

“Do Screen Captures in Manuals Make a Difference?” A comparison between textual and visual manuals

Mark Gellevij, Hans van der Meij, Ton de Jong, and Jules Pieters
University of Twente
Faculty of Educational Science and Technology
P.O.Box 217, 7500 AE
Enschede, The Netherlands

gelevij@edte.utwente.nl
meij@edte.utwente.nl
jong@edte.utwente.nl
pieters@edte.utwente.nl

This study examines the use of screen captures in manuals. Three types of manuals were compared, one textual and two visual manuals. The two visual manuals differed in the type of screen capture that was used. One had screen captures that showed only the relevant part of the screen, whereas the other consisted of captures of the full screen. All manuals contained exactly the same textual information. We examined immediate use on time (use as a job aid) and on learning (use as a teacher). For job-aid purposes, there was no difference between manuals. The visual manual with full screen captures and the textual manual both were better for learning than the visual manual with partial screen captures. We found no effect on user motivation. The tentative conclusion of this study is that screen captures seem not vital for learning or immediate use. If one opts for including screen captures, then the conclusion is that full screen captures are better than partial ones.

Introduction

Nowadays, the use of visuals in user manuals for the computer industry seems to be a must. Manual designers devote much time and resources to creating attractive manuals, often by offering various screen captures throughout their manuals. These screen captures are undoubtedly included for more than merely decorative functions. As a designer, an important question is when (for which type of information), where (at which places in the manual) and how (full or partial) to use screen captures.

Handbooks for designing technical documentation reveal very little about the use of screen captures. Price and Korman (1993) treat the topic in one paragraph, stating that screen captures should be used for two purposes: (1) to show the results of action steps taken, and (2) to show the object to act upon in the next action step. They offer one design guideline that advises the use of callouts to draw the users' attention to key parts of a window. In her book *Dynamics in Document Design*, Schriver (1997) does not cover the specific topic of using screen captures in technical documentation, although she does give several general guidelines on combining the use of words and graphics in document design.

A Contemporary Renaissance: Changing the Way We Communicate

The most extensive discussion comes from Horton (1983). Among other things, Horton questions whether screen captures always have a purpose that justifies their costs. Horton mentions that screen captures offer visual relief on pages full of text. He states that "when used appropriately and placed wisely, they make procedures easier to learn and quicker to follow" (p. 148). What actually is "appropriate" and "wise" is described in the following guidelines (p. 148):

- In tutorials, screen captures should be offered to let the user imagine how to use the system
- Screen captures should be used to let the user verify the display, especially when the target group is the novice computer user
- If only part of the screen is important, only that part should be shown. The pages 'should not be cluttered with what the users already know'.

In short, information, research, or advice about the use of screen captures in technical documentation is limited. Because we believe that designers think primarily in terms of presenting different types of information in a manual, and only thereafter consider their expression in words or graphics, a classification of information types is the starting point for discussion.

A common and valuable classification is the distinction between procedural and conceptual information. Conceptual information offers explanations and supports goal setting. Procedural information supports direct or indirect user actions. It can be divided into action information, error information, and coordinative information (Van der Meij, 1998). Screen captures can be used to support all these information types.

Whether the presence of screen captures improves learning, and speeds up or slows down task execution is a question that has partly been answered by Van der Meij (1998). In a study comparing a visual and a textual manual, he found a highly significant positive effect of screen captures on task execution time. He offers three explanations for this. One, the connection between what is written and what is shown on the screen is now presented in a single source: the manual. Users may thus have fewer difficulties in handling the two separate sources. Two, there is no need for the user to translate the text into an image. Three, fewer switches between manual and screen are needed. Because of the screen captures, the manual becomes more self-contained. In general, these arguments all share the core idea that screen captures reduce coordination problems.

Van der Meij (1998) also mentions some drawbacks of screen captures. One of these is possible user passivity caused by the self-containment of visual manuals. Because there is no need to translate text into an image, exploring the computer system and its interface may be discouraged. Also, the redundancy of screen captures and text may be disadvantageous.

This raises the intriguing question of whether what is gained from using screen captures in speeding up task execution is not lost in learning to use the program. When screen captures speed up task execution, they may simultaneously fail to maximally activate the user in using and exploring the interface to stimulate learning.

Thus, theory seems to give predictions that make it difficult to create a manual that serves both goals. For this study, the main question is whether this prediction holds. In addition, we look at

A Contemporary Renaissance: Changing the Way We Communicate

the design issue of screen captures. More specifically, we examine the role of full screen captures versus partial screen captures (screen shots that display relevant parts of the screen).

