Interactive Mathematical Books

A.M. Cohen, L.G.L.T. Meertens, S. Pemberton

1. Introduction

Recent developments in information technology—hypertext techniques, multimedia facilities, CD-ROMs, networking—as well as the increasing affordability of sufficiently powerful computers with high-resolution graphical displays, have greatly improved the possibilities of using computers for making knowledge accessible. Nevertheless, the prevalent way of learning about some advanced subject is still through the reading of books, whether or not aided by a human teacher who serves as an expert ready to explain those bits that you did not grasp from reading the book.

In classical Computer-Aided Instruction the assumption is that there is a given syllabus, a well-delineated body of knowledge, that students are supposed to master, whether motivated or not. Accordingly, there is generally a fixed route through the material, with strong emphasis on testing if the student has indeed mastered a morsel of knowledge, typically with a dialogue controlled by the automated instructor.

In contrast, our assumption is that the reader, in consulting an interactive book, *is* motivated to learn. Thus, the *book* metaphor is essentially more apt than the *instructor* metaphor. In a paper book, the reader can just browse, instead of reading it from cover to cover, consult it in any haphazard order, skip any sections, examples, exercises, etc., and come back to any point at any time. To call an interactive system an interactive *book*, not only should there be content prepared by an author for readers, but we also hold the tenet that *all interaction is initiated, and in general controlled, by the reader*.

Interactive books are not superior to more traditional books in all aspects. For the time being, they will be less easy to carry around. The resolution of affordable screens is an order of magnitude less than that of high-quality printing, which makes it tiring to read the fine print in mathematical formulae like 'n^{p_i}'. Also, the initial investment in acquiring the necessary equipment is sizable, and the initial production cost is appreciably higher (although, on the other hand, the unit reproduction cost of the book texts themselves is very low).

2. An example

A book's interactive potential has to be substantive before it offsets the disadvantages. As an example of the kinds of possibilities an interactive book can offer let us consider a cookbook (most of the sophisticated interactive 'features' mentioned below for this cookbook can be found in one or more commercially available interactive-cookbook programs).

2.1. Navigation support

Navigation comprises 'standard' hypertext facilities, which can be used both for unfolding/folding portions of a hierarchically structured text (as in what is called *outline processing*) and for excursions to related subjects, as well as browsing and search facilities.

Unfolding/folding can be used for quick navigation through a table of contents. In a cookbook, clicking on the first item of the following list

- MEATS
- FISH AND SHELLFISH
- VEGETABLES

will cause this line to 'open up', so that the screen displays

- MEATS
  * *Beef*
  * *Pork*
  * *Veal*
  * *Lamb*
  * *Poultry*
- FISH AND SHELLFISH
- VEGETABLES

Repeating this process on a well-organized table can quickly lead to any desired destination.

A beginning chef might not know the meaning of some technical term, say '*sauté*'. Clicking on such a term will display an explanation. An alternative possibility is to 'macro expand' the sentence in which the term occurs. For example, '*Sauté the slices of liver*' is expanded into '*Cook the slices of liver in a small amount of cooking fat or oil until brown and tender, using a skillet on medium fire and stirring well*'.

Note that the extent of navigation facilities need not be confined to the book itself; the book proper may be a coordinated collection of nodes in a larger web of active knowledge. This is of obvious relevance to mathematics (as well as many other fields of knowledge).

2.2. Annotation facilities

As in a paper book, readers may want to highlight portions, or to jot down remarks, add references, and so on. Unlike a paper book, these are reversible actions, so that the reader need not feel the anxiety of possibly spoiling the book. Possible annotations in a cookbook could be a suggestion which wine to serve with a dish, or marks for excellent or disappointing dishes. Cooks can also add their own favorite recipes to the cookbook, right where they belong in its organization.

2.3. Tailoring to user need

Given the interests and needs of the reader, the book can present different versions of its text. An interactive cookbook can give the measures in recipes in pints and ounces, or in liters and grams, according to user preference. It can also present the actual quantities needed for the expected number of people, where a paper cookbook leaves it to the chef to do a multiplication and division, the latter by a number you can never find in the book when you need it. Likewise, the number of calories (or joules) can be computed and presented. The recipes presented can be filtered on the availability of ingredients or the time available for cooking. The book can even turn into a specialized cookbook for certain dietary restrictions: a vegetarian cookbook by omitting meat-based recipes and doing some standard substitutions; or a cookbook trimmed down to low-calorie recipes for people who are minding their waist.

