COMPUTER SUPPORT FOR AUTHORING MULTILINGUAL SOFTWARE DOCUMENTATION[1]

Donia R. Scott
Information Technology Research Institute
University of Brighton

At the meeting of Translation and the Computer in 1993, I presented a paper on the beginnings of a new approach to computer assisted production of parallel instructions in multiple languages: *multilingual document drafting*. This paper presents the system that we have developed at the University of Brighton to put these ideas into practice (DRAFTER), and its evaluation by professional technical translators.

INTRODUCTION

The DRAFTER system provides an environment for the author of software documentation to construct a model of the *knowledge* to be conveyed in the document and to have draft manuals generated automatically in multiple languages: currently English and French. The system is relevant to both authoring and translating. However, it brings the rôle of the translator closer to that of author, in that it does not involve a 'source' document[2].

DRAFTER is founded on the following principles, based on the results of a user requirements analysis carried out with technical authors and translators specialising in software documentation [3]:

- *Support for knowledge reuse:* Authors and translators spend much of their time acquiring the content of the document to be produced, either from the software developers, through experience with the product, or from an existing document. They often encounter difficulties in accessing required information, primarily because this information is not available in a manner that is useful for their task. A relevant, formal model of the the product would allow them to structure this information in a useful way, re-examine or re-use it later (e.g., for another type of manual of the same product or for a new manual for a similar product), and share it with their colleagues.
- *Support for early drafts in several languages:* Authoring and translating take time, and delays in the production of manuals can be critical to the market success of the product. The ability to generate early drafts would help to speed up the authoring process (including early identification of gaps in the required knowledge), thus reducing the time-to-market.
- *Support for maintaining consistency when making changes:* When parts of a document need changing, either because it was previously specified incorrectly or because the product has changed, keeping track of all the necessary changes in the document can

[1]The work presented here is the result of the efforts a team of researchers, including also: Dr Roger Evans, Mr Markus Fischer, Ms Louise Gorman, Prof Tony Hartley, Dr Cécile Paris, Dr Lyn Pemberton, Dr Richard Power and Dr Keith Vander Linden.

[2]An extensive discussion of the distinction between multilingual generation and translation is provided in [2].

be a difficult and tedious task. The facility to specify each change once, and for it to be automatically propagated to all the relevant instances in the document, would make the process easier and less tedious, and would lead to greater consistency.

- *Support for terminological and stylistic accuracy and consistency:* Documents are often produced by more than one author, and invariably reproduced in other languages by different translators. Many organisations impose a house-style – a set of constraints on the terminology and style of their documents. Maintaining consistency over these constraints within and between documents can be problematic, and writers want a tool that can help them to overcome this.
- *Support for producing alternative formulations:* Authors and translators find it useful to be able to select among alternative ways of expressing the same idea and would welcome support for this aspect of their task.

THE **DRAFTER** SYSTEM

DRAFTER comprises two tools: a *specification tool*, for building a model of the knowledge to be embodied in the documentation, and a *drafting tool*, which converts these models into texts in English and French.

Building the knowledge model for a document

The life of any multilingual manual produced by DRAFTER begins with the construction of a knowledge-base containing a formal model of the software product the writer wishes to document. This facility is provided through a window-based interface for recording the user-oriented information the writer acquires during the knowledge-acquisition phase of the documentation task. To prevent the author from losing orientation, DRAFTER also provides a *Knowledge Visualiser*, for viewing and manipulating the model graphically. DRAFTER functions are invoked through menus and mouse-sensitive objects.

Each model is a collection of entities representing the information commonly found in software manuals: actions, states, objects, and the relations among them. Authors use these concepts and relations to build the procedures that the user of the particular software product must follow. These procedures are called *methods* and contain the following elements (the examples refer to a word processing package):

goal: the objective of the procedure (e.g., delete a word);

sub-actions: the steps to achieve the goal (e.g., click on the "Cut" button);

preconditions: the conditions that must hold for the action(s) to achieve the goal (e.g., the word must be selected);

side-effects: the indicators that the procedure has been successfully carried out (e.g., the word disappears from the screen);

cancellations: ways to cancel procedures (e.g. click on the "Undelete" button).