Three manuals (tutorials) are compared: a textual manual (Text), a manual supporting procedural information with partial screen captures (V-Part), and a manual that supports procedural information with full screen captures (V-Full). The textual manual followed the minimalist principles and heuristics for design (Van der Meij & Carroll, 1995). This manual formed the basis for the two visual manuals. The manuals were written in English and all contained exactly the same text. Partial screen captures were added to the action steps in the V-Part manual, whose design was inspired by Stuur's visual manuals (Stuur, 1992). Full screen captures were added to the action steps in the V-Full manual, whose design was inspired by the 'Visual learning guide' manuals by Gardner and Beatty (1993). To avoid differences in reading, we attempted to keep the layout of the three manuals as similar as possible. Example pages of the three manuals can be found in Appendix A (Text), B (V-Part) and C (V-Full).

The main goal of the study was to find out if these manuals have a different effect on learning and task execution. To find out whether the visual manuals gave visual relief and were viewed as more attractive, user motivation was measured.

Method

Subjects

Seventy-three Dutch students from the Faculty of Educational Science and Technology participated in the experiment. The mean age of the experimental group was 21.2 years (SD=2.4 years). The subjects were classified as intermediate or experienced computer users. For this purpose, part of the Computer Self-Efficacy Scale-questionnaire was translated into Dutch (Murphy et al., 1989). Subjects were randomly assigned to one of the three experimental conditions Text, V-Part, or V-Full. Table 1 shows how the subjects were assigned exactly.

Manual	Computer experience		Row total (m/f)
	Intermediate (m/f)	Experienced (m/f)	
Text	10 (0/10)	15 (3/12)	25 (3/22)
V-Part	10 (1/9)	15 (5/10)	25 (6/19)
V-Full	9 (1/8)	14 (6/8)	23 (7/16)
Column total	29 (2/27)	44 (14/30)	73 (16/57)

m = male, f = female

Table 1. Distribution of subjects per condition

A Contemporary Renaissance: Changing the Way We Communicate

Materials

Materials - Computers

The sessions were held in a computer room with 20 IBM compatible Pentium Pro 166 computers with 32 MB of RAM. During the experiment, all subject actions with the computer program were logged automatically.

Materials - SIMQUEST and Motion application

The subjects used the SIMQUEST 1.1 authoring tool. SIMQUEST lets teachers design and develop a multimedia learning environment based on simulations. The subjects for this experiment were students of the Faculty of Educational Science and Technology. A main component in their study is learning how to systematically design instruction using various media. As SIMQUEST is a state-of-the-art tool designing multimedia instruction, we expected that the subjects would be very interested in learning to use it.

An application about Motion was used as the basis for the exercises in the manual. These were about modifying and creating simulations, assignments, and explanations.

Materials - Manuals

During practice, each subject received one of the three manuals (Text, V-Part, or V-Full). Each chapter in the manual consisted of two sections: (1) a guided section, in which users could simply follow the detailed action steps to accomplish a certain task, and (2) an explorative section, which offered one or more exercises comparable to the task practised in the guided section. As encouraging exploration is one of the main heuristics in minimalist designs (Van der Meij & Carroll, 1995), explorative parts are an important feature in this type of tutorial. Time used on the guided parts shows a direct effect of manual type on execution of action steps. Time used on explorative parts shows which manual encourages users more to engage in exploration.

Materials - Questionnaires and tests

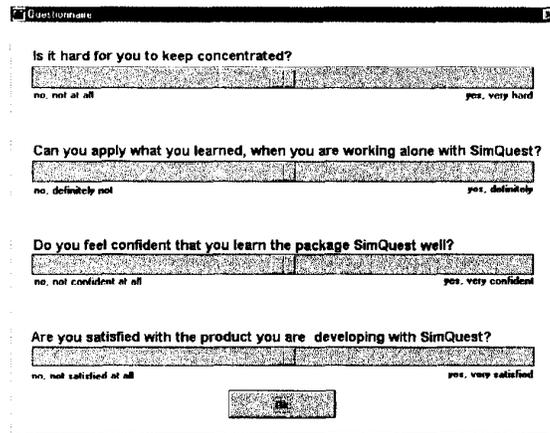
At the start of the practice session, the subjects received a questionnaire with general questions about gender, age, and their previous experience with authoring tools. In addition, there were 20 questions to classify the subjects as intermediate or experienced computer users.

