4:

The sequence of actions is not clear. Strange sequence of substeps in the last step. ”Наконец” – bad style. It is not clear the features of what point of the multiline we describe.

3.2.3.4 Short instructions

The mean values of the evaluators' results are shown in the table below. The Russian Text quality evaluators rated the text of Short instructions as good with respect to text quality and agreed it is a useful draft for a high-quality manual.

The text is:	**Text1**	**Text2**	**Text3**	**Text4**	**All texts**
Well written	3.33	3.00	2.83	3.00	3.04
Useful as a draft	3.66	3.66	3.17	3.33	3.45
Satisfactory detail	1.83	1.5	1.33	1.67	1.58
Too much / too little detail	1	1	1	1	1

One of the general objection to short instructions is that it is not good to enumerate steps in sublists with a single step.

3.2.3.5 General comments

The evaluators’ results for the three text types under evaluation are integrated in the table below:

The text is:	**Overviews**	**Full instructions**	**Short instructions**
Well organised	3.20	3.17	-
Well written	2.60	2.87	3.33
Useful as a draft	3.40	3.83	3.66
Satisfactory detail	1.60	-	1.83
Too much / too little detail	1	-	1

The table shows that Short Instructions are best evaluated as well written, but Full instructions are best evaluated as useful as a draft for high quality manual. The Overviews

have obtained the a little lower scores, while Full instructions have obtained medium scores. The level of details was evaluated as ‘Too little detail’ for Overviews and for Short instructions.

The subjects claimed that it is hard to evaluate the acceptability of an instruction without having a system at hand and acquiring personal experience by following the instruction.

4. Evaluation of the output text quality: grammaticality

The purpose of this task was to assess the formal, grammatical correctness of the output texts. Accordingly, the evaluators were (a) native speakers of the language they judged, and (b) trained in linguistics.

The task was designed to complement acceptability evaluation (see Section 3), which aimed to assess the appropriateness of the output texts for inclusion in a CAD/CAM manual and for which the judges were trained technical communicators.

4.1 Methodology

4.1.1 The evaluators and their task

For each of the three languages, we obtained judgments from two native speakers trained in the linguistic description of their own language. In order to keep the content constant across the three languages, the six judges were asked to evaluate texts which originated from one and the same Model Set (based on MaxiModel4, cf. Section 2.1.2). The output texts are shown in Appendix 5.

Their evaluations covered all of the running-text types, i.e., Overviews, Full Instructions, Quick Reference texts (Short instructions) and Functional Descriptions. For Bulgarian and Russian, these text types were made available in two stylistic variants: Personal and Impersonal. Since Czech has two ways of expressing personal style, the Czech judges evaluated three variants: Personal Indicative, Personal Explicit and Impersonal Explicit.

The evaluators were asked to focus on grammatical correctness. However, they also made numerous unsolicited comments under the headings of appropriateness and lexical choice, which we summarise in Section 4.2.

4.1.2 The error categories

In conducting this part of the experiment, we sought a qualitative as well as a quantitative assessment of the error rate. Thus, in order to guide the evaluators, we provided them with the 16 error categories defined in Table 5, together with a simple annotation scheme for marking up the generated text for grammatical errors.

Grammatical category	Explanation	Code
Conjunction	The conjunction is wrong or absent	CO
Preposition	The preposition is wrong or absent	PR
Case	The case of the noun is wrong	CA
Agreement	The agreement between subject and verb is wrong	VG
Agreement	The agreement between noun and modifier is wrong	NG
Tense	The tense of the verb is wrong	TE
Aspect	The aspect of the verb is wrong	AS
Person	The person of the verb is wrong	PE
Number	The number of the verb is wrong	NU
Morphology	The inflexion of the constituent is absent	MO
Determiner (Bulgarian)	The determiner marker is wrong or absent	DT
Subject	The subject of the verb is absent when it should be present (i.e. not dropped)	SA
Subject	The subject of the verb is present when it should be absent (i.e. dropped)	SP
Object	The object of the verb is absent	OB
Indirect object	The indirect object of the verb is absent	IO
Word order	The ordering of constituents is wrong	WO

Table 5: Grammatical error categories and codes

Judges were invited to add categories if the need arose, but the original suggestions proved adequate for their purposes.

4.2 Results

In this section, we refer to our six judges where necessary as Bu1, Bu2, Cz1, Cz2, Ru1 and Ru2, in order to bring attention to interesting convergences or divergences of opinion.

4.2.1 Grammaticality—observed error types

We disregarded some perceived errors which we attributed to the judges' lack of familiarity with the CAD/CAM domain. For example, JRu1 preferred 'line' to 'multiline', whereas 'multiline' is a well-defined concept in the domain. And both Russian judges suggested 'add the style of the drawing' instead of 'add the style to the drawing', without knowing that in this domain 'style' is an individual component of a drawing rather than an attribute of the drawing as a whole.

Such misconceptions are inevitable when linguists are asked to judge technical texts in an unfamiliar domain and without access to the artefact that the text describes. It is for this reason that we elicited comments on text quality from our usability and technical author judges also (see Sections 2 and 3).

4.2.1.1 Bulgarian

No grammatical errors were reported for the Bulgarian texts.

4.2.1.2 Czech

The two Czech evaluators reported only one (the same) error. This was an error or word order (WO):

Generated:	Zmizí dialogový panel Vlastnosti multičár.
Gloss:	Disappears dialogue box Properties multiline.
Correction:	Dialogový panel Vlastnosti multičár zmizí.
Gloss:	Dialogue box Properties multiline disappears.

This particular sentence occurred twice in two of the texts given to evaluators. Subject Cz2 considered this to be possibly best qualified as a stylistic rather than strictly grammatical error: while this utterance is grammatical in some contexts (e.g., the two occasions on which it appeared in the third text), the ordering in the identified cases was inappropriate since the dialogue pane (dialogový panel) had already been mentioned in the immediately preceding context.

4.2.1.3 Russian

Very few errors were identified, and only in the word order (WO) category. Subject Ru2 considered these to be stylistic rather than grammatical errors.

1. **Word order—purpose clauses:** In 3 instances out of a total 6 occurrences, subject Ru1 ordered the subordinate clause introduced by *чтобы* (‘in order to’) before the main clause. Ru2 made no such corrections.

   We could discern no reason— for example, on the grounds of information structure— for correcting just these 3 instances.

2. **Word order—location expressions:** Evaluator Ru2 preferred location expressions like ‘in the X menu’ to be ordered before the action, e.g. ‘select’. Evaluator Ru1 preferred locations to be ordered relative to one another on a *ground before figure* principle, e.g. ‘on the T toolbar in the P palette’ rather than ‘in the P palette on the T toolbar’.

4.2.2 Other errors identified

Here we record unsolicited but helpful feedback from the judges on stylistic issues. In contrast to the well-defined decisions about grammatical correctness, these comments reflect judgments—perhaps personal—of acceptability and appropriateness.

4.2.2.1 Bulgarian

Layout: One judge (Bu1) complained about the number of apparently spurious bullets appearing in the texts. He also pointed to places where the labels "Windows" and "Dos/Unix", although appearing on separate lines, were not correctly rendered to signal their status as headers— presumably means of a distinguishing font or point size.

4.2.2.2 Czech

The only additional comment was positive:

> "From the viewpoint of stylistics, I really appreciated the consistency in the choice of person and mood within each of the texts. If these choices were not consistent, i.e., if person or mood would be changing within a text, it would be disturbing." (judge Cz2)

4.2.2.3 Russian

1. **Tense (TE):** Judge Ru1 expressed a preference for the future rather than the present tense in expressing side-effects in the Personal style, where the user actions are expressed by imperative mood. In the Impersonal style, where the user actions are expressed by indicative mood, the present tense—as generated—is appropriate.
2. **Lexical choice—temporal markers:** Both judges preferred *затем* to *потом* ('next'). One judge preferred *в конце* to *наконец* ('finally').
3. **Lexical choice—•••••:** Judge Ru1 considered ••••• ('menu item') in such contexts as *выберите* ••••• *MultilineStyles* ('select …') to be redundant.
4. However, the same judge did not object to inclusion of *кнопка* ('button') in analagous contexts, i.e. before labels of interface objects. The role of these words is syntactic rather than semantic—they serve to carry the grammatical case marker, which cannot be applied to a literal label. We conclude, then, that the real objection concerns the choice of lexeme.
5. **Lexical choice—document headings/parts:** Both judges objected to *справка* ••••••• ('object reference'), suggesting instead *список* ('list') or *функциональность* ('functionality').
6. **Layout:** Judge Ru2 commented on the inconsistencies in the levels of numbering and indentation, and on redundant list bullets.

4.3 Conclusions

4.3.1 Grammatical errors

Almost no grammatical errors were identified by the judges: none at all for Bulgarian, only one for Czech, and four for Russian. All of the identified grammatical errors were classified as ones of word order (WO), a well-known problem for Eastern European languages. While some of identified problems of word order in the generated texts were clearly grammatical, others were less clearly so; in at least two cases, judges thought they might be stylistic.

We take this result as strong validation of the quality of the grammars we have developed for these three languages. To our knowledge, the AGILE project is the first successful

attempt at developing computational grammars for the generation of Bulgarian, Czech or Russian.

4.3.2 Other errors

Once again, there were few errors identified. Of these most were stylistic. For example, both Russian judges expressed a preference for the Personal style over the Impersonal style for Full Instructions and Quick Reference (short instruction texts), the only two text types subject to stylistic variation in the output they reviewed. With the exception of layout, the remaining errors are probably best classified as (as one judge put it) instances of "émigré speak". In other words, they are not errors in the true sense, and do not seriously affect the comprehensibility of the generated text; not surprisingly, there was little agreement between judges on their identification.

4.3.3 Further remarks

In carrying out this evaluation, we had planned to support our results on grammaticality with strong quantitative measures. We had hoped to use the *generation accuracy* measure proposed by Bangalore and his colleagues for this purpose, but our close examination of it showed it to be unsuitable for measuring the output of a text generation system (although it can be usefully applied to another task related to text generation: symbolic authoring of domain models, as discussed in Section 2.2.1.2).

If the *generation accuracy metric* is to be taken as meaningful, it must provide a firm enough basis for one to judge the quality of a piece of generated text against a gold standard (i.e., to determine whether the text that is generated is in some sense good enough), and to make comparisons between texts generated by different systems. It also needs to be able to distinguish between *mistakes* and *really bad mistakes*. For the reasons given above, we find the metric provided by Bangalore *et al.* to be too crude, too ill-defined and too misleading for us to adopt for evaluating the grammatical accuracy of an NLG system.

For the purpose of our present evaluation, it would appear that even if the generation accuracy metric were in fact appropriate to the task, it would not provide us with any more information than we already have, given the extremely small number of grammatical errors encountered in the AGILE-generated texts. Other alternatives, including ones also suggested by Bangalore *et al* (e.g., based on dependency trees) were also not appropriate to the AGILE system. Luckily, our results were such that they were able to stand on their own, without qualitative measures to support them. We plan to explore alternative ways of measuring NLG output, but this remains future work beyond the scope of AGILE.

5. Linguistic coverage evaluation

One of the primary goals of the AGILE project has been the development of lexico-grammatical resources suitable for multilingual generation:

> "The overall goal of AGILE is to make available a *generic* set of tools and *linguistic resources* for generating Czech, Bulgarian and Russian: no such resources are currently available." (Agile, 1997; our emphasis).

The AGILE project thus represents the first attempt ever at a computational account of Bulgarian, Czech and Russian for the purpose of natural language generation. In evaluation of our resources with respect to their coverage of linguistic phenomena, we addressed two types of questions:

1. what is the list of phenomena which are covered by the resources, and how can we control the output?
2. how do our resources cover phenomena that really exist in instructional texts?