The elements of procedures are created and edited in one of two ways[3]:

[3]A third way is for elements to be created automatically by links between DRAFTER and an interface design tool; I will not be discussing this facility here.

- *using an interface to a Message Specification Language (MSL):* MSL is a very restricted subset of English or French. The interface allows the author to construct MSL sentences through cascading pop-up menus. For example, the MSL sentence for the action of "Schedule an appointment" for a diary-management package would be *reader schedule arbitrary appointment.* Figure 1 shows how this sentence is constructed with the MSL interface.

- *dragging and dropping existing instances from the knowledge base:* As elements are constructed, they appear both graphically and in a scrolling window; they can be selected using the mouse and then dragged and dropped into empty method slots. Figure 2 shows an example of this facility.

Generating multilingual drafts

Authors can generate multilingual drafts at any point in the construction of a model[4]. The result of the drafting process is congruent texts in English and French, appropriate to the genre of instructions. These texts are generated independently from the model — i.e. *without* translation from one language to another. A sample output is shown in Figure 3.

The draft text appears in a special window and is mouse-sensitive, allowing the author to access the knowledge base entry for selected parts of the text. In this way the author can modify the underlying knowledge while working from the text; this modification will be evidenced throughout the draft and in both language versions. The author can request alternative expressions of all or selected parts of the text.

The draft text can also appear in a separate text editor window; this is useful for those cases where the author decides to modify the text rather than the underlying knowledge. However, changes made in this environment are not reflected in the knowledge base and cannot be propagated thoughout the document or to the other language.

USER EVALUATION

We have performed a user evaluation analysis with two professional technical translators in the area of software documentation [4]. The evaluation session took place over two days. Following training on the system, users were asked to model and generate a specified set of instructions. They then evaluated (a) the Specification Tool (i.e., the usability of the interface), (b) the Drafting Tool (i.e., the quality of a set of output texts) and (c) the potential for exploitation (i.e., the general utility of the DRAFTER concept and prototype for technical translators).

Evaluation of the Specification Tool

Conceptual representation of procedures. The ease of use of any system, particularly in the learning phase, is affected by how 'natural' the concepts being manipulated are; it is also affected by how 'natural' the visual representation of these concepts by the system is to the user.

Before introducing the evaluators to DRAFTER, we gave them a short tutorial on an imaginary software product and asked them to show us what their first step in documenting the instructions for a selected use of the product would be like. We then compared this with the representation used by the system. We found that the chosen representation of procedures in DRAFTER (i.e., goals,

[4]We have found that users tend not to wait until they have built a complete model of the task before generating but instead generate when they have completed individual procedures; this suggests that text drafting is also being used as a means of verifying the model as it is being built.

actions, side-effects etc.) is a natural one for users: these elements were invariably marked in their initial 'sketch' document.

Visualisation of procedures. In addition to the multilingual text describing a given procedure in the model, DRAFTER provides two other types of representations of the procedural model:

- *a graphical representation of the model:* Procedures are visualised on the screen as nested boxes. The outermost box represents the goal of the procedure; within this are boxes for the other elements. The more complex the procedure, the deeper the nesting shown. For example, a precondition (e.g. 'select a word') of one procedure can itself become the goal of another procedure (e.g., where the sub-actions are 'move to the beginning of the word' and 'double-click the mouse button', and the side-effect is that the word is highlighted on the screen).
- *a list of actions as they are constructed:* This list appears in a separate window and is given a rough structure through the use of indentation (e.g., for the *deleting a word* procedure, the action list would show 'select the word' and 'click on the CUT button' indented relative to 'delete a word');

The type of graphics DRAFTER uses to display the model as it is built up was at times confusing to the user. The evaluators commented that this problem would probably be overcome with more experience on the system. They thought that users would probably tend to concentrate more on the action list during early stages of using the system, shifting their attention to the graphical representation after they had become more experienced.

Constructing models of procedures. Technical authors often have to write documentation of a system they have never encountered before, without the support of a full demonstration from the developers. In carrying out the evaluation, we tried to partially simulate this scenario, which we felt would provide a stringent test of the system.

The evaluators were given a short training manual of DRAFTER, which they were asked to work through; this involved completing a number of set examples of model building and draft generation. Both evaluators were able to work through the manual within an hour.