During practice, every 15 minutes, a pop-up screen (Figure 1) appeared asking the subject four questions about their current motivational state. These four questions were based on Keller's ARCS theory (1983).

Two tests were used: a recency test and a post-test. Both tests contained four exercises. The exercises, or parts of these exercises, had two levels of difficulty:

- trained tasks, parts of the exercises that were exactly the same as practised with the manual,
- untrained (transfer) tasks, new tasks that were different than practised tasks.

A Contemporary Renaissance: Changing the Way We Communicate



The screenshot shows a window titled "Questionnaire" with four questions, each followed by a horizontal scale from 1 to 5. The questions are:

- 1. "Is it hard for you to keep concentrated?" with anchors "no, not at all" and "yes, very hard".
- 2. "Can you apply what you learned, when you are working alone with SimQuest?" with anchors "no, definitely not" and "yes, definitely".
- 3. "Do you feel confident that you learn the package SimQuest well?" with anchors "no, not confident at all" and "yes, very confident".
- 4. "Are you satisfied with the product you are developing with SimQuest?" with anchors "no, not satisfied at all" and "yes, very satisfied".

At the bottom center of the window is a small button labeled "OK".

Figure 1. Pop-up motivation questionnaire

Procedure

The experiments consisted of three sessions: practice, a recency test, and a post-test.

The *practice session* lasted a maximum of 4 hours. At the start of this session the subjects were told that their task was to individually get to learn to work with SimQuest. The manual was their only source of information. They were further told that they could stop practising when they felt they mastered SimQuest.

The *recency test session* took place at the same day and lasted a maximum of 2 hours.

The *post-test session* took place one week after the first two sessions. The post-test and accompanying introduction were the same as for the recency test, except for the topics.

Coding and scoring

Coding and scoring - Computer experience

The questionnaire on computer experience used a 5-point disagree-agree scale. Means were calculated. Subjects with a mean above 3 were classified as experienced users, 3 or below as intermediates.

Coding and scoring - Time

During practice, all subjects' actions were logged. These logs allowed us to determine how long the subjects worked on each chapter or part of a chapter.

Coding and scoring - Motivation

The data of the motivation pop-up questionnaire consisted of a maximum of 12 repeated measures. The first measurement was removed because it was used for practice. After the ninth measurement the number of subjects that answered the questionnaire dropped below 84%. Therefore, the measurements t2 to t9 were used in the analysis.

Coding and scoring - Learning effects

Each test-element was classified as trained or untrained. Next, it was scored if it was performed correctly (1 point) or incorrectly (0 points).

A Contemporary Renaissance: Changing the Way We Communicate

Results

Results - Time

No statistically significant differences for time during practice were found on *guided sections* of the manuals. One explanation is that the texts already provided sufficient coordinative information. Because the action steps clearly explain *what* to do and *where* to act, the screen captures may have been redundant, offering no vital or new information. Another reason might be the transparency of the interface. The interface may have been so easy to use, that (extra) coordinative information was not necessary at all.

Yet a third explanation may lay in the specific content of the guided sections. As these sections contain procedural (doing) as well as conceptual (reading) information, examining the results on only the procedural information by filtering out all reading time from the current data, may shed a better light on differences between speed of task execution.

Differences between the three conditions on time used on *explorative sections* were quite large. Almost twice as much time was used by subjects that used the Text manual (1133 seconds (SD=834)) than users of the V-Part manual (596 seconds, SD=597)), which is a statistical significant difference ($F(2,72)=3.37, p<0.05$). These results corroborate the findings for motivation (see next section). As users of the V-Part manual seemed less motivated than the other groups, and were even getting less motivated, it may be fair to assume that they did not feel encouraged to carry out the (optional) explorative parts.

Results - Motivation

No significant differences were found on motivation. Despite this, the results consistently favour the V-Full manual (see Table 2). Apparently, this type of manual does indeed offer the visual relief that it is supposed to offer compared to the Textual manual. Surprisingly, there are no clear differences between the V-Part and Text manual. There seem to be no extra benefits in offering partial screen captures in comparison with text only. On the contrary perhaps. As motivation slightly drops over time using the V-Part manual, it may well be that partial screen captures tend to de-motivate.

