Replacing paper notebooks with their electronic counterparts seems to be a very appealing perspective. No longer can we lose the important notes that were made on a napkin during the last night’s dinner. No longer can we forget where did we scribble the description of a new, revolutionary design. With the electronic notebooks all these problems go away, that is at least, what we expect. All our notes are now stored in one place and we can search their contents with the tap of a button. We can organize notes easily and fax or e-mail them to our colleagues. Finally, if from time to time we want to go back to paper, we can still print our notes.

The idea of using computer as a notebook is not new. Until recent years, however, there were no suitable devices available. Personal computers were not portable, and even popular nowadays notebook computers could not become electronic notebooks, since they are too big to be carried and used in every situation. Moreover, in most cases, they do not have pen input allowing for a natural user interface. With advent of PDAs, and similar devices, the concept of an electronic notebook is coming closer to reality.

At the University of Toronto we are exploring the notion of electronic notebooks, and in particular, electronic engineering notebooks (EEN) as the replacement for traditional, paper-based engineering notebooks. In this article I describe the idea of electronic notebooks, their basic functionality and the prototype of an electronic engineering notebook implemented on the Newton.

WHAT ARE ELECTRONIC NOTEBOOKS?
Timely access to the right information is one of the typical problems of our era. Providing a solution means making information available to those who need it, when they need it, and where they need it. For this to happen, more information must be captured digitally onto systems and better methods for organizing and structuring this information must be provided to make it accessible to others. To achieve this, we need to (1) provide easy-to-use tools that will encourage users to capture more information, (2) provide users with the capability to organize and structure this information in ways that make it accessible, and (3) provide them with the capability to browse and retrieve this information efficiently. Electronic notebooks are the tools that attempt to meet the above goals in many application domains.

But let us take a step back and first consider electronic notebooks as a mere replacement of paper notebooks. Of course, limiting utilization of paper is by itself very beneficial to our environment. Production of paper consumes a lot of energy and natural resources; significant portion of landfills is currently taken up by paper. While this is a good cause, it is rarely the motivation for development of electronic notebooks. We are too familiar with paper and we like using it. In the course of our lives we develop many skills related to manipulating paper. Furthermore, reading from paper is less straining for our eyes than reading from computer displays (at least with the currently available technology).

When we move to computerized notebooks, we are trading in our familiarity with paper for additional functionality. To make this move smooth and to make us accept the technology electronic notebooks should be designed around the well known notebook metaphor. Therefore, the design of user interface for electronic notebooks should incorporate many elements from a paper notebook.

If we look at the process of using notebook, first we write, scribble, sketch, delete, and cut&paste, that is we create the content. Then we usually organize and structure the content. In a simple case, it may be just data and time stamping or filing notes under separate folders. If we want to refer to the content later, we browse through it and search for required topics. If we want to share our scribbled down ideas with a colleague, we either have to send or fax him/her a copy of our note, or we can meet him/her in person to show our notes. The following summarizes the basic notebook user activities:

- Acquisition of information – creating content;
- Authoring – organizing and structuring the content (filing, marking, linking);
• Browsing, navigating, and searching; and
• Transferring and sharing information.

CAN WE USE NEWTON'S NOTEPAD?
We chose Newton for our prototype electronic notebook implementation platform as the most sophisticated pen-based PDA currently available. Newton already has quite a few of the required features. In addition to that, its operating system, built-in view classes and prototype templates make it relatively easy to extend.

I asked myself first: Can I use Newton’s built-in NotePad application for our purpose? Unfortunately, the NotePad application has its limitations. Most importantly, very limited means exist for organizing and structuring of captured information. Notes can be organized only in a simple one-level folders. Each note has a date- and time-stamp, and an optional title, which make it easier to locate the note in the overview display. The content of a note, however, cannot be further structured. If you want to use Find, you need to know what to search for, and that is not always the case. Existing organization methods impose the navigation capabilities. In the NotePad, navigation is limited to linear, that is determined by the sequence of notes.

Provided NotePad functionality is not sufficient for our purpose, it does not give efficient access to stored notes. Stationery, introduced in the new Newton operation system 2.0, does not solve the problem. Notes based on any stationery type are still separated, that is they are not hyperlinked. Stationary does not allow for dynamic note definitions in the sense of user changeable interface elements (for example, adding new buttons to notes). These are the two important features required for the EEN implementation. The missing notebook functionality has to be added.