The answer to the first question involves development of example sets for the regions of the resources. The answer to the second question involves an analysis of the annotated AutoCAD corpora with respect to phenomena which are implemented in the grammar (the annotated AutoCAD corpora were developed as the output of WP3-1, see the CORP deliverable). In addition to the annotated corpora, we consider phenomena, which do not occur in those texts, but are important for software manuals or instructional texts in general. We do not discuss grammatical phenomena in depth in this deliverable (for extended discussions about the lexico-grammar resources cf. SPEC3-IMPL3, SPEC2 and IMPL2). Examples list only phenomena and the input SPL expressions required for those phenomena. Discussions about the grammatical structure of the output also pertain to the grammar specification and implementation deliverables.

5.1 Coverage in regions of the lexico-grammars

As specified in the SPEC3/IMPL3 deliverable, development of lexico-grammar resources has covered the following regions:

Clause rank

- **nuclear transitivity**: realization of Upper Model processes of dispositive, creative and non-directed material actions, realization of existential relations, as well as of several domain-specific types, like SNAP, SWITCH-MODE and SAVE;
- **circumstances**: different types of spatial relations;
- **aspect** choice;
- **mood** (imperative and declarative, plus conditional for Czech)
- **diathesis (voice)** (active, medio-passive and passive)
- realization of some **modality** options, notably possibility, plus necessity and ability for Czech;
- **word order** constraints;

- **textual conjunction;**
- **hypotactic clause complexity** (restricted to purpose, means, logical-condition relations);
- **paratactic clause complexity** (conjunction and disjunction);
- **Subject-Finite agreement**
- **Selection of case** for functional elements that are realized by case-bearing grammatical categories (e.g., nominal cases for NGs realizing participants at clause rank)
- **Subject Dropping** in Czech and Bulgarian

Nominal group rank

- **agreement** (between Deictic, Quantity, Status and Thing elements);
- selection of **lexical gender** for lexical items;
- **determination**;
- expressions with cardinal and ordinal numbers and **quantification** constructions of the type "*one of X*"
- **nominalization** of actions;
- **nominal group complexity**: conjunction and disjunction, generalized possessive relations.

Prepositional phrase rank

- language-specific **extensions of prepositional phrase systems** (case government of prepositions)

SFG-based generation involves two subtasks (cf. Teich, 1999):

- task 1: to interpret a semantic input expression in terms of the grammar available;
- task 2: to spell out the syntactic constraints of the language in which an utterance is to be generated.

(Yang et *al*., 1991) characterize task 1 as involving ”deciphering the goals given by the speaker program and determining how they can be realized in language” (Yang *et al*., 1991, p.207) and task 2 as ensuring ”that the end product adheres to the syntactic rules of the language” (Yang *et al*., 1991, p.207). The first is the functional aspect of tactical generation; the latter is the syntactic aspect. The generation process starts with an input expression in the form of SPL (Sentence Planning Language) (Kasper, 1989). SPL expressions provide all semantic information required for task 1, while constraints imposed by task 2 are realized within the grammar. Thus, the phenomena-specific sections below either accompany linguistic phenomena with semantic input conditions that are required for their choice (task 1) or present the output produced by the grammar without specification of semantic input conditions (task 2). In our resources task 2 concern phenomena of agreement (subject-finite and agreement within the nominal group), selection of case for nominal groups, subject dropping, selection of lexical gender for lexical items and case government of prepositions.

5.1.1 Transitivity

In terms of the nuclear transitivity types defined in the Upper Model (Bateman, et al, 1990), our lexico-grammar resources deal with: directed-actions, nondirected-actions (both are material processes), and some types of relational processes. Thus, mental and verbal processes have not been implemented.

Directed-actions necessarily involve an external causer that brings about the action defined by the process and medium combination. The presence of a second partipant requires that they have both an actor role and an actee role. Thus, they are always transitive:

(1) Directed Actions

En: *The user draws a line.*

Ru: *Пользователь рисует линию.*

Bg: *Потребителят чертае линия.*

Cz: *Uživatel kreslí úsečku.*

In the Czech grammar, some Directed actions need to be realized using an intransitive verb and a prepositional phrase realising the actee role. This is required by the government pattern of the particular verbs which do not take a direct complement in the accusative case.

(2) Directed Actions with intransitive verbs

En: *The user clicks on a button.*

Cz: *Uživatel klepne na tlačítko.*
User-nom clicks on button-acc

Such cases do not occur in the Bulgarian and Russian grammars.

The process type is either directly expressed in an SPL:

```
(S / DIRECTED-ACTION :LEX VERB
          :ACTOR X
          :ACTEE Y)
```

or is inherited by the respective DM concept:

```
(S / DM::DRAW
          :ACTOR X
          :ACTEE Y)
```

Non-directed-actions are those material actions which require no external causation in addition to the combination of process and medium. Such processes only have an actor role, and are intransitive in our grammars[6]:

(3) Non-directed Actions

En: *The window disappears from the screen.*

Ru: *Окно исчезнет с экрана.*

Bg: *Прозорецът изчезва от екрана.*

[6] The Range complement of non-directed processes possible in the UM is not implemented in our grammars.

Cz: *Okno zmizí z obrazovky.*

The process type is again either directly expressed in an SPL:

```
(S / NONDIRECTED-ACTION :LEX VERB
          :ACTOR X)
```

or is inherited by the respective DM concept:

```
(S / DM::DISAPPEAR
          :ACTOR X)
```

Processes can be expressed by nominal groups (for example, in headings or clause complexes). The actee of a nominalized directed action is expressed using the genitive case in Czech and Russian and a prepositional phrase in Bulgarian:

(4) Nominalized Actions

En: *Drawing of an arc.*

Ru: *Рисование дуги.*
drawing arc-gen

Bg: *Чертане на дъга*
drawing of arc

Cz: *Kreslení oblouku.*
drawing arc-gen

In headings, this choice is controlled by means of the following SPL specification:

```
:EXIST-SPEECH-ACT-Q NOSPEECHACT
```

A process that appears in the position of a participant, for example, actor, actee or instrument, is also expressed by a nominal group:

(5) Actions as participants

En: *Issuing the DRAW command draws a line.*

Ru: *Запуск команды DRAW рисует линию.*
starting command-gen DRAW draw-3sg line-acc.

Bg: *Стартиране на командата DRAW чертае линия.*
start-nom. of command-det. DRAW draw-3sg line

Cz: *Zadání příkazu DRAW kreslí čáru.*
issuing-nom command-gen DRAW draw-3sg line-acc
Zadáním příkazu DRAW se kreslí čára.
issuing-ins command-gen DRAW refl draw-3sg line-acc

The range of lexical items for realization of a concept is specified in lexical annotations for the resource set as follows[7]:

```
(annotate-concept DM::DRAW :lex-items
    (:russian narisovatj risovatj risovanie
     :Czech-991105 kreslit nakreslit kresleni2 nakresleni2
     :bulgarian NACHERTAJA CHERTAJA CHERTAENE))
```

Thus, the complete SPL expression for (4) is:

[7] In Czech, numbers are used to express Czech diacritics in ASCII (e2 means é, e3 means ě)

```
(S / DM::DRAW
     :EXIST-SPEECH-ACT-Q NOSPEECHACT
     :ACTEE (P / DM::ARC))
```

Within the circumstantial transitivity, mostly spatial relations have been implemented. The Upper model distinguishes between static and dynamic spatial processes and dimensionality of objects. Within dynamic processes, the distinction is between movement towards an object (destination) and from an object (source).

(6) Static circumstances

En: *Choose an item in a menu.*

Ru: *Выберите пункт в меню.*

Bg: *Изберете елемент в меню.*

Cz: *Zvolte položku v nabídce.*

The circumstantial modifier is expressed in SPL by the following statement:

```
:SPATIAL-LOCATING (D / DM::MENU)
```

(7) Static two-dimensional circumstances

En: *Click a button on a toolbar.*

Ru: *Нажмите кнопку на панели.*

Bg: *Щракнете бутон на функционален ред.*

Cz: *Klikněte na tlačítko na panelu.*

The dimension of the circumstantial modifier is expressed in SPL by the following statement:

```
:SPATIAL-LOCATING (D / (ONE-OR-TWO-D-LOCATION DM::TOOLBAR))
```

(8) Static containment circumstances

En: *Draw a point inside a region.*

Ru: *Нарисуйте точку внутри области.*

Bg: *Начертайте точка в област.*

Cz: *Nakreslete bod uvnitř oblasti.*

This type of circumstantial modifier can be expressed in SPL as follows (only tested in the Russian grammar):

```
(S / DM::DRAW :SPEECHACT IMPERATIVE
     :RELATIONS
     (R / SPATIAL-LOCATING
          :DOMAIN S  :RANGE (D / DM::REGION)
          :EXPLICIT-CONTAINMENT-Q EXPLICIT)
      :ACTOR (hearer / DM::USER)
     :ACTEE (K / DM::POINT))
```

(9) Static noncontainment circumstances

En: *Draw a point outside a region.*

Ru: *Нарисуйте точку снаружи от области.*

Bg: *Начертайте точка извън областта.*

Cz: *Nakreslete bod mimo oblast.*

This type of circumstantial modifier can be expressed in SPL as follows (only tested in the Russian grammar):

```
(S / DM::DRAW :SPEECHACT IMPERATIVE
     :RELATIONS
     (R / SPATIAL-LOCATING
          :DOMAIN S  :RANGE (D / DM::REGION)
          :CONTAINMENT-Q NONCONTAINMENT)
      :ACTOR (hearer / DM::USER)
     :ACTEE (K / DM::POINT))
```

(10) Destination

En: *Switch to a mode…*

Ru: *Перейдите в режим…*

Bg: *Превключете към режим...*

Cz: *Přepněte do režimu…*

This type of circumstantial modifier is expressed in SPL by the following statement:

```
:DESTINATION (D / DM::MODE)
```

In a particular language and for a particular verb, the destination may also be realised in a way that depends on the government pattern of the verb:

(11) Destination with a government pattern

En: *Attach an arc to a point.*

Ru: *привяжите дугу к точке.*

Bg: *прикачване на дъга към точка*

Cz: *Přichyťte oblouk k čáře.*

This can be controlled by a specific DM type, for example, the concept SNAP:

```
(E / DM::SNAP
     :SPEECHACT IMPERATIVE
      :ACTOR (hearer / DM::USER)
     :ACTEE (P / DM::ARC)
     :DESTINATION (D / DM::POINT))
```

(12) Source

En: *Choose an item from a menu.*

Ru: *Выберите пункт из списка.*

Bg: *Изберете опция от меню.*

Cz: *Zvolte položku z menu.*

This type of circumstantial modifier is expressed in SPL by the following statement:

```
:SOURCE (D / DM::LIST)
```

(13) Two-dimensional source

En: *The window disappears from the screen.*

Ru: *Окно исчезнет с экрана.*

Bg: *Прозорецът изчезва от екрана.*

Cz: *Panel zmizí z obrazovky.*

The dimension of the circumstantial modifier is expressed in SPL by the following statement:

```
:SOURCE (C1 / DM::SCREEN)
```

The following two types of spatial specifications are specific for implementation of the Russian lexicogrammar:

(14) "set of sources" specification

En: *to draw an arc using three points.*

Ru: *нарисовать дугу по трем точкам.*
to draw arc-acc along three points-dat.

This type of circumstantial modifier is expressed in SPL by the following statement:

```
:PARALLEL-EXTENT (D / DM::POINTS)
```

(15) "with respect to" specification

En: *to specify an angle with respect to a line.*

Ru: *задать угол по отношению к линии.*
to specify angle-acc with respect to line-gen.

This type of circumstantial modifier is expressed in SPL by the following statement:

```
:ADDITIVE (D / DM::LINE)
```

5.1.2 Aspect

The aspect of independent clauses in Slavic languages depends on the following considerations (cf. the SPEC3/IMPL3 deliverable):

- repetition of an action leads to the choice of the imperfective aspect;
- relational processes (being static) lack inherent temporal structure, so they are always expressed in the imperfective aspect
- focusing on the result of a process leads to the choice of the perfective aspect; focusing on activity involved in a process leads to the imperfective aspect.