	Order of conditions		
	V-Part	Text	V-Full
Attention (being concentrated)	64.21 (25.36)	67.59 (20.44)	70.65 (18.84)
Relevance (feeling of performing meaningful tasks)	64.31 (22.49)	66.19 (13.57)	69.38 (13.84)
Confidence (confidence in quality of performance)	67.98 (11.58)	68.91 (17.17)	72.23 (10.98)
Satisfaction (satisfied with developed products)	62.89 (9.95)	63.02 (19.92)	66.86 (11.89)
Motivation (combination of the four factors)	65.11 (17.05)	66.16 (11.21)	69.78 (11.94)

Table 2. Means (standard deviations) of motivational factors (scale 0 - 100)

A Contemporary Renaissance: Changing the Way We Communicate

Results - Learning effects

All subjects, regardless of the manual with which they had practised, performed very well on the items that measured *trained tasks*. On the recency test as well as the post-test, more than 87% of the tasks were performed correctly. This ceiling effect is troublesome because it strongly limits the chances of finding any significant differences on trained tasks.

The *untrained tasks* were performed somewhat less well (see Table 3). Both the V-Full and the Text manual outperformed the V-Part manual on the post-test. The difference between V-Full and V-Part suggests that a better mental model is built by the V-Full users. They have gained a better understanding of the system from the full screen captures. The obvious question then is what can explain the difference between the Text and V-Part manual. The Text manual may force users to more actively explore the system. The results on time, where significantly more time was spent on explorative parts by Text users than V-Part users, could validate this explanation. Apart from devoting more time, it could also be that the partial screen captures interfere with the mental model being built by the users itself. In that case, partial screen captures may not support, but in fact may obstruct such a construction.

	Recency test	Post test
Text	14.68 (5.13)	15.40 (4.76) *
V-Part	13.36 (5.44)	11.96 (4.86)
V-Full	14.87 (4.30)	16.00 (3.88) *

* $p < 0.05$ with regard to V-Part

Table 3. Test-scores on untrained tasks (max. 23)

Discussion

The experiment does not make it perfectly clear whether screen captures are a necessary feature in manuals. Looking at the results, there is proof that a design in which partial screen captures are coupled to action steps is *not* a good solution. On several measures, the subjects who had worked with the V-Part manual, performed worse than the other subjects. When the V-Full and Text manual are compared, there is no proof that one leads to better learning so far. Also, in time used on guided and explorative parts, no differences were found between the V-Full and Text manual. The use of full screen captures suggests a motivating influence. The experiment has not proven the assertion, however.

One might conclude that devoting much time and resources on presenting screen captures in manuals is not worth the effort. A closer look at the experiment cautions against such a conclusion. An important premise for this experiment was that we wanted to have a situation that was close to reality. It can be argued whether individually learning a computer system with a manual as only source of information, and for three hours in a row, is a realistic situation. Learning a computer system at home or at work may go quite differently. It may take 4 half hour sessions over a period of two weeks instead of one long session. Using a visual manual instead of a textual one in this case, where you have to re-start several times, may then have its benefits.

A Contemporary Renaissance: Changing the Way We Communicate

Also, the subjects that participated in this experiment may not represent computer users in general. As students at our faculty must and do use computers quite a lot, their computer experience is probably higher than that of regular computer users. For real novices, differences between using textual and visual manuals may again be stronger.

Looking at the type of manual itself used in the experiment, a few remarks can be made. It was surprising to see that there were no differences found in time on task. Especially because Van der Meij (1998) found quite strong effects. This may very well be explained by the type of manual used in both experiments. Van der Meij used a reference guide that consisted almost completely of procedural information whereas the tutorial in this experiment was a balanced combination of conceptual and procedural information. As the focus of a tutorial is primarily on learning, or better, in getting to understand the system by doing, reading, and exploring, less gain in time can be expected.

A second remark pertains to the quality of the manual. The results on learning effects show that subjects were very capable of performing tasks that they had trained and even those they did not train, both in the recency and the post test. These results may indicate that the manual did its job well in teaching the subjects how to use the system. Too well, perhaps?! If this is indeed the case, additional benefits from screen captures can only be small.

Finally, the two designs of the visual manuals were in a way unsophisticated. Within the manuals, only one type of screen capture was used, full or partial. A consistent approach was chosen, meaning that each action step was supported with a screen capture.

As screen captures may have very different functions at different places in procedures, for example, to support focusing attention at the start of a procedure, identifying and locating during the performance of the procedure, and verifying at the end of a procedure, effects may be levelled by neutral to negative effects for another. A visual manual in which screen captures are presented in such a way that their role and design are optimally tuned to one another may function much better than the current visual (or textual) manuals.