Using the system in a more realistic setting, the evaluators said that training on DRAFTER would be essential before "letting the user loose". At present, writing documentation involves a strongly text-bound approach, but the task of constructing input models in DRAFTER requires a new skill: the conceptualisation of tasks or instructions. The need for prior training was felt to be particularly important for this aspect of the system.

Evaluation of the Drafting Tool

From the user's point of view, generating a multilingual draft document is a simple matter of selecting the GENERATE TEXT option once a full or partial model has been constructed. To test this part of the system, we asked the evaluators to critique and rate two pairs of congruent instructions in English and French which were generated by DRAFTER (see Figure 3). These texts cover most of the linguistic features of the genre of instructions (revealed from a detailed corpus analysis [1]). The evaluators were asked to focus their criticisms on issues of terminology, style, structure and grammar, and to give an overall grade on the scale:

A High quality: no revision needed
B Good quality: minor revision needed
C Fair quality: substantial revision needed
D Poor quality: needs to be completely rewritten

The English Texts. Both texts were rated B by both evaluators. Below is a list of the revisions they said they would want to make to the draft texts:

Terminology:

- The term *execute*, used in the first step of the Macwrite instructions, is one that users find unhelpful: it is too ambiguous. A better term would have to be substituted.
- In both texts, some instructions requiring operations with the mouse buttons were confusing. Often this was because the instructions failed to mention which (of the three available) mouse buttons should be used. In one case (Calendar Manager) the relevant mouse button was given too technical a label (*the Select mouse button*); more explanation, or a clearer label (e.g., *the left mouse button*) would be needed. The instruction to simply *hold the mouse button down* (Macwrite) was thought to be incomplete: it would be better to add that the user should continue to hold the button down.
- The instruction should distinguish between buttons on the mouse and buttons on the interface, either graphically (see below) or through the use of different terms. For example, the sentence *"Click the Select mouse button on the Insert button"* (Calendar Manager) is confusing.

Style:

- The draft instructions should make use of special fonts, or any other graphical convention, for the names of application buttons and file options, e.g., Save File button, New option (Macwrite), Delete button, Appointment option (Calendar Manager). This would have to be revised; for example:
  "... by clicking on the $\boxed{\text{SAVE FILE}}$ button"
- The initial letters of "start time" and "end time" (Calendar Manager) should be upper case.

Structure:

- Only one structural revision was needed: in the sentence *"Type a name in the Save Current Document As field in the Save As window"*(Macwrite). Both evaluators suggested reversing the clauses to make the sentence read: *"In the Save As window, type a name in the Save Current Document As field"*

Grammar:

- No grammatical revisions were judged to be necessary.

The French Texts. Both texts received high scores from both evaluators: The Macwrite example received an A and a B; the Calendar Manager example received two Bs. The evaluators' suggested revisions were basically the same as for the English, with the following additions:

Terminology:

- One evaluator said that he would probably want to replace *double cliquer* with *cliquer deux fois*[5]

[5] Our corpus analysis shows that the expression *double cliquer* is now well established in the French terminology for software products; however, DRAFTER provides the freedom to choose at the editing stage, or in the definition of the lexicon for a given production job (the latter would be preferred where expressions are part of the house-style).

Structure:

- The sentence *"Ouvrir le menu Fichier en pressant sur le bouton de la souris sur le menu"* (Macwrite) is structurally ambiguous and will therefore need to be changed.

Grammar:

- One of the evaluators commented that it may be better to replace the *"à"* in the sentence *"Sélectionner l'option Nouveau dans le menu Fichier en déplaçant le pointer à l'option"* (Macwrite) by *"vers"*. This view was not supported by the other evaluator, or by the corpus analysis.

Evaluation of the potential for exploitation

This part of the evaluation was carried out by the two technical translators already mentioned and the director of a software house specialising in technology transfer from research prototypes to product development. Here we were primarily interested in the potential of DRAFTER as a commercial product, and in its possible impact on the skills of professional translators.

Benefits of DRAFTER over current methods. The view of the translators was that, compared with MT (Machine Translation) and MAT (Machine Assisted Translation), a tool like DRAFTER will lead to major cuts in project durations, and to improved document management and coordination. They thought that the system would be particularly helpful for producing documents in a given style and with suitable and consistent discourse structures. They also expressed the view that by making technical authors conceptualise the task of carrying out instructions, use of the system will facilitate multilingual document production "in comparison with the laborious business of translating unstructured texts". Although DRAFTER can be used for single-language authoring, it was felt that the main benefit would be to the production of drafts in multiple languages.