The perfective aspect is the default choice, the choice of the imperfective aspect is controlled using two inquiry-response pairs:

```
:REPEATED-Q REPEATED
:NONSTATIC-PROCESS-ASPECT-Q ACTIVITY-HIGHLIGHTED
```

(16) Perfective aspect

En: *Draw a line segment.*

Ru: *Нарисуйте отрезок прямой.*

Cz: *Nakreslete rovný segment.*

Bg*: Начертайте отсечка*

```
(S / DM::DRAW :SPEECHACT IMPERATIVE
       :NONSTATIC-PROCESS-ASPECT-Q RESULT-HIGHLIGHTED
        :ACTOR (hearer / DM::USER )
       :ACTEE (K / DM::LINE-SEGMENT))
```

(17) Imperfective aspect

En: *A line segment is being drawn.*

Ru: *Рисуется отрезок прямой.*

Cz: *Kreslí se rovný segment.*

Bg*: Чертае се отсечка.*

```
(S / DM::DRAW
       :CAUSATIVE-MENTION-Q WITHHOLD
       :PREFER-MENTION-AGENT-Q WITHHOLD
       :NONSTATIC-PROCESS-ASPECT-Q ACTIVITY-HIGHLIGHTED
        :ACTOR (hearer / DM::USER )
       :ACTEE (K / DM::LINE-SEGMENT))
```

In Russian, a noun expressing a process typically has one aspect, so the choice is not controlled. In Czech, the choice between perfective and imperfective nominalizations (when available) is controlled in the same way as for verbs..

5.1.3 Mood

According to (Halliday, 1985), the clause is organized as an interactive event between the speaker and the audience. The basic options at the clause rank are: giving information (statements), demanding information (questions) and demanding goods&services (commands). Demanding information (interrogative clauses) at the clause rank as well as clausettes (incomplete clauses, like greetings *Happy New Year*) have not been implemented.

In expression of commands, all the three languages can use the imperative mood. This choice is controlled in SPL expressions by specifying the following information:

```
:SPEECHACT IMPERATIVE
```

There also are different realizations of the imperative depending on whether the imperative is expressed politely There also are different realizations of the imperative depending on whether the imperative is expressed politely (a respect is expressed towards the addressee, this expression is typical for the formal registers) or nonpolitely (the addressee is equal to the author in the social status, this expression is typical for the informal communication). The first option is default. In Bulgarian and Czech, the second option is not modeled in the grammars. In Russian, the choice of the non-polite (informal) imperative is controlled in SPL expressions by specifying the following information:

```
:RESPECTED-ADDRESSEE-Q EQUAL
```

The distinction is valid for all three languages:

(18) Imperative polite

En: *Draw a line.*

Ru: *Нарисуйте линию.*
Draw-imper-plur line-acc

Bg: *Начертайте линия.*
draw-imper-plur line

Cz: *Nakreslete úsečku.*
Draw-2pl line-acc

(19) Imperative informal (not implemented in Bulgarian and Czech)

En: *Draw a line.*

Ru: *Нарисуй линию.*
Draw-imper-sg line-acc

Bg: *Начертай линия.*
Draw-imper-sg line

Cz: *Nakresli úsečku.*
Draw-2sg line-acc

Also, commands can be expressed by means of infinitive clauses in Czech and Russian:

(20) Infinitive

En: *To draw a line.*

Ru: *Нарисовать линию.*

Cz: *Nakreslit úsečku.*

This choice is controlled in SPL expressions by specifying the following information:

```
:PROPOSAL-Q PROPOSAL :REPORT-Q REPORT :ENTIRENESS-Q PARTIAL
```

An SPL macro is provided for this purpose:

```
:SPEECHACT INFINITIVE
```

In terms of expression of statements, the linguistic resources for all languages provide possibilities to express the indicative mood (other mood options, like the subjunctive in Russian are not considered). The three persons involved in the exchange in the indicative mood can be expressed as follows:

(21) first person singular (not implemented in Bulgarian)

En: *I draw*

Ru: *Я рисую.*

Cz: *(Já) kreslím.*
(I) draw-1sg

```
:ACTOR (SPEAKER / PERSON )
```

(22) second person singular (not implemented in Bulgarian)

En: *You draw*

Ru: *Ты рисуешь.*
You-sg draw-2sg

```
:ACTOR (HEARER / PERSON :RESPECTED-ADDRESSEE-Q EQUAL)
```

Cz: *(Ty) Kreslíš.*
(You) *Draw-2sg*

```
(|a19| / DM::DRAW
 :SPEECHACT assertion
 :MEMBER-SET-Q NOTINCLUDED
 :MULTIPLICITY-Q UNITARY
 :ACTOR (hearer / DM::USER)
 :ACTEE  (|a21| / DM:: LINE))
```

(23) second person polite/plural

En: *You draw*

Ru: *Вы рисуете.*
You-pl draw-2pl

```
:ACTOR (HEARER / PERSON :RESPECTED-ADDRESSEE-Q RESPECTED)
```

Bg: *Вие чертаете.*
You-pl draw-2pl

Cz: *(Vy) kreslíte.*
(You) Draw-2pl

```
(|a19| / DM::DRAW
 :SPEECHACT assertion
 :MEMBER-SET-Q NOTINCLUDED
 :MULTIPLICITY-Q MULTIPLE
 :ACTEE  (|a21| / DM:: LINE))
```

(24) third person

En: *The user draws*

Ru: *Пользователь рисует.*

Bg: *Потребителят чертае.*

Cz: *Uživatel kreslí.*
User draw-3sg

```
:ACTOR (U / DM::USER)
```

5.1.4 Voice

(25) Active voice

En: *Draw a line segment.*

Ru: *Нарисуйте отрезок прямой.*

Cz: *Nakreslete rovný segment.*

Bg*: Начертайте отсечка*

```
(S / DM::DRAW :SPEECHACT IMPERATIVE
        :NONSTATIC-PROCESS-ASPECT-Q RESULT-HIGHLIGHTED
        :ACTEE (K / DM::LINE-SEGMENT))
```

(26) Medio-passive voice

En: *A line-segment is being drawn.*

Ru: *Рисуется отрезок прямой.*

Cz: *Kreslí se rovný segment.*

Bg*: Чертае се отсечка.*

```
(S / DM::DRAW
          :CAUSATIVE-MENTION-Q WITHHOLD
          :NONSTATIC-PROCESS-ASPECT-Q ACTIVITY-HIGHLIGHTED
          :ACTEE (K / DM::LINE-SEGMENT))
```

5.1.5 Modality

Modality (possibility, ability, necessity, etc) could be expressed as linguistic function in all three languages either by a modal auxiliary or by some type of complex verbal group.

(27) Possibility

En: *The program can draw lines.*

Ru: *программа может рисовать линии*
program can-3sg draw-inf lines-acc

Cz: *program může kreslit čáry*
program can/may-3sg draw-inf lines-acc

Bg: *Програмата може да чертае линия.*
program-det can/may-3sg draw3sg line

```
(S / DM::DRAW
      :MODAL-PROPERTY-ASCRIPTION     GENERAL-POSSIBILITY
      :ACTOR (hearer / DM::USER  :IDENTIFIABILITY-Q IDENTIFIABLE)
      :ACTEE (|a2| / DM::LINE))
```

(28) Ability (not implemented in Bulgarian)

En: *The program can draw lines.*

Cz: *program umí kreslit čáry*
program can-3sg draw-inf lines-acc

Bg: *Програмата може да чертае линия*
program-det. can-3sg draw-3sg line

Ru: *Програма может рисовать линии*
program can-3sg draw-inf lines-acc

```
(S / DM::DRAW
      :MODAL-PROPERTY-ASCRIPTION     ABILITY
      :ACTOR (A / DM::PROGRAME :IDENTIFIABILITY-Q IDENTIFIABLE)
      :ACTEE (|a2| / DM::LINE))
```

(29) Necessity (not implemented in Bulgarian)

En: *The program can draw lines.*

Cz: *program musí kreslit čáry*
program must-3sg draw-inf lines-acc

Bg: *Програмата трябва да чертае линия.*
program-det. must-3sg draw-3sg line

Ru: *Програма должна рисовать линии.*
program must-3sg draw-inf lines-acc

```
(S / DM::DRAW
       :MODAL-PROPERTY-ASCRIPTION     NECESSITY
       :ACTOR (A / DM::PROGRAME  :IDENTIFIABILITY-Q IDENTIFIABLE)
       :ACTEE (|a2| / DM::LINE))
```

(30) Conation: allow

En: *The program allows to draw lines.*

Ru: *программа позволяет рисовать линии*
program enables drawing-acc lines-gen

Cz: *program umožňuje kreslení čar*
program enables drawing-acc lines-gen

Bg: *Програмата позволява да чертаете линия*
program-det enables to draw-2pl line

```
(S / DM::DRAW
       :CONATION-ASPECT     POTENTIAL
       :ACTOR (A / DM::PROGRAME  :IDENTIFIABILITY-Q IDENTIFIABLE)
       :ACTEE (|a2| / DM::LINE))
```

(31) Trial (implemented only in Russian)

En: *The program tries to draw lines.*

Ru: *программа пытается рисовать линии*
program try-3sg-refl draw-inf lines-acc

```
(S / DM::DRAW
       :CONATION-ASPECT     CONATIVE
       :ACTOR (A / DM::PROGRAME  :IDENTIFIABILITY-Q IDENTIFIABLE)
       :ACTEE (|a2| / DM::LINE))
```

(32) Success (implemented only in Russian)

En: *The program succeeds in drawing lines.*

Ru: *программа сумеет рисовать линии*
program succeed-3sg draw-inf lines-acc

```
(S / DM::DRAW
       :CONATION-ASPECT     REUSSIVE
       :ACTOR (A / DM::PROGRAME  :IDENTIFIABILITY-Q IDENTIFIABLE)
       :ACTEE (|a2| / DM::LINE))
```

5.1.6 Word order

In the Czech grammar, ordering at the clause level has been implemented in such a way that word order reflects information structure. The informational status of individual semantic units is reflected in the SPLs by stating whether they are contextually bound or contextually non-bound.

```
contextual-boundness yes
contextual-boundness no
```

Only contextual boundness needs to be stated explicitly, non-boundness is the default. The mutual ordering of contextually bound and non-bound entities follows the (default)

systemic ordering, while the cluster of contextually bound entities precedes the clusters of the non-bound ones. The word order variation that can be achieved by this mechanism is demonstrated below.

(33) *A/the file opens with the OPEN command*

Cz: *Otevírá se příkazem* Open *soubor.*
opens refl command-ins *Open* file

```
(S1 / directed-action :lex otevr3i2t
 :speechact assertion :prefer-mention-agent-q withhold
 :ACTOR (hearer / Person )
 :ACTEE (file / object :lex soubor )
 :Instrumental (command / object :lex pr3i2kaz
   :class-ascription (label / object :name GUI-OPEN)))
```

Cz: *Soubor se otevírá příkazem* Open.
file refl opens command-ins *Open*

```
(S1 / directed-action :lex otevr3i2t
  :speechact assertion :prefer-mention-agent-q withhold
  :ACTOR (hearer / Person )
  :ACTEE (file / object :lex soubor :contextual-boundness yes)
  :Instrumental (command / object :lex pr3i2kaz
    :class-ascription (label / object :name GUI-OPEN)))
```

Cz**:** *Příkazem* Open *se otevírá soubor.*
command-ins *Open* refl opens file

```
(S1 / directed-action :lex otevr3i2t
  :speechact assertion :prefer-mention-agent-q withhold
  :ACTOR (hearer / Person )
  :ACTEE (file / object :lex soubor )
  :Instrumental (command / object :lex pr3i2kaz
    :contextual-boundness yes
    :class-ascription (label / object :name GUI-OPEN)))
```

Cz: *Příkazem* Open *se soubor otevírá.*
command-ins *Open* refl file opens
(S1 / directed-action :lex otevr3i2t

```
  :speechact assertion :prefer-mention-agent-q withhold
  :ACTOR (hearer / Person )
  :ACTEE (file / object :lex soubor :contextual-boundness yes)
  :Instrumental (command / object :lex pr3i2kaz
    :contextual-boundness yes
    :class-ascription (label / object :name GUI-OPEN)))
```