There are more exotic possibilities. Given a menu (a collection of recipes, one for each course) the book can present a merged and sorted shopping list. Also, it can merge the preparation steps of the recipes for the various dishes into one time axis and produce one combined list of instructions, so that a cook can work in parallel on several dishes without ever having to jump to and fro between different recipes.

2.4. *Expert assistance*

An interactive cookbook can also assist a cook more actively. While going through the actual preparation, it can highlight the next step to be done. When the reader clicks the DONE button, it highlights the following step. In addition, it can serve as a timer and tell our cook when to lower the heat, etc. If things do not go quite as smoothly as planned, it can rearrange the steps on the time axis in the culinarily most appropriate way. Finally, if something goes really wrong, it could offer expert advice on the best way to save the meal—with '*Call* 1-800-DIAL-A-PIZZA' as the last resort.

Expert assistance can also be helpful in composing a menu. In its simplest form, the user composes a tentative menu, whereupon the book reacts with an evaluation, using basic rules about the desirable variation in a menu as well as 'common-sense' and user-supplied constraints. In a more advanced setup expert assistance can be a truly interactive affair, with the book also suggesting modifications, and using the user reaction to those to adjust the constraints.

This does not exhaust the possibilities; neither in general (for example, we have not mentioned setting bookmarks), nor those specific to cookbooks (for example, planning ahead to use parts of a preparation for a later day, or giving advice what to do with leftovers), and, in fact, the possibilities are limited more by our imagination than by technical constraints.

3. INTERACTIVE MATHEMATICAL BOOKS

A mathematical textbook is not a cookbook. Yet, the reader will have few problems in transforming many of the examples given above of how interactivity can be helpful in a mathematical context. Although the examples were made concrete with subject-specific detail, they reflect more general principles of the kind of support that is helpful to human beings when approaching a complex task, in particular in an area that they have not yet mastered. Thus, it might seem that any mathematical aspects are only in the content of the book, and not in the medium, just as for traditional books.

Yet, there is something unique to the mathematical world: it is a formal world that does not of itself refer to the outside world. Any such references, as in mathematical physics, are externally imposed interpretations on symbols that by themselves have no meaning grounded in reality.

To learn to cook, the aspirant cook has to get hands-on experience and stir the pots and pans. No amount of reading, however interactive, can fully replace that experience. In mathematics, the 'real' world is the mathematical world 'in here': if some piece of mathematics can be done at all, it can—in principle—be done here and now, in the medium of the book. This opens unique possibilities for active reader involvement. For instance:

- *Navigation support.* Half of mathematics consists of giving definitions, and the other half of applying definitions. So the number of defined notions that are used is consistently high, and, more than in any other subject, the precise formulation of the definition counts. Thus, the ability to look up the definition of a mathematical term or symbol at the click of a button is especially important. Defined notions usually have arguments. For example, 'is continuous' takes a function as argument, as well as some arguments that are usually implicit, namely the topologies of the domain and codomain. It should be possible to view an unfolded application with its actual arguments in place—including the implicit ones, which may become important if the definition is unfolded further.

  Often not all mathematical knowledge used will be contained in the current interactive book. Thus, next to references to traditional texts, there is a clear value in allowing links to other interactive documents, mathematical encyclopedias in electronic form, etc., which will increase as more mathematical texts become available on-line, for instance via the World-Wide Web (T.J. Berners-Lee, R. Cailliau and J.-F. Groff [2]—see for instance figure 1.

  When studying a large theory it is very easy to get lost. In such cases a reader can be greatly helped when provided with an easily navigable presentation of the mathematical structure of that particular theory, which is basically a graph containing the definitions, lemmas, and theorems and their interrelations. This graph can be automatically extracted from the text if all hyperlinks are in place, and properly annotated as to their nature. Obviously, such a facility is not only useful for not getting lost; it may also be very helpful for authors, or for researchers who want to generalize a theory to a related area of mathematics.