Effect on the translation profession. The consensus view was that DRAFTER would not lead to a de-skilling of translators, but to *re-skilling*. The new skill would be the conceptualisation of tasks and their instructions. Already, translators are having to make use of desktop publishing or word processing technology, which involves a new skill of document design; reader-oriented (rather than producer-created) documentation is another new skill. It was thought that DRAFTER will help with these too.

> "Any successful enabling technology accelerates or automates low-skill tasks, freeing skilled staff to concentrate on harder, more demanding tasks, and thus bringing a more effective use of skills. It is too early at this stage to know which of these high-skill areas should receive the effort that has been freed." (Colin Shearer, evaluator)

Suitable users and markets. The original motivation behind DRAFTER was to provide a support tool for technical authors and translators. However, the system may be relevant to others — for example, software developers and interface designers. The evaluators felt that although it will be useful to technical authors and translators, a new type of user would develop (sometimes as a result of the above mentioned re-skilling): a *multilingual document creator*. In the words of one evaluator: "...this will lead to a blurring of the demarcation between technical authors, translators and interface designers, and to authoring perhaps being carried out by interdisciplinary teams involving all these professions." Graphical designers may also be included, although their work is not currently supported by the system.

The general view is that the markets for DRAFTER will probably be global, with localised PC users communicating via the internet, and coordinated by their companies. The immediate market may well be software-system suppliers with a large and dynamic portfolio of products, requiring frequent generation and revision of documents.

The director of the software house thought that there could well be two markets:

1. A product based on the current DRAFTER prototype: a relatively low-cost, high-volume, of-the-shelf product aimed at smaller software suppliers or technical authoring/translation companies.

2. Bespoke versions of DRAFTER with enhancements aimed at specific product lines for different categories of clients (in the computer industry or in other sectors). These would necessarily be of much higher cost because of the large service requirements for customisation.

Potential impact of advances in the computer industry. There was unanimous agreement that there will be radical changes in software (and other) manuals over the next decade. The move will be aware from software applications that are driven through simple user interface components towards true multi-media interfaces. Manuals, in whatever form they take, will have to reflect this: they will become multi-media, and delivered on-line. Increasingly, they will make greater use of graphics. However, the need for text will remain; text will be an important component in the multi-media delivery of instructions and book-style manuals will continue to be needed.

Range of applications. It was felt that DRAFTER could be usefully applied to a wide range of areas other than software: from educational toys to pharmaceuticals, precision engineering, engineering diagnostics — indeed to any area where instructions are needed for goal or method oriented systems.

Improvements needed for commercialisation. Given the novelty of the DRAFTER concept, the evaluators stressed the need for training, including a conceptual introduction to the system. The system would also need to be more robust and "idiot-friendly", and include greater flexibility in editing the knowledge model and in customising the environment (e.g. resizing and moving windows and dialogue boxes). Scalability, in terms of document/project management and navigation was also thought to be important. Before DRAFTER could be produced as a general tool (i.e. for use with a range of applications), the ease of including new concepts for another application, and of increasing the number of languages, would also have to be addressed.

Time to market. The evaluators were convinced that there is a current need to produce the kind of texts which DRAFTER supports, and that the system, or one based on it, has immediate commercial applicability. A demonstration prototype is the commercial requirement for any new product, and DRAFTER already provides this.

We asked the evaluators how long they thought it would take to develop DRAFTER from a research prototype to a commercial product. They said that the time needed would depend on the intended scope of the product, and gave 6-18 months as a ballpark figure.

CONCLUSIONS

The DRAFTER system was developed as a proof-of-concept of the application of natural language generation for a support tool for writing multilingual instructions. The result has been positively evaluated by professional translators and by a specialist in the development of software products.

Although improvements are clearly required to make the system more user-friendly, the drafts are of high-quality and the potential for commercialisation is encouraging.