Cz: *Soubor se příkazem* Open *otevírá.*
file refl command-ins *Open* opens

```
(S1 / directed-action :lex otevr3i2t
  :speechact assertion :prefer-mention-agent-q withhold
  :ACTOR (hearer / Person )
  :ACTEE (file / object :lex soubor :contextual-boundness yes)
  :Instrumental (command / object :lex pr3i2kaz
    :contextual-boundness yes
    :class-ascription (label / object :name GUI-OPEN)
    :theme file))
```

5.1.7 Textual conjunctions

(34) *First define a multiline.*

Cz: *Nejprve/Nakonec/Nyní definujte multičáru.*
First/Last/Now define-imp *multiline*-acc

Bu: *Отначало/ Накрая/След това дефинирайте мултилинията.*
First/Last/Now define-imp *multiline*

Ru: *Сначала/Наконец/теперь задайте мультилинию.*
First/Last/Now define-imp *multiline*-acc

The SPL statement:

```
:CONJUNCTIVE SEQUENCE-FIRST/-LAST/-OTHER/-RELATIVE
```

The SPL statements shown above use macros. Their definitions are shown in Figure 9.

```
Conjunctive-first =
         (:conjunctive-relation-q conjunctive
          :conjunctive-relation-id (?rr / rhetorical-relation)
          :process-regulated-q processregulated
          :necessity-q nonecessity
          :sequence-q sequence
          :absolute-position-q absolute
          :extremal-position-q extremal-first)
Conjunctive-last =
        (:conjunctive-relation-q conjunctive
         :conjunctive-relation-id (?rr / rhetorical-relation)
         :process-regulated-q processregulated
         :necessity-q nonecessity
         :sequence-q sequence
         :absolute-position-q absolute
         :extremal-position-q extremal-last)
Conjunctive-other =
         (:conjunctive sequence-other
          :conjunctive-relation-q conjunctive
          :conjunctive-relation-id (?rr / rhetorical-relation)
          :process-regulated-q processregulated
          :necessity-q nonecessity
          :sequence-q sequence
          :absolute-position-q absolute
          :extremal-position-q nonextremal)
Conjunctive-relative =
         (:conjunctive-relation-q conjunctive
          :conjunctive-relation-id (?rr / rhetorical-relation)
          :process-regulated-q processregulated
          :necessity-q nonecessity
          :sequence-q sequence
          :absolute-position-q notabsolute
          :relative-position-q immediate)
```

Figure 9: SPL macros for textual conjunctives

5.1.8 Clause complexity

5.1.8.1 Paratactic

(35) **paratactic extension: positive addition (conjunction)**

En: *Specify a point and press a key.*

Ru: *Укажите точку и нажмите клавишу.*
Specify-2pl point and press-2pl key-acc.

Cz: *Určete bod a stiskněte klávesu.*
Specify-2pl point-acc and press-2pl key-acc.

Bg: *Въведете точка и натиснете клавиш.*
Specify-2pl point and press-2pl key.

```
(c / conjunction
    :Domain
    (S1 / DM::SPECIFY-COMPONENT
        :SPEECHACT IMPERATIVE
        :ACTOR (hearer / DM::USER)
        :ACTEE (P1 / DM::POINT
                              :IDENTIFIABILITY-Q notIDENTIFIABLE)
    :Range
    (S2 / DM::PRESS
        :ACTOR (hearer / DM::USER)
        :ACTEE (A / DM::key
             :IDENTIFIABILITY-Q NOTIDENTIFIABLE))))
```

(36) **paratactic extension: positive variation (disjunction)**

En: *Specify a point or press a key.*

Bg: *Въведете точка или натиснете клавиш.*
Specify-2pl point or press-2pl key.

Cz: *Určete bod nebo stiskněte klávesu.*
Specify-2pl point-acc or press-2pl Return.

Ru: *Укажите точку или нажмите клавишу.*

```
(c / disjunction
    :Domain
    (S1 / DM::SPECIFY-COMPONENT
        :SPEECHACT IMPERATIVE
        :ACTOR (hearer / DM::USER)
        :ACTEE (P1 / OBJECT :LEX bod
                             :IDENTIFIABILITY-Q notIDENTIFIABLE)
    :Range
    (S2 / DM::PRESS
        :ACTOR (hearer / DM::USER)
        :ACTEE (A / DM::key
              :IDENTIFIABILITY-Q NOTIDENTIFIABLE))))
```

(37) **paratactic enhancement: temporal circumstantial**

En: *Choose OK and then close the window.*

Bg: *Изберете OK и след това затворете прозорец*
Choose-2pl OK and after that close-2pl window

Cz: *Vyberte OK a potom zavřete okno.*
Choose-2pl OK and then close-2pl window-acc

Ru: *Нажмите OK и затем закройте окно*
Choose-2pl OK and then close-2pl window-acc

```
 (c / RST-sequence
    :Domain
    (S1 / DM::CHOOSE
        :SPEECHACT IMPERATIVE
        :ACTOR (hearer / DM::USER)
```

```
            :ACTEE (P1 / DM::gui-ok
                                :IDENTIFIABILITY-Q IDENTIFIABLE ))
        :Range
        (S2 / DM::CLOSE
            :ACTOR (hearer / DM::USER)
            :ACTEE (A / DM::window
                    :IDENTIFIABILITY-Q IDENTIFIABLE))))
```

5.1.8.2 Hypotactic

(38) **hypotactic enhancement: purpose circumstantial**

En: *Press a key to end the line.*

Cz: (dependent: nominalization)
Stiskněte klávesu pro ukončení úsečky.
Press-2pl key-acc for ending-acc line-gen

Ru: *Нажмите клавишу, чтобы завершить линию.*
Press-2pl key , in-order-to end line-gen

```
(R / RST-PURPOSE
   :DOMAIN
   (P / DM::PRESS :SPEECHACT IMPERATIVE
      :ACTOR (hearer / DM::USER)
      :ACTEE (A1 / DM::key))
   :RANGE
   (E / DM::END-LINE
      :ACTOR (hearer / DM::USER)
      :ACTEE (M1 / DM::LINE)))
```

Cz: (dependent: finite)
Stiskněte klávesu, abyste ukončili úsečku
Press-imp-2pl key would-2pl end-pastparticiple line-gen
Press a key in order to end the line.

Bg: *Натиснете клавиш, за да завършите линията.*
Press-2pl key in-order-to end-2pl line-det.

Additional statement at the top level of the SPL:

```
(R / RST-PURPOSE
   :dependent-type-q subordination
   :DOMAIN …
   :RANGE …  )
```

Bg: (dependent: finite)
Натиснете клавиш, за да завършите линията.
Press-2pl key in-order-to end-2pl line-det.

SPL for Bulgarian must contain aspect-specifications:

```
(R / RST-PURPOSE
 :DOMAIN
   (P / DIRECTED-ACTION :LEX NATISNA
   :SPEECHACT IMPERATIVE
   :ASPECT-Q PERFECTIVE-ASPECT
  :ACTOR (hearer / DM::USER)
```

```
      :ACTEE (D / DM::KEY))
      :RANGE
      (E / DIRECTED-ACTION :LEX ZAVURSHA
     :ASPECT-Q PERFECTIVE-ASPECT
    :ACTEE
         (P2 / OBJECT :LEX POLILINIA :IDENTIFIABILITY-Q IDENTIFIABLE)))
```

(39) **hypotactic enhancement: manner (means) circumstantial**

En: *End the line by pressing a key.*

Bg: (dependent: finite)
Завършете линията, като натиснете клавиш.
end-2pl.imp. line-det. by press-2pl.ind. key

Cz: (dependent: nominalization)
Úsečku ukončete stisknutím klávesy.
Line-acc end pressing-ins key-gen.

Ru: *Завершите линию, нажав клавишу*

```
(c / (RST-Means Manner)
    :Domain
    (S1 / DM::END-LINE
        :SPEECHACT IMPERATIVE
        :ACTOR (hearer / DM::USER)
        :ACTEE (P1 / DM::LINE :IDENTIFIABILITY-Q IDENTIFIABLE))
    :Range
    (S2 / DM::PRESS
        :ACTOR (hearer / DM::USER)
        :ACTEE (A / DM::key
                 :IDENTIFIABILITY-Q IDENTIFIABLE))))
```

(40) **hypotactic enhancement: conditional circumstantial** (not implemented in Bulgarian)

En: *If you use Windows, press a key.*

Cz: *Pokud používáte Windows, stiskněte klávesu.*
If use-2pl Windows press-2pl key

Ru: *Если вы используете Windows, нажмите клавишу.*
If you use Windows press-2pl key-acc

```
(G15490 /
        (RST-Logical-Condition)
        :RANGE
        (|a19| / dispositive-material-action :lex pouz3i2vat
             :ACTEE (|a21| / DM::gui-windows
                           :IDENTIFIABILITY-Q DM::IDENTIFIABLE)
             :ACTOR
                (HEARER / DM::USER :IDENTIFIABILITY-Q DM::IDENTIFIABLE))
       :theme |a19|
        :DOMAIN
        (|a11| / DM::PRESS
        :speechact imperative
        :ACTEE (|a13| / DM::key)
        :ACTOR
               (HEARER / DM::USER :IDENTIFIABILITY-Q DM::IDENTIFIABLE))
)
```

Combinations of various paratactic and/or hypotactic relations can also be handled.

5.1.9 Determination

Czech and Russian do not have the definite or indefinite article, Bulgarian has no indefinite article, but has a definite article, which appears as a suffix. In all three languages, we thus have nominal groups without any explicit Deictic element. For Czech, the absence of a Deictic element has been implemented as the default choice. When an explicit Deictic is required, this is reflected by the SPL statement

```
:explicit-deictic-q explicit
```

(41) **Implicit Deictic**

En: *You add the point.*

Cz. *Bod přidáte.*
Point-acc add-2pl.

Ru: *Вы добавляете точку.*
You add-2pl point-acc.

```
(|a121| / DM::ADD
 :ACTEE (|a105| / DM::POINT
                  :IDENTIFIABILITY-Q IDENTIFIABLE)
 :ACTOR (HEARER / DM::USER
                  :IDENTIFIABILITY-Q DM::IDENTIFIABLE ) )
```

(42) **Explicit Deictic**

En: *You add the point.*

Cz. *Ten bod přidáte.*
That point-acc add-2pl.

Ru: *Вы добавляете эту точку.*
You add-2pl this point-acc.

```
(|a121| / DM::ADD
 :ACTEE (|a105| / DM::POINT
                  :IDENTIFIABILITY-Q IDENTIFIABLE
                  :explicit-deictic-q explicit )
 :ACTOR (HEARER / DM::USER
                  :IDENTIFIABILITY-Q DM::IDENTIFIABLE ) )
```

The realization of the Deictic element can be further specified using the determiner macro

```
:determiner X
```

where X stands for the chosen determiner. For example, the macro for the counterpart of "this" is defined as follows:

```
(this
   :identifiability-q  identifiable
   :proximity-modification-q proximity
   :proximity-mod-id (speaker / person
                      :distance-q nondistant))
```

5.1.10 Subject Dropping

Czech and Bulgarian are characterized as so-called *pro-drop* languages. This characteristic reflects the fact that the Subject in a declarative clause (indicative or interrogative mood) can be unexpressed/unrealized (on the surface). It might also be said the Subject is left implicit, perhaps in a similar way as a Subject is left implicit in a sentence in imperative mood (also in English).