- *Annotation facilities.* For mathematics it is important that reader annotations not be limited by the size of the margin. Likewise, they need not be confined by the limitations of a purely textual rendering. Whenever the reader can ask the book to perform certain computations on mathematical objects contained in the original text of the book (say, compute the dimension of a given Lie algebra), this should be equally possible on mathematical objects occurring in annotations added by the reader. Thus, annotations are not second-class citizens, but may have recourse to the full spectrum of interactive capabilities.

- *Tailoring to user need.* An interactive book can allow readers to change the mathematical notations used to preferred notations, for

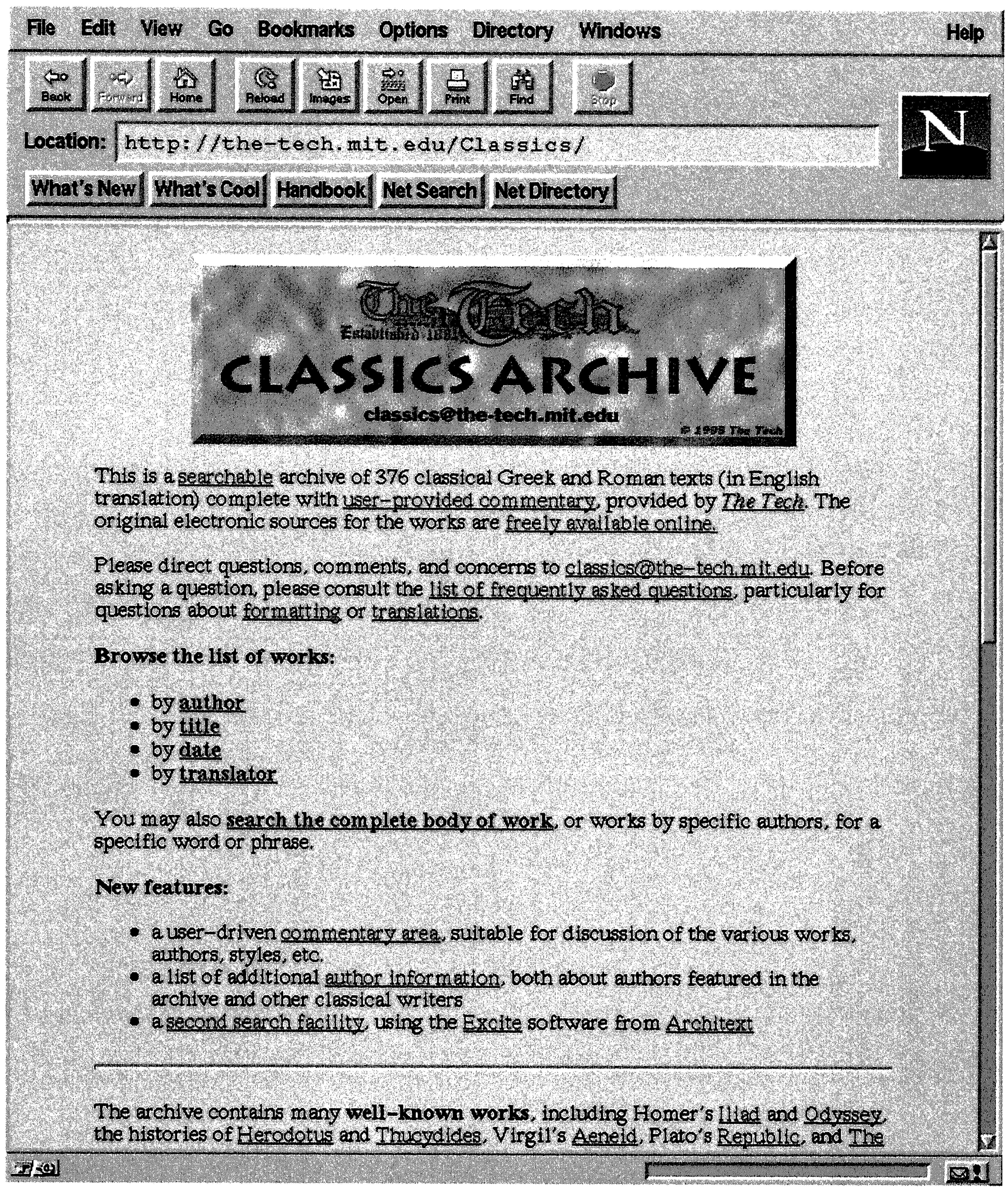

Figure 1. A WWW page that could serve as an external hyperlink from an encyclopedia entry on classical scientific literature.

example because of familiarity. The book can then also accept these notations for user-supplied input.