The view arising from the evaluation is that eventual commercialisation of the system will probably require a customer that is large enough to pay for productisation, and far-sighted enough to invest in novel technology. The uptake of a product like DRAFTER will probably be slow at first, but with a large eventual impact on the profession of manual production, be it mono- or multi-lingual.

The DRAFTER research team is currently exploring new directions of the current system. With further funding from the EPSRC, and with new funding from the EC and the Office of Naval Research in the United States, we are now developing extensions to the DRAFTER concept through the addition of three new languages (Bulgarian, Czech and Russian), the inclusion of constraints on layout, content and style, the extension to on-line documentation, and the integration with interface-building tools.

Acknowledgements

The DRAFTER project is funded by the Engineering and Physical Sciences Research Council (EPSRC) on Grant J19221.

References

[1] Hartley, A. and Paris C. (1996). Two Sources of Control over the Generation of Software Instructions. In *Proceedings of the 1996 Meeting of the Association for Computational Linguistics (ACL'96).* Santa Cruz, California, USA.

[2] Hartley, A. and Paris C. (forthcoming). Multilingual Document Production: From Support for Translating to Support for Authoring. *Machine Translation*

[3] Pemberton, L., Gorman, L, Hartley, A and Power, R (1996). Computer Support for Producing Software Documentation: Some Possible Futures. In M. Sharples and T. van der Geest (eds.) *The New Writing Environment: Writers at Work in a World of Technology.* London: Springer-Verlag, pp. 59-72.

[4] Power, R. and Scott, D. (forthcoming). Usability and Commercial Potential of DRAFTER: report of an evaluation by technical translators and software developers. ITRI Technical Report ITRI-96-13.

More information on DRAFTER is available at the following web site:

http://www.itri.brighton.ac.uk/projects/drafter

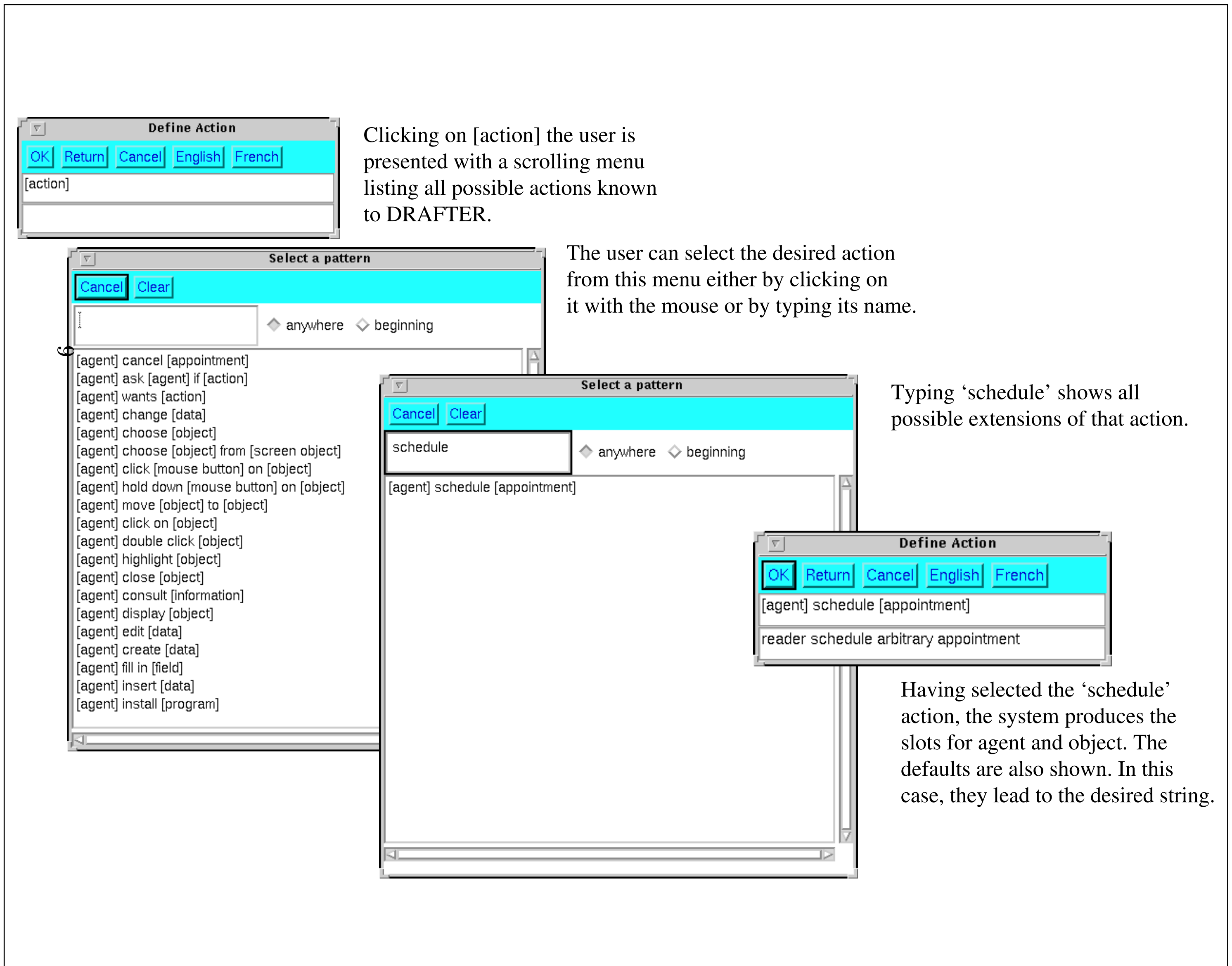

Figure 1: Building information models with the Message Specification Language

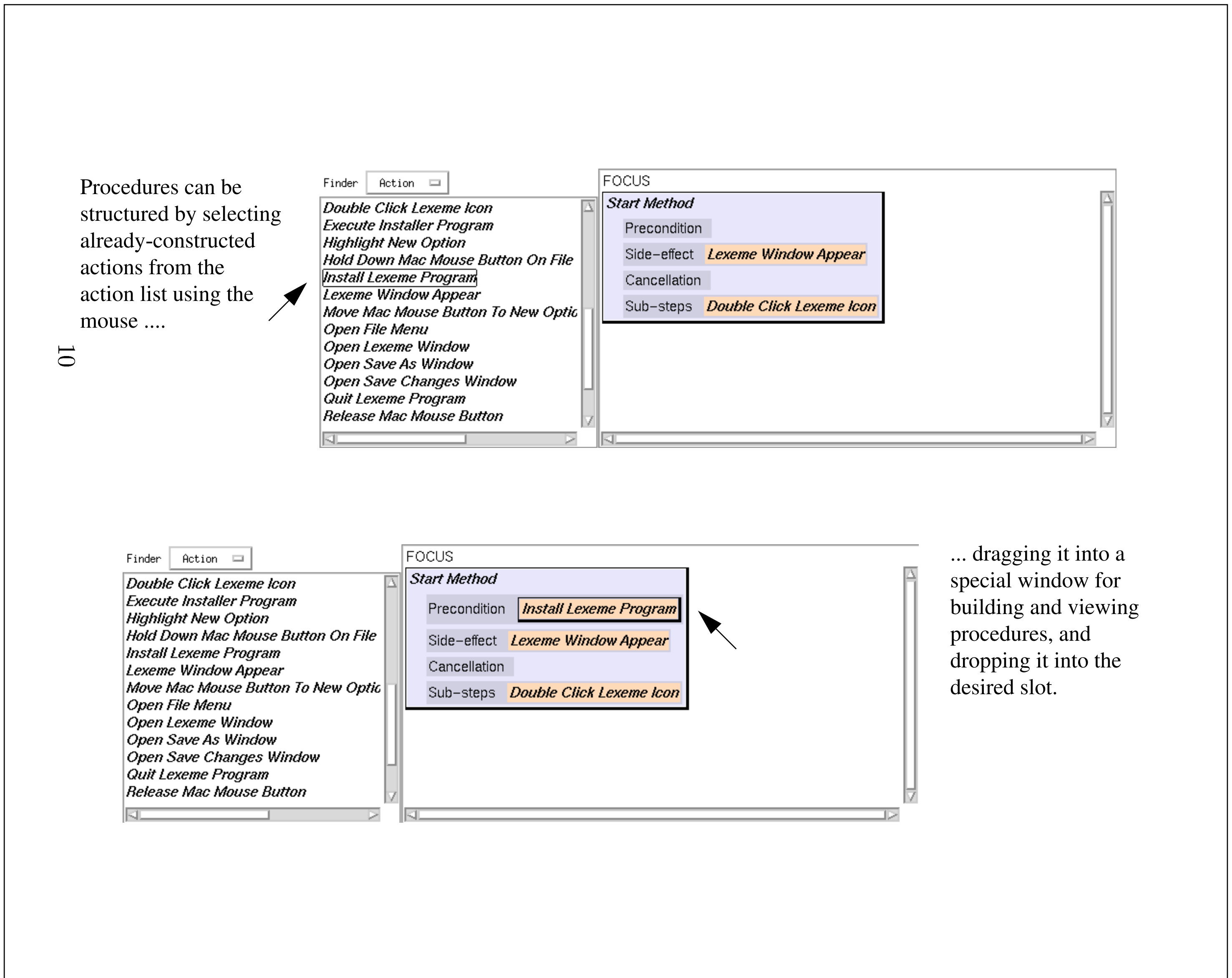

Figure 2: Building information models using the drag-and-drop facility

Macwrite

To start Macwrite
You must install Macwrite by executing Installer.
Double click on the Macwrite icon.
The Macwrite window appears.

To create a document
Choose the New option from the File menu.

To choose the New option from the File menu
1. Open the File menu by holding the mouse button down on the menu.
2. Highlight the New option from the File menu by moving the pointer to the option.
3. Release the mouse button.

To save the document
1. Open the Save As window by choosing the Save option from the File menu or by clicking on the Save File button.
2. Type a name in the Save Current Document As field in the Save As window.
3. Click on the Save button.
You can close the Save As window by clicking on the Cancel button.

To quit Macwrite
1. Choose the Exit option from the File menu.
2. Click on the Yes button.

Lancement de Macwrite
Il est nécessaire d'installer Macwrite en exécutant Installer.
Double cliquer sur l'icône Macwrite.
La fenêtre Macwrite apparaîtra.

Création d'un document
Choisir l'option Nouveau dans le menu Fichier.

Choix de l'option Nouveau dans le menu Fichier
1. Ouvrir le menu Fichier en pressant sur le bouton de la souris sur le menu.
2. Sélectionner l'option Nouveau dans le menu Fichier en déplaçant le pointeur à l'option.
3. Relâcher le bouton de la souris.

Enregistrement du document
1. Ouvrir la fenêtre Enregistrer Sous en choisissant l'option Enregistrer dans le menu Fichier ou en cliquant sur le bouton Enregistrer Document.