(43) **Dropped interactant subject in indicative declarative mood**

En: *You create a line.*

Cz. *Vytvoříte úsečku.*
Create-2pl line-acc.

```
(|a121| / DM::CREATE
 :ACTEE (|a105| / DM::LINE )
 :ACTOR (HEARER / DM::USER
                  :IDENTIFIABILITY-Q DM::IDENTIFIABLE ) )
```

Bu. *Създавате линия.*
create-2pl.ind. line.

```
(|a121| / DM::CREATE
 :ACTEE (|a105| / DM::LINE )
 :ACTOR (HEARER / DM::USER
                  :IDENTIFIABILITY-Q DM::IDENTIFIABLE )
:SUBJECT-MINIMAL-ATTENTION-Q MINIMAL )
```

(44) **Dropped non-interactant subject in indicative declarative mood**

En. *It appears on the screen.*

Cz. *Objeví se na obrazovce.*
Appears refl on screen-loc.

```
(|a121| / DM::APPEAR
 :ACTOR (L / DM::LINE
           :RECOVERABILITY-Q DM::RECOVERABLE
           :CONTEXTUAL-BOUNDNESS yes)
 :SPATIAL-LOCATING (D / (ONE-OR-TWO-D-LOCATION DM::SCREEN))
```

Bg: *Появява се на екрана.*
Appears refl on screen.

```
(|a121| / DM::APPEAR
:ACTOR (L / DM::LINE
          :IDENTIFIABILITY-Q DM::IDENTIFIABLE)
:SUBJECT-MINIMAL-ATTENTION-Q MINIMAL
:SPATIAL-LOCATING (D / (ONE-OR-TWO-D-LOCATION
                        DM::SCREEN
                        :IDENTIFIABILITY-Q DM::IDENTIFIABLE))
```

(45) **Not-dropped interactant subject in indicative declarative mood**

En: *You create a line.*

Cz. *Vytvoříte úsečku.*
Create-2pl line-acc.

```
(|a121| / DM::CREATE
 :ACTEE (|a105| / DM::LINE )
 :ACTOR (HEARER / DM::USER
           :SUBJ-MINIMAL-ATTENTION-Q NONMINIMAL
           :IDENTIFIABILITY-Q DM::IDENTIFIABLE
           :CONTEXTUAL-BOUNDNESS yes ) )
```

Bg. *Вие създавате линия.*
you-2pl create-2pl.ind. line

```
(|a121| / DM::CREATE
 :ACTEE (|a105| / DM::LINE )
 :ACTOR (HEARER / DM::USER
          :IDENTIFIABILITY-Q DM::IDENTIFIABLE ) )
```

(46) **Not-Dropped interactant subject in indicative declarative mood**

En: *It appears on the screen.*

Cz. *Čára se objeví na obrazovce.*
Line refl appears on screen-loc.

```
(|a121| / DM::APPEAR
 :ACTOR (L / DM::LINE
          :RECOVERABILITY-Q DM::RECOVERABLE
          :SUBJ-MINIMAL-ATTENTION-Q NONMINIMAL
          :CONTEXTUAL-BOUNDNESS yes)
 :SPATIAL-LOCATING (D / (ONE-OR-TWO-D-LOCATION DM::SCREEN))
```

Bg: *Линията се появява на екрана.*
Line-det. refl appears on screen-det.

```
(|a121| / DM::APPEAR
 :ACTOR (L / DM::LINE
          :IDENTIFIABILITY-Q DM::IDENTIFIABLE)
 :SPATIAL-LOCATING (D / (ONE-OR-TWO-D-LOCATION DM::SCREEN
                         :IDENTIFIABILITY-Q DM::IDENTIFIABLE))
```

5.1.11 Numerals in nominal group

5.1.11.1 Cardinal numbers

Expressions with cardinal numbers are used to specify the number of objects that operate as a participant in a construction.

(47) Cardinal

En: *Specify seven points.*

Bg: *Задайте седем точки.*

Cz: *Určete sedm bodů.*

Ru: *Задайте семь точек.*

The cardinal number in SPL expressions is specified using the following information:
`:QUANTITY X`

where X is a number, which can be within range of 1 to 99 for Bulgarian and Russian and 1 to 9 for Czech.

5.1.11.2 Ordinal numbers

Expressions with ordinal numbers are used to specify the ordinal number of an object:

(48) Cardinal

En: *Specify the seventh point.*

Bg: *Задайте седмата точка.*

Cz: *Zadejte sedmý bod.*

Ru: *Задайте седьмую точку.*

The cardinal number in SPL expressions is specified using the following information:

```
:ORDINAL X
```

where X is the English ordinal corresponding to the number and is within range of *first* to *ninth* for Russian, Czech and Bulgarian.

5.1.11.3 Quantification

We model a subset of quantification expressions that are aimed at specification of number of objects to be selected in a larger set.

(49) Quantity selection

En: *Open the dialog box using one of the following methods:*

Bg: *Отворете диалоговия прозорец с един от следните методи:*

Cz: *Otevřete dialogové okno jedním z následujících způsobů.*

Ru: *Откройте диалоговое окно одним из следующих способов*:

The cardinal number in SPL expressions is specified using the following information:

```
:QUANTITY-SELECTION-Q QUANTITY
:QUANTITY-SELECTION-id X
```

where X is a number, which can be within range of 1 to 99 for Bulgarian and Russian and 1 to 9 for Czech.

5.1.12 Subject-Verb agreement

In Czech, Bulgarian and Russian, a predicate (that means all parts of a compound verb) agrees with its nominative subject in person, number and gender (if applicable). This holds even if the subject is realized by a zero pronoun (so called pro-drop). We do not model agreement with a coordinated subject, in such a case, the most probable morphological categories are assigned to predicate: feminine gender, plural number and the third person.

(50) Subject – simple predicate agreement

En: *The dialog box disappears.*

Ru: *Диалоговое окно исчезнет с экрана.*

Bg: *Диалоговият прозорец изчезва.*

Cz: *Dialogový panel zmizí.*

(51) Subject – compound predicate

En: *When PLINE has been started, …*

Ru: *Когда была запущена команда PLINE. …*

Bg: *Когато е стартирана командата PLINE*

Cz: *Když byl spuštěn příkaz PLINE, …*

5.1.13 Nominal group agreement

Within the nominal group, there is agreement between the head (pro)noun (Thing) and its modifiers, i.e., deictics, qualities and ordinal numerals.

(52) Nominal group agreement (adjective)

En: *in the Multiline Style dialog box.*

Bg: *в диалоговия прозорец Multiline Style*

Cz: *v dialogovém okně Styl multičáry.*

Ru: *в диалоговом окне Multiline Style.*

```
:SPATIAL-LOCATING
       (|a15| / THREE-D-LOCATION :LEX PANEL :PROPERTY-ASCRIPTION
              (G5639 / QUALITY :LEX DIALOGOVY2))
```

(53) Nominal group agreement (ordinal numeral)

En: *Specify the seventh point.*

Bg: *Задайте седмата точка.*

Cz: *Zadejte sedmý bod.*

Ru: *Задайте седьмую точку.*

5.2 Coverage of target texts

Within the CORP deliverable, we introduced a set of lexicogrammatical features and coded texts from the AutoCAD manual, Chapter 2 "Drawing objects" in terms of this set. At the end of the project we tested the coverage of the target texts as well as of software manuals and instructional texts in general by the grammar regions identified above.

The following list of attributes and their possible values were used for corpora coding:

- **Lexico-Grammatical Features**
  - **LGF 1: Rank**

    {CL - Clause, **NGR** - Nominal Group, **PGR** - Prepositional group}
  - **LGF 2: Process-type**

    {RP - relational**, MNP -** mental**, VEP -** verbal,

    DMP - directed-material**, NMP -** nondirected-material**}**
  - **LGF 3: Finiteness**

    {FIN - finite**, NFIN -** nonfinite}
  - **LGF 4: Polarity**

    {POS - positive**, NEG -** negative}
  - **LGF 5: Modality**

    {NMD - notmodal**, MDZP -** probability**, MDZU -** usuality**,**

    MDLI - inclination**, MDLA -** ability**, MDLO -** obligation}
  - **LGF 6: Mood**

    {IMP - imperative**, INT -** interrogative**, IND -** indicative}
  - **LGF 7: Voice**

{VMID - middle**, VEFA -** active**, VEFP -** passive}

- **LGF 8: Agency-specified**

{AGSU - agency-specified-user**, AGSP -** agency-specified-program**,**
AGSA - agency-specified-act**, AGN -** agency-notspecified**, AGG -**
agency- specified-generic}

- **LGF 9: Member-of-clause-complexity**

{CCS - clause-complexity-simplex**, CCC -** clause-complexity-complex}

- **LGF 10: Clause-dependency**

{NDC - notdependent-clause**, DCT -** thematized-dependent-clause**,**
NDCT - notthematized-dependent-clause}

- **LGF 11: Clause-taxis**

{PRC - paratactic-relation clause**, HRC -** hypotactic-relation clause}

- **LGF 12: Conjunction-type**

{CTAD - conjunction-type-additive**, CTAL -** conjunction-type-alternative**,**
CTT - conjunction-type-temporal**, CTS -** conjunction-type-spatial,
CTM - conjunction-type-manner, **CTCR** - conjunction-type-causal-
Reason, **CTCP** - conjunction-type-causal-purpose,
CTCC - conjunction-type-causal-condition}

The coded corpus contained step-by-step procedures as well as other text types, i.e. non-procedural text bits. In the CORP deli we presented an analysis of the correlation of features with text types. The selection of features for coding was informed by earlier analyses of software-instructional texts. Thus, the coding set does not include all NIGEL features and the corpus does not include some features relevant for instructional texts.

In Table 6 below we present the coverage of the corpus by our grammars in terms of these features. We also added to this list phenomena that concern word order and circumstantial transitivity. For each language and each feature, we provide information on whether the feature exists in the corpus for the language and whether it is implemented. In addition to the coded corpus, we consider importance of phenomena for software manuals or instructional texts in general. There are also some differences between coverage of linguistic resources and coverage of the complete integrated system caused by limitations in the knowledge representation and the text structuring components. In the case of knowledge representation, some semantic specification types lack in the Domain Model or some relations cannot be expressed in the knowledge editor. In the case of the text structuring component, some types of SPL input cannot be not provided by the sentence planner. On the other hand, the coded manual is expressed in a particular style, e.g. using polite imperatives for instructions, and therefore, is limited with respect to features relevant for other styles (cf. the discussion of these features in deliverables TEXS2, TEXS3).

The marks in Table 6 have the following meaning:

- E a feature exists in the coded corpus and is important for instructional texts in general;

- R a feature exists in the coded corpus, but is rare (less than 5%);
- S a feature does not exist in the coded corpus, but is significant for software manuals or instructional texts in general (some features that might be rare in the coded corpus are abundant in other corpora; this mark is compatible with R);
- M a feature is of minor importance for instructional texts (some features rarely occur in instructional texts, for example, VEP or MDLI in Russian; this mark is compatible with R);

There are also three marks with respect to implementation of features in the resources:

- I a feature is implemented in the resource set (it covers all the cases relevant for software instructions, for example, this is true for IMP in Russian);
- P a feature is partly implemented (i.e. it covers some possibilities, in particular, found in the reference texts, but we know important implementation gaps for instructional texts);
- N a feature is not implemented in the resource set.

Mood				
IMP	**E/I**	**E/I**	**E/I**	
INT	**M/N**	**S/N**	**M/N**	
IND	**E/I**	**E/I**	**E/I**	
Voice				
VMID	**S/I**	**R/I**	**S/I**	
VEFA	**E/I**	**E/I**	**E/I**	
VEFP	**R/N**	**R/N**	**S/I**	
Agency				
AGSU	**E/I**	**E/I**	**E/I**	
AGSP	**R/I**	**R/I**	**E/I**	
AGSA	**R/I**	**R/I**	**S/I**	
AGN	**E/I**	**E/I**	**S/I**	
AGG	**M/N**	**S/N**	**M/N**	
Clause compl.				
CCS	**E/I**	**E/I**	**E/I**	
CCC	**E/I**	**E/I**	**E/I**	
Clause depend.				
NDC	**E/I**	**E/I**	**E/I**	
DCT	**E/I**	**E/I**	**E/I**	
NDCT	**E/I**	**E/I**	**E/I**	
Clause taxis				
PRC	**E/I**	**E/I**	**E/I**	

HRC	**E/I**	**E/I**	**E/I**	
Conjunction				
CTAD	**E/I**	**E/P**	**E/I**	
CTAL	**E/I**	**E/I**	**E/I**	
CTT	**E/I**	**E/I**	**E/I**	
CTS	**S/N**	**S/N**	**S/N**	
CTM	**S/I**	**S/I**	**E/I**	
CTCR	**S/N**	**S/N**	**M/N**	
CTCP	**E/I**	**E/I**	**E/I**	
CTCC	**S/I**	**S/I**	**S/I**	
Word Order	**E/P**	**E/I**	**E/P**	Information structure (**Error! Reference source not found.**) can alter default order

Table 6 Coverage of phenomena in the coded corpora

6. Evaluation of exploitation potential

The Bulgarian and Russian sites gathered informal opinions on AGILE's exploitation potential through contacts in academia and industry.