For advanced readers the book could hide from view elementary sections or expositions, give only informal and intuitive versions of the proofs, and skip elementary or add more advanced examples. It could also be possible for a reader to ask for a view on the book tuned to a specialized subject treated in it (for example, free Lie algebras), in which case it only presents those parts that are relevant to this subject.

More generally, there may be several linear routes through an essentially non-linear document, and the best choice may be determined by the interest and expertise of the reader. For example, a reader conversant with the topic of a book, consulting it as a reference on a specific issue, has other needs than the reader who is gradually learning about an entirely new subject, and this difference in needs can be reflected in a different presentation of the material. It is important here that the choices are under the control of the reader, and that locally other choices can be made than globally, for example for the professional reader who wants to take a quick 'refresher course' in some specific part of a generally familiar area. The concept of routes and its implications for interactive books is worked out in more detail in [4].

- *Expert assistance.* For interactive mathematical books, the recent and rapid development of high-quality computer-algebra packages is of paramount importance. Provided the infrastructure needed to connect the two is present, a computer-algebra package can bring life to the examples and exercises given in the book. Instead of one static example with predefined data, we can have *active examples*: the reader can change the data and immediately see the effect on the results.

  In exercises, a book can provide hints and access to the computing machinery needed; tedious calculations can be left to a computer-algebra engine, and the reader can concentrate on understanding the essence of the exercise. Furthermore, the book can (for some exercises) check the reader-supplied answer by simply computing it and, if the given answer is wrong, present a related exercise, which is recomputed for the given context.

  Readers can use the book as a platform to perform their own mathematics within the context of the book. They will be able to insert solutions to exercises, and to invoke algorithms from computer-algebra systems and other external applications transparently, meaning that

> the book takes care of the translation between the application's input and output format and the book's internal format.

It is this promising integration of mathematical textbooks with recent developments in computer science, and especially computer algebra, that instigated us to start the ACELA project (A.M. Cohen and L.G.L.T. Meertens [1]).

By taking Lie algebras as the subject matter for our first prototype, we hope to establish convincing evidence that an interactive book offers the reader a more active involvement than is possible with a paper book, a better guided tour than is provided by a computer-algebra package, and a more rewarding experience than obtained by using these two together but separately, as independent entities.

4. ARCHITECTURE

In designing and implementing the prototype system, we can distinguish between a subject-independent part—the kernel system—and a subject-dependent part—in our case, mathematics, and more specifically, Lie algebras.

The kernel system needs content, like a record player needs a record; in this case, the content is the text of the book. For a proper separation of concerns, we wish to consider the 'truly' subject-dependent part as belonging to the content. So the content of a book can consist, next to a text, also of algorithmic entities: author-supplied *methods* to work on the types of objects occurring in the text. When the book 'record' is loaded into the system 'player', these methods are loaded as well. Part of the project is indeed devoted to creating powerful implementations of various algorithms for Lie algebras.

Still, we must take the future mathematical content into account right from the start, also for the design of the kernel system. The non-trivial demands posed by the requirement that mathematics can be brought to life would be virtually impossible to meet by afterthought modifications.

A simple example of this is the user control over the mathematical notation. This requires a strong logical separation of content structure and visual presentation, something that is missing in almost every mathematical-document preparation system. For example, although the *presentation* structure of $f(x)$ is juxtapose(identifier('f'), parenthesis(identifier('x'))), its *content* structure is apply(function=identifier('f'), argument=identifier('x')). If content structure and presentation structure are not kept separate in the software *ab initio*, it is practically impossible to pry them apart later. Such a separation is only possible if it is catered for in the architecture itself.

More generally, the desire to achieve transparent interoperability with external engines requires a flexible juggling with various representations for the same (mathematical) object. The actual representation rules are, of course, part of the (algorithmic) content of the book, but it would be awkward, to say the least, if the infrastructure for handling pluriform representations and interposing the right transformations is not provided by the kernel system.