2. Introduire un titre dans la zone de texte Enregistrer Le Document dans la fenêtre Enregistrer Sous.
3. Cliquer sur le bouton Enregistrer.
Il est possible de fermer la fenêtre Enregistrer Sous en cliquant sur le bouton Annuler.

Pour quitter Macwrite
1. Choisir l'option Quitter dans le menu Fichier.
2. Cliquer sur le bouton Oui.

Calendar Manager

To schedule an appointment
1. Open the Appointment Editor window by double clicking on the date of the appointment or by choosing the Appointment option from the File menu.
2. Choose the start time, then choose the end time.
3. Type the description of the appointment.
4. Click the Select mouse button on the Insert button.

To cancel an appointment
1. Open the Appointment Editor window by double clicking on the date of the appointment or by choosing the Appointment option from the File menu.
2. Choose the appointment from the Appointments list.
3. Click the Select mouse button on the Delete button.

Insertion d'un rendez-vous
1. Ouvrir la fenêtre Appointment Editor en double cliquant sur la date du rendez-vous ou en choisissant l'option Appointment dans le menu Edit.
2. Choisir l'heure de début, ensuite choisir l'heure de fin.
3. Introduire la description du rendez-vous.
4. Cliquer le bouton de la souris Sélectionner sur le bouton Insert.

Suppression d'un rendez-vous
1. Ouvrir la fenêtre Appointment Editor en double cliquant sur la date du rendez-vous ou en choisissant l'option Appointment dans le menu Edit.
2. Choisir le rendez-vous dans la liste Appointments.
3. Cliquer le bouton de la souris Sélectionner sur le bouton Delete.

FIGURE 3: Pairs of generated instructions used in the evaluation.