The general opinion is that multilingual linguistic resources that are suited for generation can be certainly used in the wide variety of applications, in particular those related to Internet. As for the integrated system, the opinion on its exploitation potential is that the extensibility of the Domain Model and development of large scale task models is questionable using the existing interface. Negative comments also concerned the slowness of the system, its lack of robustness, and the cumbersome selection of concepts from lists. As a research product, such a system was judged highly useful; however, its commercial application requires its redesign taking into account modern practices in software engineering and user interface development.

7. 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 that 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.

Evaluators rated the abilities of the Final Prototype—on the scale of Unsatisfactory, Poor, Good and Excellent—as Good, with the exception of the knowledge editing abilities and general robustness of the interface, which were rated slightly below Good (on average). Detailed analysis of task models produced by evaluators shows that the most of them have the correct structure, but some of them lack proper co-indexation of concept instances and do not distinguish between preconditions and first steps of methods. We also employed a novel methodology for the analysis of the accuracy of the task models developed, in both analysing their formal content and also using the Generation Accuracy metric, which is based on the generated texts. The quality of the metric correlates well with the results of the formal content analysis, while the metric calculation is much less labour-intensive.

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 all three languages. The quality of the generated texts in different types (Overviews, Functional Descriptions, Full Procedures, Quick References) was judged between Good and Excellent, with the exception of Overviews, which were rated between Poor and Good.

Almost no grammatical errors were identified by the judges: none at all for Bulgarian, only one for Czech, and four for Russian. All of the identified grammatical errors were classified as ones of word order (WO), a well-known problem for Eastern European languages. While some of identified problems of word order in the generated texts were clearly grammatical, others were less clearly so; in at least two cases, judges thought they might be stylistic. We take this result as strong validation of the quality of the grammars we have developed for these three languages. To our knowledge, the AGILE project is the first successful attempt at developing computational grammars for the generation of Bulgarian, Czech or Russian.

The final stage of evaluation concerned the extent to which the grammatical constructions found in the target texts, CAD/CAM manuals, software manuals in general and other types of instructions are covered by the multilingual resources developed within the project. These resources cover a significant proportion of the grammar (for example, imperative instructions for the user, and clause complexity), but there are some gaps in coverage (for example, concerning negation and relational processes, though development of such constructions was hindered by the inability to represent their semantics in the Domain Model).

References

Agile Project (1997) Automatic generation of instructions in languages of Eastern Europe. Technical Annex, Commission of the European Community, Brussels, DG XIII, 1997.

Alshawi, H., Bangalore, S. and Douglas, S. (1998). Automatic acquisition of hierarchical transduction models for machine translation. *Procs. ACL/COLING'98*, Montreal, Canada.

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

Bateman J. A. (1997) Enabling technology for multilingual natural language generation: the KPML environment. *Natural Language Engineering* 1(1):1-42, 1997.

Bangalore, S. and Rambow, O. (2000). Exploiting a probabilistic hierarchical model for generation. *Procs. COLING 2000*, Saarbruecken, Germany. pp. 42—48.

Bangalore, S., Rambow, O., and Whittaker, S. (2000) Evaluation Metrics for Generation. In: Proceedings of INLG'2000, pp. 1-8. Also available from:

http://www.cs.bgu.ac.il/~nlg2000/archive/rambow/bangalore-et-al.ps.gz

Kasper R. T. (1989) A flexible interface for linking applications to PENMAN's sentence generator. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, 1989. Available from USC/Information Sciences Institute, Marina del Rey, CA.

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

Teich E. (1999) *Systemic Functional Grammar in Natural Language Generation: Linguistic Description and Computational Representation.* Cassell Academic, London, 1999.

Yang G., K. F. McCoy, and K. Vajay-Shanker (1991) From functional specification to syntactic structures: Systemic Grammar and Tree Adjoining Grammar. *Computational Intelligence*, 7(4):207 --219, 1991.

Шведова Н.Ю.(Ed). (1970) *Грамматика современного русского литературного языка.* Наука, Москва.

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. Deliverable for WP3.

[EVAL1] Serge Sharoff, Donia Scott, Anthony Hartley, Danail Dochev, Jiri Hana, Martin Cmejrek, Ivana Kruijff-Korbayova, Geert-Jan Kruijff, *Evaluation of the intermediate prototype.* April, 2000. Deliverable for WP9.1.

[SPEC3-IMLP3] Serge Sharoff, Lena Sokolova, Danail Dochev, Ivana Kruijff-Korbayová, Geert-Jan Kruijff, Jiří Hana, Kamenka Staykova, Elke Teich, John Bateman, *Formal specification of full grammar models. Implementation of tactical generation*

resources for all three languages in a Final Prototype, July, 2000. Joint deliverable for WP6.3 and WP7.3.

[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*. Deliverable for WP5.2, April 1999.

[TEXS3] Tony Hartley, Ivana Kruijff-Korbayová, Geert-Jan Kruijff, Danail Dochev, Ivan Hadjiiliev, Lena Sokolova *Text Structuring Specification for the Final Prototype.* Deliverable for WP5.3, December, 1999.

Appendices

Appendix 1: The usability evaluation scenario

Usability evaluation of AGILE

Preparation

Lispworks with the AGILE interface should be started prior to conducting the evaluation.

Duration of tasks is announced to the subjects. You should not help them, while they accomplish their tasks, but you may record some of their reactions to help in filling the questionnaire.

Verbal task model specifications are shown here framed in a box, whose first line expresses the task of a procedure. Steps are expressed by imperative clauses, their side-effects by indicative clauses. A sequence of substeps which constitute a method for achieving a complex instruction step is indented. Alternative methods are marked by M followed by the method number. 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 bold face. Labels of entities are marked by the italic script. For non-bold non-labelled entities, the reader should check for their previous occurrence in the specification.

Task 1: Edit a simple task model

The directory \Agile\A-boxes\Task1\ contains a task model that has been developed by another member of your team. You have 10 minutes for completion of this task.

The task is to produce the MiniModel-1 which matches Minispec-1.

1. Load the file “MiniSpec1-a.abx”.
2. Understand the content specified by the model.
3. Specify a precondition: “start-command the *Pline* command.”
4. Change the last step to mean: “Specify the endpoint of **a segment** of the polyline.”
5. Generate the output into your own native language as “Long Instructions” text type.
6. Save the model as “MiniSpec-1.abx”.

Task 2: Create a simple task model

You have 20 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 and hide complex steps.

2. Generate the text and check how it matches your intentions. If there are discrepancies between your intensions and the model content, edit the model and generate the text again to make sure that the text conveys your intended meanings.
3. Save the model as “MiniSpec-X.abx” in the directory \Agile\A-boxes\Task2\.

Task 3: Compose a task model set

You have 15 minutes for completion of this task.

1. Compose a task model set from four simple task models in the directory \Agile\A-boxes\Task3\.
2. Generate from it output in all text types and in all 3 languages.
3. Save the task model set as “MiniSet.abx”

Task 4: Create two complex task models

You have 2 hours for completion of this task.

1. Create a new model according to the specification below. The model in \Agile\A-boxes\Task3\MiniSpec1.abx may be used for this task.

MaxiSpec-1: common to all sites.

2. Generate the text and check how it matches your intentions.
3. Save the model as “MaxiSpec-1.abx” in the directory \Agile\A-boxes\Task4\.
4. Create a new model according to the specification below. The model in \Agile\A-boxes\Task3\MiniSpec1.abx may be used for this task.
5. Generate the text and check how it matches your intentions.
6. Save the model as “MaxiSpec-X.abx” in the directory \Agile\A-boxes\Task4\.

Task 5: Compose a complex task model set

You have 30 minutes for completion of this task.

1. Compose a task model set from four task models in the directory \Agile\A-boxes\Task5\.
2. Generate from it output in all text types and in all 3 languages.
3. Save the task model set as “MaxiSet.abx”

Appendix 2: The design of usability evaluation 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

Task5

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

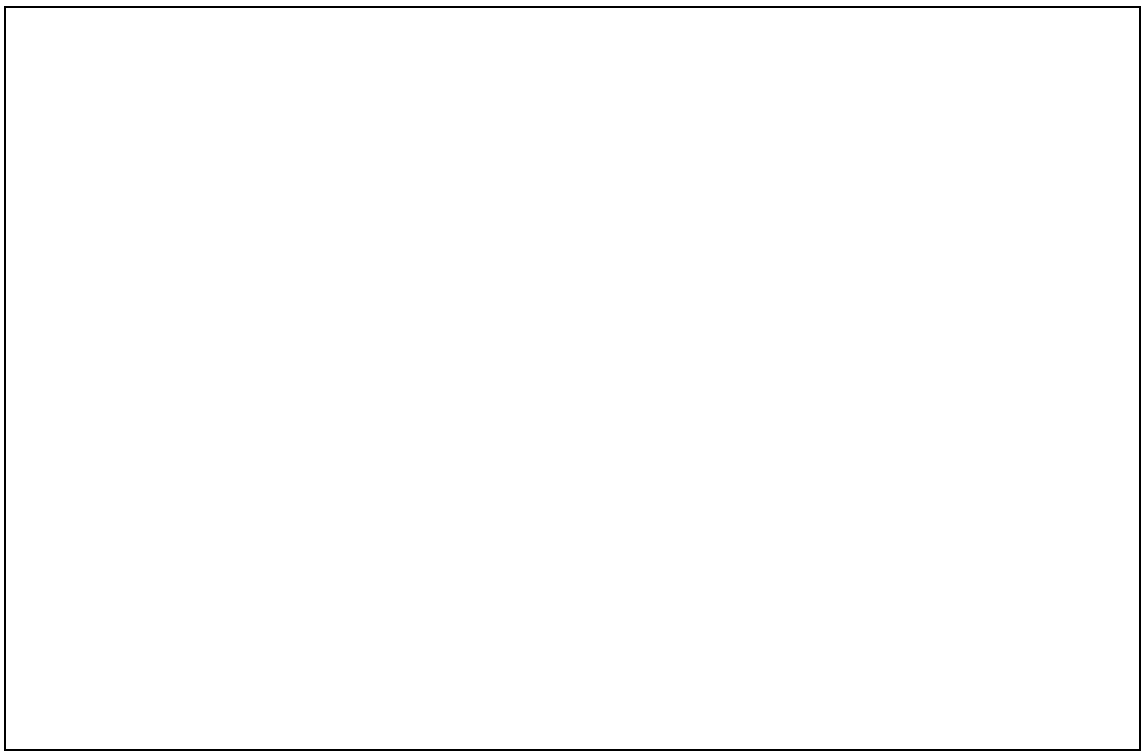
Use the following scale of scores for your judgements:

Score	Explanation
4	Excellent
3	Good
2	Poor
1	Useless

Score the AGILE interface capabilities by writing your scores in the table below.

Ref	Question	Score
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 expresses the content you specified?	
8	Adequacy of the user documentation	

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



Appendix 3: Task model specifications for usability evaluation

MaxiSpec-1: common to all sites.

To draw **a polyline**
First, start-tool the *Pline* command.
M1. Using the *Windows* operating system: choose the *Polyline* option from the *Polyline* flyout on the *Draw* toolbar.
M2. Using the *DOS* or *UNIX* operating system: choose the *Polyline* option from the *Draw* menu.
Specify **the first segment** of the polyline.
Specify **the first point** of the first segment.
Specify **the endpoint** of the first segment.
Specify **the second segment** of the polyline.
Repeat **these steps**.
End the polyline.
Press the *Return* key.

MaxiSpec-2 (BAS specific)

Draw **a line-arc-polyline**
First, start-tool the *Pline* command.
M1. Using the *Windows* operating system: choose the *Polyline* option from the *Polyline* flyout on the *Draw* toolbar.
M2. 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**.
First switch to the *Arc* mode.
Enter the *a* command line. 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.
Specify **the second point** of the arc.
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.
Click 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.