Amongst aims of the planned system are:

- *Interoperability with external engines.* For the direct purpose of having a prototype system, we could resort to an architecturally less pleasing design and create an *ad hoc* interface to a Lie-algebraic engine. However, our plan is to approach the interoperability issues in a more general way: a further goal of the ACELA project is to establish a test ground for the transparent integration of computer algebra, document processors, and proof checkers, using a variety of external engines.

- *User interaction.* The infrastructure provided by the book for the interaction with external engines will also be used for the interaction with the user. Thus, the user can use familiar presentation-oriented notation while entering formulae, which are translated by the book into content-oriented structures.

  We want the reader of the book to get into it with a minimum learning effort, and in general that all access to external engines is mediated transparently by the book.

  'Transparently' means here that the user interface keeps the same 'look and feel', whatever the engine used. To achieve this we use the interaction paradigm developed in the Views project as described in [3], specifically with the aim of creating a unified interaction model for the interoperability of varying applications (see figure 2). The conceptual model underlying the interaction paradigm of Views is that all interaction is achieved by the user by editing documents. These documents may be structured, and contain mathematical objects. The notion of editing comprises ways to enter and change text, as well as facilities for moving around and searching—in short, navigation.

  An essential part of the Views interaction paradigm is that the semantics of the system as it presents itself to the user is based on defined logical relationships between various documents. If one is modified by the user, the system recomputes and updates the other. A basic example of this is a spreadsheet. A more involved example can be the relationship between a query on a database and the result of that query. By editing the query, the result is automatically updated. The same holds when the database is modified. Exercises and active examples can easily be created with this mechanism.