Quite often manuals not only serve the purpose of instructing and supporting users. Visually attractive manuals can very well be part of the selling strategy for software packages, which could be a legitimate reason to choose for a visual manual. If the marketing department takes the decision, they should, as this experiment suggests, opt for presenting full rather than partial screen captures.

Acknowledgement

We would like to thank Dave Farkas for his valuable help on the study and the review of this paper.

References

Gardner, D.C., and G.J. Beatty. *Visuele leermethode Windows 3.1* [Windows 3.1: The visual learning guide]. Utrecht: A.W. Bruna Uitgevers B.V. 1994.

Horton, W. Visual Literacy - Dump the dumb screendump. *Technical Communication*. 40: 146-147, 1983.

A Contemporary Renaissance: Changing the Way We Communicate

Keller, J.M. Motivational Design of Instruction. In: Reigeluth, C.M. (Ed). *Instructional-design Theories and Models* (pp. 383-434). Hillsdale, New Jersey: Lawrence Erlbaum Associates. pp. 383-434, 1983.

Murphy, C.A., D. Coover, and S.V. Owen. "Development and validation of the computer self-efficacy scale. *Educational and Psychological Measurement.*" 49: 893-899, 1989.

Price, J., and H. Korman. *How to communicate technical information*. Redwood City, CA: Benjamin/Cummings. 1993.

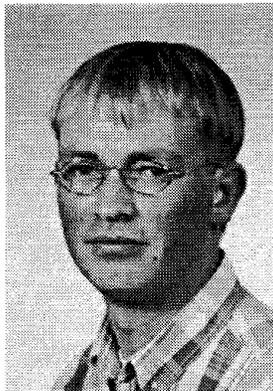
Schrifer, K. A. *Dynamics in document design*. New York: Wiley Computer Publishing. 1997.

Sweller, J., and P. Chandler. "Why some material is difficult to learn." *Cognition and Instruction*. 12: 185-233, 1994.

Stuur, A. *Windows voor kinderen: deel2 [Windows for children: part 2]*. Utrecht: A.W. Bruna Informatica. 1996.

Van der Meij, H. Optimizing the joint handling of manual and screen. In: Carroll, J.M. (Ed.) *Minimalism beyond the Nurnberg Funnel*. Cambridge, Massachusetts: MIT Press. pp. 275-310, 1998.

Van der Meij, H., and J.M. Carroll. "Principles and Heuristics for designing minimalist instruction." *Technical Communication*. 42: 243-251, 1995.



Mark Gellevij works at the Department of Instructional Technology of Twente University, Enschede, The Netherlands. After graduating at this department, he worked as a technical writer in trade and industry. He is now working on his doctoral thesis about designing user support for complex software environments.

A Contemporary Renaissance: Changing the Way We Communicate



Hans Van der Meij works at the Department of Instructional Technology at Twente University, Enschede, The Netherlands. His main areas of interest are information seeking behaviour and technical documentation. In the latter area he has published sixteen chapters and thirty-three articles. One of his articles, co-authored by John M. Carroll received an Award for Distinguished Article of 1995” from The Society of Technical Communication. Hans also received the “Best Paper Award for 1997” from the IEEE Professional Communication Society. His studies in technical documentation combine theory, practice and empirical research. He currently conducts inventories on the roles and designs of screen captures, procedures, warnings and indexes.



Ton de Jong is professor of Instructional Technology at the University of Twente. He received his PhD from the Eindhoven University of Technology on the topic “problem solving and knowledge representation in physics for novice students”. He specialises in problem solving and discovery learning with computer simulation.



Jules M. Pieters is Professor of Instructional Technology (appointed 1991) Faculty of Educational Science and Technology at the University of Twente. After graduating in experimental Psychology at the University of Nijmegen, he wrote his doctoral thesis in 1980. Currently he is dean of the Faculty of Educational Science and Technology. Jules M. Pieters is co-director of the research program cognitive Tools and Instructional Design. He is involved in research projects on Knowledge acquisition and transfer and on designing learning environments.

APPENDIX A: Text Example Page

The Servive project
SimQuest V-September

Introduction

SIMQUEST makes it easy for you to create your own learning environment. You can design an environment in which media such as simulation, sound and video all interact with one another as in a video game. SIMQUEST is especially suited for creating a discovery learning environment. That is for situations in which students explore the domain rather than that is 'told' what the domain is about.