MaxiSpec-3 (CU specific)

Draw **an arc**
First, start-tool the *ARC* command.
M1. Using the *Windows* operating system: choose the *3 Points* option from the *Arc* flyout on the *Draw* toolbar.
M2. Using the *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.
Specify **a point**. The arc snaps to the point.
Specify **the endpoint** of the arc.
End-line 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.
Enter **a description** in the *Description* field.
Close the *Save* dialogue box.
Click the *Save* button. The *Save* dialogue box disappears.

MaxiSpec-4 (RRIAI specific)

Define **a style** of **a multiline** in **a drawing**
First, open-screen-object the *Multiline Style* dialog box. The *Multiline Style* dialog box appears.
M1. Using the *Windows* operating system: choose the *Multiline Style* option from the *Multiline Style* flyout on the
Data toolbar.
M2. Using the *DOS* or *UNIX* operating system: choose the *Multiline Style* option from the *Data* menu.
Specify **the properties** of the multiline.
Click the *Multiline Properties* button. The *Multiline Properties* dialog box appears.
Specify **the offset** of **the elements** of the multiline.
Display **vertices** of the multiline.
Choose the *Display Joints* option.
In the *Caps* heading, specify the properties of **the startpoint** of the multiline.
In the *Caps* heading, specify the properties of **the endpoint** of the multiline.
Display **background colour** of the drawing.
In the *FIll* heading, choose the On option.
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** in the *Description* field.
Add the style to the drawing.
Click the *Add* button.
Click the *Save* button.
Click the *OK* button in the *Multiline Style* dialog box. The *Multiline Style* dialog box disappears.

Appendix 4: The design of text quality (acceptability) questionnaire sheets

Acceptability evaluation of texts produced AGILE

Data on the evaluator of AGILE texts.

0. **Name and affiliation:**

1. **Expertise with technical writing and/or translation of software manuals:**

Task 1: Assessing the quality of Overviews

2 texts: one by-object and one by-process grouping

Text N

Box containing text N

On the scale **4 = Excellent | 3 = Good | 2 = Poor | 1 = Terrible**, to what extent is this text …	**Score**	
… well organised?		N.1
… well written?		N.2
… useful as a draft for a high-quality manual?		N.3

On the scale **2 = Satisfactory | 1 = Unsatisfactory**, to what extent does this text …	**Score**	
… contain the right level of detail?		N.4
If **Unsatisfactory** …		
… is the detail **2 = Too much | 1 = Too little**?		N.5

Task 2: Assessing the quality of Procedures as generated relative to their wordings in the reference texts

In the following table examples of discrepancies are presented. Each example is presented as a pair of variant wordings R and G in a local, invariant context, for example:

To create a line:

(A) Specify the start point.
Specify the end point.
(B) Specify the start point, then specify the end point.

Save the line.

The order in which reference and generated examples are presented is random.

Ref	Text Fragment

On the scale **3 = A is better | 2 = B is better | 1 = A and B are equal**,
score each of the pairs of variants presented on the accompanying sheets.

Pair	Score
1	
2	
3	
N	

Task 3: Assessing the intrinsic quality of Procedures

4 texts

Text N

Box containing text N

On the scale **4 = Excellent | 3 = Good | 2 = Poor | 1 = Terrible**, to what extent is this text …	**Score**	
… well organised?		N.1
… well written?		N.2
… useful as a draft for a high-quality manual?		N.3

Task 4: Assessing the quality of Quick reference texts

4 texts

Text N

Box containing text N

On the scale **4 = Excellent | 3 = Good | 2 = Poor | 1 = Terrible**, to what extent is this text …	**Score**	
… well organised?		N.1
… well written?		N.2

On the scale **2 = Satisfactory | 1 = Unsatisfactory**, to what extent does this text …	**Score**	
… contain the right level of detail?		N.4
If **Unsatisfactory** …		
… is the detail **2 = Too much | 1 = Too little**?		N.5

Appendix 5: Target texts for text quality (grammaticality) evaluators

Bulgarian: personal style

Кратък преглед

Програмата позволява да добавяте стил на мултилиния към чертеж. Освен това програмата позволява да запазвате стила на мултилинията във файл.

Пълни инструкции

Дефиниране на стил на мултилиния в чертеж

- *Изберете опцията Multiline Styles от менюто Data.*
    - ***Windows*** *Изберете опцията Multiline Styles от плаващото меню Multiline Styles във функционалния ред Data.*
    - ***DOS / UNIX*** *Изберете опцията Multiline Styles от менюто Data.*

Диалоговият прозорец Multiline Styles се появява.

- Отначало задайте свойствата на мултилинията.
    - 1. Щракнете бутона Multiline Properties.
      Диалоговият прозорец Multiline Properties се появява.
    2. Задайте отместването на елементите на мултилинията.
    3. Изберете опцията Display Joints, за да покажете краищата на мултилинията.
    4. Задайте свойствата на началната точка на мултилинията в заглавното поле Caps.
    5. Задайте свойствата на крайната точка на мултилинията в заглавното поле Caps.
    6. Изберете опцията On от заглавното поле Fill, за да покажете фоновия цвят на чертежа.
    7. Щракнете бутона OK в диалоговия прозорец Multiline Properties.

    Диалоговият прозорец Multiline Properties изчезва.
- След това запазете стила на мултилинията във файл.
    - Отначало въведете името на стила на мултилинията в полето Name. След това въведете описание в полето Description. Щракнете бутона Add, за да добавите стила на мултилинията към чертежа. Накрая щракнете бутона Save.

Накрая щракнете бутона OK в диалоговия прозорец Multiline Styles.
Диалоговият прозорец Multiline Styles изчезва.

Кратки инструкции

Дефиниране на стил на мултилиния в чертеж

Windows

Изберете опцията Multiline Styles от плаващото меню Multiline Styles във функционалния ред Data.

DOS / UNIX

Изберете опцията Multiline Styles от менюто Data.

Щракнете бутона Multiline Properties.

Задайте отместването на елементите на мултилинията.

Изберете опцията Display Joints.

Задайте свойствата на началната точка на мултилинията в заглавното поле Caps.

Задайте свойствата на крайната точка на мултилинията в заглавното поле Caps.

Изберете опцията On от заглавното поле Fill.

Щракнете бутона OK в диалоговия прозорец Multiline Properties.

Въведете името на стила на мултилинията в полето Name.

Въведете описание в полето Description.

Щракнете бутона Add.

Щракнете бутона Save.

Щракнете бутона OK в диалоговия прозорец Multiline Styles.

Функционално описание

Бутонът Add добавя стил на мултилиния към чертеж.

Бутонът Multiline Properties задава свойствата на мултилинията.

Опцията Display Joints показва краищата на мултилинията.

Опцията On показва фоновия цвят на чертежа.

Bulgarian: impersonal style

Пълни инструкции

Дефиниране на стил на мултилиния в чертеж

- *Избира се опцията Multiline Styles от менюто Data.*
    - ***Windows*** *Избира се опцията Multiline Styles от плаващото меню Multiline Styles във функционалния ред Data.*
    - ***DOS / UNIX*** *Избира се опцията Multiline Styles от менюто Data.*

Диалоговият прозорец Multiline Styles се появява.

- Отначало се задават свойствата на мултилинията.
    - 1. Щраква се бутонът Multiline Properties.
      Диалоговият прозорец Multiline Properties се появява.
    2. Задава се отместването на елементите на мултилинията.
    3. Избира се опцията Display Joints, за да се покажат краищата на мултилинията.
    4. Задават се свойствата на началната точка на мултилинията в заглавното поле Caps.
    5. Задават се свойствата на крайната точка на мултилинията в заглавното поле Caps.
    6. Избира се опцията On от заглавното поле Fill, за да се покаже фоновият цвят на чертежа.
    7. Щраква се бутонът OK в диалоговия прозорец Multiline Properties.

  Диалоговият прозорец Multiline Properties изчезва.
- След това се запазва стилът на мултилинията във файл.
    - Отначало се въвежда името на стила на мултилинията в полето Name. След това се въвежда описание в полето Description. Щраква се бутонът Add, за да се добави стилът на мултилинията към чертежа. Накрая се щраква бутонът Save.

  Накрая се щраква бутонът OK в диалоговия прозорец Multiline Styles.
  Диалоговият прозорец Multiline Styles изчезва.

Кратки инструкции

Дефиниране на стил на мултилиния в чертеж

Windows

Избира се опцията Multiline Styles от плаващото меню Multiline Styles във функционалния ред Data.

DOS / UNIX

Избира се опцията Multiline Styles от менюто Data.

Щраква се бутонът Multiline Properties.

Задава се отместването на елементите на мултилинията.

Избира се опцията Display Joints.

Задават се свойствата на началната точка на мултилинията в заглавното поле Caps.

Задават се свойствата на крайната точка на мултилинията в заглавното поле Caps.

Избира се опцията On от заглавното поле Fill.

Щраква се бутонът OK в диалоговия прозорец Multiline Properties.

Въвежда се името на стила на мултилинията в полето Name.

Въвежда се описание в полето Description.

Щраква се бутонът Add.

Щраква се бутонът Save.

Щраква се бутонът OK в диалоговия прозорец Multiline Styles.

Czech: personal style (explicit)

Shrnutí

Program umožňuje přidávat styl multičáry do výkresu. Dále styl multičáry program umožňuje ukládat do souboru.

Dlouhé instrukce

Definování stylu multičáry ve výkrese

-
  - *Vyberte volbu Styly multičár z nabídky Data.*
    - ***Windows*** *Vyberte volbu Styly multičár z plovoucí nabídky Styly multičár na nástrojovém panelu Data.*
    - ***DOS a Unix*** *Vyberte volbu Styly multičár z nabídky Data.*

  Objeví se dialogový panel Styly multičár.

  - Nejprve určete vlastnosti multičáry.
    -
      1. Stiskněte tlačítko Vlastnosti multičár.
         Objeví se dialogový panel Vlastnosti multičár.
      2. Určete vzdálenost elementů multičáry.
      3. Pro zobrazení vrcholů multičáry vyberte volbu Zobraz klouby.
      4. Určete vlastnosti počátečního bodu multičáry v nadpise Zakončení.
      5. Určete vlastnosti koncového bodu multičáry v nadpise Zakončení.
      6. Pro zobrazení barvy pozadí vyberte volbu Ano z nadpisu Vyplnění.
      7. Stiskněte tlačítko OK v dialogovém panelu Vlastnosti multičár.

    Zmizí dialogový panel Vlastnosti multičár.
  - Potom styl multičáry uložte do souboru.
    -

- Nejprve zadejte název stylu multičáry v poli Jméno. Potom zadejte popis v poli Popis. Pro přidání stylu multičáry do výkresu stiskněte tlačítko Přidat. Nakonec stiskněte tlačítko Uložit.

Nakonec stiskněte tlačítko OK v dialogovém panelu Styly multičár.
Zmizí dialogový panel Styly multičár.

Krátké instrukce

Definování stylu multičáry ve výkrese

2. :

a. **Windows** :

1. Vyberte volbu Styly multičár z plovoucí nabídky Styly multičár na nástrojovém panelu Data.

b. **DOS a Unix** :

0. Vyberte volbu Styly multičár z nabídky Data.

3. Stiskněte tlačítko Vlastnosti multičár.
4. Určete vzdálenost elementů multičáry.
5. Vyberte volbu Zobraz klouby.
6. Určete vlastnosti počátečního bodu multičáry v nadpise Zakončení.
7. Určete vlastnosti koncového bodu multičáry v nadpise Zakončení.
8. Vyberte volbu Ano z nadpisu Vyplnění.
9. Stiskněte tlačítko OK v dialogovém panelu Vlastnosti multičár.
10. Zadejte název stylu multičáry v poli Jméno.
11. Zadejte popis v poli Popis.
12. Stiskněte tlačítko Přidat.
13. Stiskněte tlačítko Uložit.
14. Stiskněte tlačítko OK v dialogovém panelu Styly multičár.