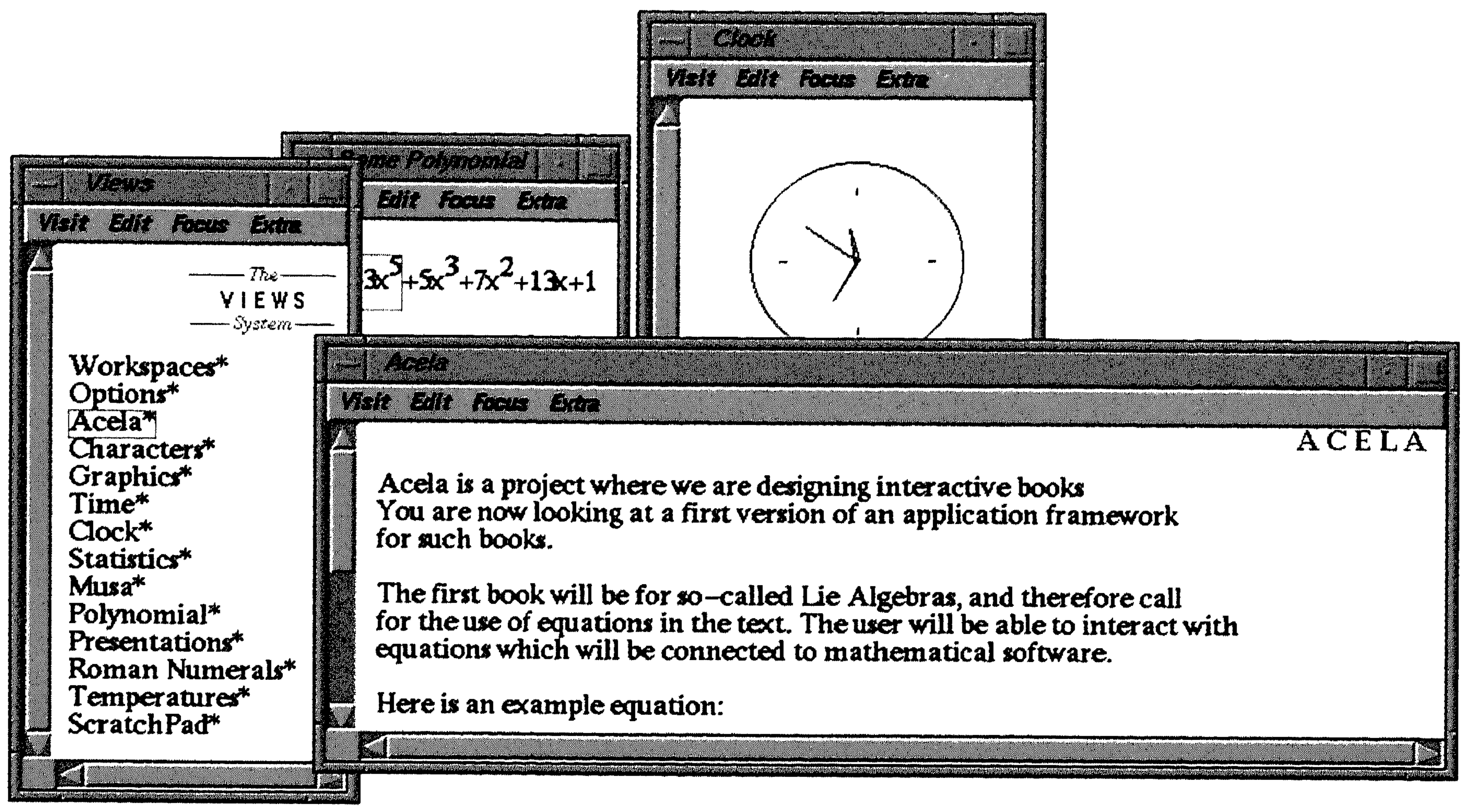

Figure 2. Sample Views session.

Similarly, the visual presentation of formulae is determined by one or more 'notation' sheets, which are style sheets for the type or sub-types of 'formula'. By editing these sheets, the user can exercise control over the notation.

Since all users are familiar with the basic edit paradigm and edit mechanisms, getting to work with the intended interaction paradigm needs hardly any learning.

- *Authoring.* The prototype system is not only intended to be for readers; the same system will be used by authors. Thus, there will be no distinction between a 'reading' system and an 'authoring' system. Obviously, authors have to be able to read what they have just written, and interact with it just like prospective readers could do, and readers, while creating annotations, become authors. The only hard difference is that, presumably, arbitrary readers cannot *change* the content of the book they are reading, but only change its *appearance* and further add annotations, while an author must be able to change the contents. However, we envisage the addition of a number of tools that, although accessible to readers, are primarily intended as facilities for authors creating a complete book text. For example, a constraint on a route through the text is that the sections may only depend on previous sections, and authors will be served with tools for managing routes that keep track of such constraints. As for reading, the interaction paradigm for authoring will be that of editing text. The additional tools will facilitate structuring and restructuring, creating hyperlinks as well as the dependencies that are used for setting up routes, and interactively plugging in interfaces to external engines, together with monitoring and debugging aids.

5. CONCLUSION

The design we have set forth in this paper is ambitious, and having some priorities among them is not a luxury. One priority is that we strive for getting the things that we do right. As we see it, the architectural decisions made now will have a strong impact on the long-term potential of the system, and ill-chosen decisions will eventually emerge as unsurmountable limitations, but given the right architectural set-up, further enhancements in functionality will always be possible.

None of the individual aims we have is by itself unique, in the sense that some product exists that achieves it. At the same time, each existing product—in its current incarnation—has limitations that make it unsuitable as a platform for realizing the interactive book we have in mind. The main contribution of our project in this context would be in showing how the various desiderata can be combined gracefully.

Any form of interoperability can ultimately only be successful through the use of a common standard, whether used at the scale of a single project (requiring some form of adaptation for externally supplied software), or adopted at a wider range. System-level standards for document-centred computing are now emerging and will be watched closely. An essential requirement for us, though, is that such standards be both platform-independent and vendor-independent.

REFERENCES

1. A.M. COHEN, L.G.L.T. MEERTENS (1995). *The ACELA Project: Aims and Plans*, CWI, Amsterdam. Available on-line: `http://www.cwi.nl/cwi/projects/acela/papers/Aims+Plans.html` .
2. T.J. BERNERS-LEE, R. CAILLIAU, J.-F. GROFF (1992). The World-Wide Web. *Computer Networks and ISDN Systems 25*, 454-459.
3. L.G.L.T. MEERTENS, S. PEMBERTON (1992). *The Ergonomics of Computer Interface—Designing a System for Human Use*, Report CS-R9258, CWI, Amsterdam. Available on-line: `http://www.cwi.nl/ftp/CWIreports/AA/CS-R9258.ps.Z` .
4. O. WEBER (1994). *Routes through the maze*, ACELA working document, CWI, Amsterdam. Available on-line: `http://www.cwi.nl/cwi/projects/acela/papers/maze.ps.Z` .