For a quick start and overview, this tutorial uses a learning environment about motion. The various chapters will guide you through the tasks of (re)creating important features of this environment, such as: the learner interface, and learner-tasks, called assignments. Meanwhile, you may also learn something about motion and how to simulate that.

Loading an existing environment

You first load the Motion application.

- 1 In the *SimQuest* menu, click **File** and choose **Load**
- 2 If necessary, choose the directory **apps**
- 3 Select **motion.sqa**
- 4 Click **OK**

Check if the application named **motion** is loaded.

What can you do with SIMQUEST

SIMQUEST enables you to design a multimedia discovery learning environment for your students. In this environment, you can build:

- models
- learner interfaces
- assignments
- explanations

The model as core of the simulation

A model is the mathematical formula that is used to calculate the values of your simulation. The Motion application is about the domain Uniformly accelerated motion. It uses the model:

$$v(t) = v(0) + F_drive*time$$

<i>where:</i>	<i>is meant:</i>
v(t)	speed at point in time t
v(0)	speed at starting time (point in time 0)
F_drive	acceleration
time	time

APPENDIX B: V-Part Example Page

The Servive project
SimQuest V-September

Introduction

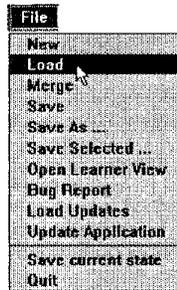
SIMQUEST makes it easy for you to create your own learning environment. You can design an environment in which media such as simulation, sound and video all interact with one another as in a video game. SIMQUEST is especially suited for creating a discovery learning environment. That is for situations in which students explore the domain rather than that is 'told' what the domain is about.

For a quick start and overview, this tutorial uses a learning environment about motion. The various chapters will guide you through the tasks of (re)creating important features of this environment, such as: the learner interface, and learner-tasks, called assignments. Meanwhile, you may also learn something about motion and how to simulate that.

Loading an existing environment

You first load the Motion application..

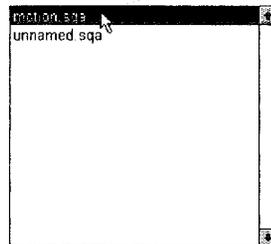
- 1 In the *SimQuest* menu, click **File** and choose **Load**



- 2 If necessary, choose the directory **apps**



- 3 Choose **motion.sqa**



- 4 Click **OK**



APPENDIX C: V-Full Example Page

The Servive project
SimQuest V-September

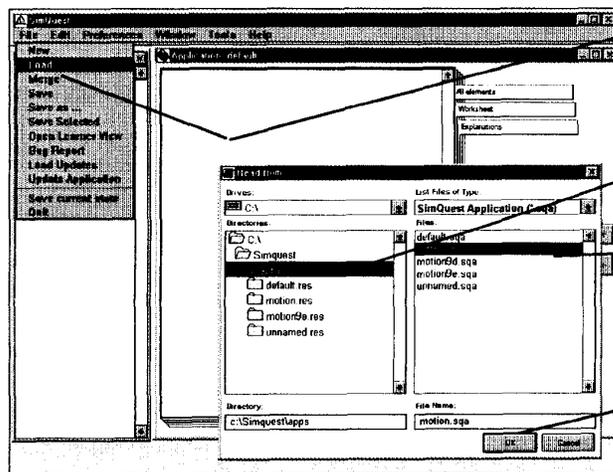
Introduction

SIMQUEST makes it easy for you to create your own learning environment. You can design an environment in which media such as simulation, sound and video all interact with one another as in a video game. SIMQUEST is especially suited for creating a discovery learning environment. That is for situations in which students explore the domain rather than that is 'told' what the domain is about.

For a quick start and overview, this tutorial uses a learning environment about motion. The various chapters will guide you through the tasks of (re)creating important features of this environment, such as: the learner interface, and learner-tasks, called assignments. Meanwhile, you may also learn something about motion and how to simulate that.

Loading an existing environment

You first load the Motion application.



1 In the *SimQuest* menu, click **File** and choose **Load**

2 If necessary, choose the directory apps

3 Choose **motion.sqa**

4 Click **OK**

Check if the application named **motion** is loaded.

What can you do with SIMQUEST

SIMQUEST enables you to design a multimedia discovery learning environment for your students. In this environment, you can build:

- models
- learner interfaces
- assignments
- explanations