Funkce tlačítek

Tlačítko Přidat umožňuje přidávat styl multičáry do výkresu.

Tlačítko Vlastnosti multičár umožňuje určovat vlastnosti multičáry.

Volby v menu

Volba Zobraz klouby umožňuje zobrazovat vrcholy multičáry.

Volba Ano umožňuje zobrazovat barvu pozadí.

Czech: personal style (indicative)

Shrnutí

Program umožňuje přidávat styl multičáry do výkresu. Dále styl multičáry program umožňuje ukládat do souboru.

Dlouhé instrukce

Definování stylu multičáry ve výkrese

-
  - *Vybereme volbu Styly multičár z nabídky Data.*
    - ***Windows*** *Vybereme volbu Styly multičár z plovoucí nabídky Styly multičár na nástrojovém panelu Data.*
    - ***DOS a Unix*** *Vybereme volbu Styly multičár z nabídky Data.*

  Objeví se dialogový panel Styly multičár.
  - Nejprve určíme vlastnosti multičáry.
    -
      1. Stiskneme tlačítko Vlastnosti multičár.
         Objeví se dialogový panel Vlastnosti multičár.
      2. Určíme vzdálenost elementů multičáry.
      3. Pro zobrazení vrcholů multičáry vybereme volbu Zobraz klouby.
      4. Určíme vlastnosti počátečního bodu multičáry v nadpise Zakončení.
      5. Určíme vlastnosti koncového bodu multičáry v nadpise Zakončení.
      6. Pro zobrazení barvy pozadí vybereme volbu Ano z nadpisu Vyplnění.
      7. Stiskneme tlačítko OK v dialogovém panelu Vlastnosti multičár.

    Zmizí dialogový panel Vlastnosti multičár.
  - Potom styl multičáry uložíme do souboru.
    -

- Nejprve zadáme název stylu multičáry v poli Jméno. Potom zadáme popis v poli Popis. Pro přidání stylu multičáry do výkresu stiskneme tlačítko Přidat. Nakonec stiskneme tlačítko Uložit.

Nakonec stiskneme tlačítko OK v dialogovém panelu Styly multičár.
Zmizí dialogový panel Styly multičár.

Krátké instrukce

Definování stylu multičáry ve výkrese

2. :

a. **Windows** :

1. Vyberete volbu Styly multičár z plovoucí nabídky Styly multičár na nástrojovém panelu Data.

b. **DOS a Unix** :

0. Vyberete volbu Styly multičár z nabídky Data.

3. Stisknete tlačítko Vlastnosti multičár.
4. Určíte vzdálenost elementů multičáry.
5. Vyberete volbu Zobraz klouby.
6. Určíte vlastnosti počátečního bodu multičáry v nadpise Zakončení.
7. Určíte vlastnosti koncového bodu multičáry v nadpise Zakončení.
8. Vyberete volbu Ano z nadpisu Vyplnění.
9. Stisknete tlačítko OK v dialogovém panelu Vlastnosti multičár.
10. Zadáte název stylu multičáry v poli Jméno.
11. Zadáte popis v poli Popis.
12. Stisknete tlačítko Přidat.
13. Stisknete tlačítko Uložit.
14. Stisknete tlačítko OK v dialogovém panelu Styly multičár.

Funkce tlačítek

Tlačítko Přidat umožňuje přidávat styl multičáry do výkresu.

Tlačítko Vlastnosti multičár umožňuje určovat vlastnosti multičáry.

Volby v menu

Volba Zobraz klouby umožňuje zobrazovat vrcholy multičáry.

Volba Ano umožňuje zobrazovat barvu pozadí.

Czech: impersonal style

Shrnutí

Program umožňuje přidávat styl multičáry do výkresu. Dále styl multičáry program umožňuje ukládat do souboru.

Dlouhé instrukce

Definování stylu multičáry ve výkrese

- *Vybere se volba Styly multičár z nabídky Data.*
  - ***Windows*** *Vybere se volba Styly multičár z plovoucí nabídky Styly multičár na nástrojovém panelu Data.*
  - ***DOS a Unix*** *Vybere se volba Styly multičár z nabídky Data.*

Objeví se dialogový panel Styly multičár.

- Nejprve se určí vlastnosti multičáry.
  1. Stiskne se tlačítko Vlastnosti multičár.
     Objeví se dialogový panel Vlastnosti multičár.
  2. Určí se vzdálenost elementů multičáry.
  3. Pro zobrazení vrcholů multičáry se vybere volba Zobraz klouby.
  4. Určí se vlastnosti počátečního bodu multičáry v nadpise Zakončení.
  5. Určí se vlastnosti koncového bodu multičáry v nadpise Zakončení.
  6. Pro zobrazení barvy pozadí se vybere volba Ano z nadpisu Vyplnění.
  7. Stiskne se tlačítko OK v dialogovém panelu Vlastnosti multičár.

  Dialogový panel Vlastnosti multičár zmizí.
- Potom se styl multičáry uloží do souboru.
  - Nejprve se zadá název stylu multičáry v poli Jméno. Potom se zadá popis v poli Popis. Pro přidání stylu

multičáry do výkresu se stiskne tlačítko Přidat.
Nakonec se stiskne tlačítko Uložit.

Nakonec se stiskne tlačítko OK v dialogovém panelu Styly multičár.
Dialogový panel Styly multičár zmizí.

Krátké instrukce

Definování stylu multičáry ve výkrese

2. :

a. **Windows** :

1. Vybere se volba Styly multičár z plovoucí nabídky Styly multičár na nástrojovém panelu Data.

b. **DOS a Unix** :

0. Vybere se volba Styly multičár z nabídky Data.

3. Stiskne se tlačítko Vlastnosti multičár.

4. Určí se vzdálenost elementů multičáry.
5. Vybere se volba Zobraz klouby.
6. Určí se vlastnosti počátečního bodu multičáry v nadpise Zakončení.
7. Určí se vlastnosti koncového bodu multičáry v nadpise Zakončení.
8. Vybere se volba Ano z nadpisu Vyplnění.
9. Stiskne se tlačítko OK v dialogovém panelu Vlastnosti multičár.
10. Zadá se název stylu multičáry v poli Jméno.
11. Zadá se popis v poli Popis.
12. Stiskne se tlačítko Přidat.
13. Stiskne se tlačítko Uložit.
14. Stiskne se tlačítko OK v dialogovém panelu Styly multičár.

Funkce tlačítek

Tlačítko Přidat umožňuje přidávat styl multičáry do výkresu.

Tlačítko Vlastnosti multičár umožňuje určovat vlastnosti multičáry.

Volby v menu

Volba Zobraz klouby umožňuje zobrazovat vrcholy multičáry.

Volba Ano umožňuje zobrazovat barvu pozadí.

Russian: personal style

Обзор

Программа позволяет добавлять стиль в рисунок.

Также программа позволяет сохранять стиль в файле.

Полный текст инструкции

Определение стиля мультилинии в рисунке

- *Откройте диалоговое окно Multiline Styles..*
    - ***Windows*** *Выберите пункт Multiline Styles в палитре Multiline Styles на панели инструментов Data.*
    - ***DOS / Unix*** *Выберите пункт Multiline Styles в меню Data.*

  На экране появится диалоговое окно Multiline Styles.
    - Укажите свойства мультилинии.
        - 1. Нажмите кнопку Multiline Properties.
             На экране появится диалоговое окно Multiline Properties.
          2. Укажите смещение элементов.
          3. Выберите пункт Display joints , чтобы показать вершины мультилинии.
          4. Укажите свойства начальной точки мультилинии в разделе Caps.
          5. Укажите свойства конечной точки мультилинии в разделе Caps.
          6. Выберите пункт ON в разделе Fill , чтобы показать цвет фона.
          7. Нажмите кнопку OK в диалоговом окне Multiline Properties.

        Диалоговое окно Multiline Properties исчезнет с экрана.
    - Потом сохраните стиль мультилинии в файле.
        - Введите имя стиля мультилинии в поле Name. Потом введите описание в поле Description. Нажмите кнопку

Add , чтобы добавить стиль мультилинии в рисунок. Наконец нажмите кнопку Save.

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

Краткий текст инструкции

Определение стиля мультилинии в рисунке

1.
    a. Для ОС **Windows**:
        1. В палитре Multiline Styles на панели инструментов Data выберите пункт Multiline Styles.
    b. Для ОС **DOS / Unix**:
        1. В меню Data выберите пункт Multiline Styles.
2. Нажмите кнопку Multiline Properties.
3. Укажите смещение элементов мультилинии.
4. Выберите пункт Display joints.
5. В разделе Caps укажите свойства начальной точки мультилинии.
6. В разделе Caps укажите свойства конечной точки мультилинии.
7. В разделе Fill выберите пункт ON.
8. В диалоговом окне Multiline Properties нажмите кнопку OK.
9. В поле Name введите стиль.
10. В поле Description введите описание.
11. Нажмите кнопку Add.
12. Нажмите кнопку Save.
13. В диалоговом окне Multiline Styles нажмите кнопку OK.

Справка объектов

Кнопки

Кнопка Add позволяет добавить стиль в рисунок.

Кнопка Multiline Properties позволяет указать свойства мультилинии.

Пункты

Пункт Display joints позволяет показать вершины мультилинии.

Пункт ON позволяет показать цвет фона.

Пункт Save позволяет сохранить рисунок.

Russian: impersonal style

Обзор

Программа позволяет добавлять стиль в рисунок.

Также программа позволяет сохранять стиль в файле.

Полный текст инструкции в безличном стиле

Определение стиля мультилинии в рисунке

- 1. *Открывается диалоговое окно Multiline Styles..*
     - ***Windows*** *В палитре Multiline Styles на панели инструментов Data выбирается пункт Multiline Styles.*
     - ***DOS / Unix*** *В меню Data выбирается пункт Multiline Styles.*

   На экране появится диалоговое окно Multiline Styles.
   2. Указываются свойства мультилинии.
      - Нажимается кнопка Multiline Properties.
        На экране появится диалоговое окно Multiline Properties.
      - Указывается смещение элементов.
      - Выбирается пункт Display joints , чтобы показать вершины мультилинии.
      - В разделе Caps указываются свойства начальной точки мультилинии.
      - Указываются свойства конечной точки мультилинии.
      - Выбирается пункт ON в разделе Fill , чтобы показать цвет фона. В диалоговом окне Multiline Properties нажимается кнопка OK.
        Диалоговое окно Multiline Properties исчезнет с экрана.
   - В файле сохраняется стиль мультилинии.
   - В поле Name вводится имя стиля мультилинии. В поле Description вводится описание. Нажимается кнопка Add , чтобы в рисунок добавить стиль мультилинии. Нажимается кнопка Save.
- В диалоговом окне Multiline Styles нажимается кнопка OK.

Диалоговое окно Multiline Styles исчезнет с экрана.

Краткий текст инструкции

Определение стиля мультилинии в рисунке

1.
    a. Для ОС **Windows**:
        1. В палитре Multiline Styles на панели инструментов Data выберите пункт Multiline Styles.
    b. Для ОС **DOS / Unix**:
        2. В меню Data выберите пункт Multiline Styles.
2. Нажмите кнопку Multiline Properties.
3. Укажите смещение элементов мультилинии.
4. Выберите пункт Display joints.
5. В разделе Caps укажите свойства начальной точки мультилинии.
6. В разделе Caps укажите свойства конечной точки мультилинии.
7. В разделе Fill выберите пункт ON.
8. В диалоговом окне Multiline Properties нажмите кнопку OK.
9. В поле Name введите стиль.
10. В поле Description введите описание.
11. Нажмите кнопку Add.
12. Нажмите кнопку Save.
13. В диалоговом окне Multiline Styles нажмите кнопку OK.

Справка объектов

Кнопки

Кнопка Add позволяет добавить стиль в рисунок.

Кнопка Multiline Properties позволяет указать свойства мультилинии.

Пункты

Пункт Display joints позволяет показать вершины мультилинии.

Пункт ON позволяет показать цвет фона.

Пункт Save позволяет сохранить рисунок.