NEWTEEN – MY PROTOTYPE EEN IMPLEMENTATION
I developed NewtEEN as a stand-alone application. The main functions of the current version are information acquisition, authoring, browsing and navigation, and information transfer. The acquisition is no different than in the NotePad. On the other hand, capabilities for organizing and structuring are considerably extended. They provide the basis for the extended browsing, navigation and search functionality. Information transfer is based on form transmission.

Before I go on to describe the details of NewtEEN’s functionality, I shall briefly mention other parts of the system.

A Look at the Whole System
NewtEEN users are not working in the void, they collaborate and they are members of the engineering teams. NewtEEN is thus also a part of a larger information system. Central part of the whole system is a knowledge base for storing design knowledge and providing services for the management of shared design. The knowledge base and the server are currently implemented in an object-oriented manner using Prolog. The software runs on a Unix (SunOS) workstation. The knowledge base objects are stored as frames; a frame is a collection of three types of slots: attributes, relations and messages. Hierarchy of frames can contain, for example, a product design related knowledge. The information system provides tools for accessing and updating the design knowledge, and services that support collaborative and concurrent design. The knowledge base includes six layers; from inner to outer layers, they are: enterprise object model, constraint management, version management, query management, design management, and access management. The NewtEEN communicates with the knowledge base and uses its services. Further description of the knowledge base and its layers goes beyond the scope of this article.

Because no TCP/IP stack is available for the Newton (as of this writing), I developed a proxy gateway that mediates the communication between the Newton and the knowledge base server. The Newton accesses the proxy using an RF modem (see Figure 1). The second RF modem is connected to the UNIX workstation running the proxy that communicates with the knowledge base server over the Internet. The NewtEEN sends and receives ASCII strings to and from the proxy, which then translates them into the server requests and from the server responses respectively. Content of the NewtEEN is transmitted to the knowledge base by means of forms, which I describe later. The system also provides a WWW user interface. Information generated by NewtEEN can be thus accessed through the World Wide Web.

![NewtEEN communication system](image)

“**Figure 1 - NewtEEN communication system.**”

Information Acquisition
NewtEEN uses the notebook metaphor. Notebook consists of pages, and page contains objects (see Figure 2). NewtEEN page accepts the same types of input and the same operations on objects as the built-in NotePad. Apart from the user input, objects can be created by drag-and-drop between notebook pages and the Newton built-in programs, and by linking-and-embedding of data from specific applications (currently, I support one program – QuickFigure Pro spreadsheet). Objects are stored in the Newton soups. Each object on a notebook page has several attributes, some are automatically generated (date, time, author, page name) while other are manually entered by the user, the latter include tags.

Organization and Structuring

Once the content has been created in the process of recording information, it has to be organized and structured to facilitate the efficient access. NewtEEN provides four methods of organization:

- linear – sequence of pages (Figure 2),
- hierarchical – table of contents (Figure 3),
- non-linear – hyperlinks (Figure 2), and
- tag indexes (Figure 4).

Linear and hierarchical organizations are well known; they do not require further explanation, therefore I will focus on the latter two.
I call the process of organizing and structuring – authoring. I believe that acquiring information and authoring cannot be fully separated in time. If they are, loss of information may occur. I use the term semi real-time authoring for my approach. The unintrusiveness of authoring is depended in the first place on the user interface design. I employ tags and hyperlinks as the main user interface authoring elements.

Information can be tagged while it is being entered into the notebook. Tagging involves selecting an appropriate tag from the tag menu (see Figure 4). A text button with the corresponding text is then added to the current page (several tags can be seen in Figure 2). The tag button can be dragged around the page until it is placed close to the object it refers to. Entering tags automatically adds corresponding entries to the tag indexes in the table of contents.

Tags are not a novel idea. The selection of terms used for tags, however, is not a trivial task. The set of terms employed depends on the application domain. In my case it is engineering design. In order to support the selection of terms, I used the results of enterprise modeling research done here at the University of Toronto. One of the objectives of enterprise modeling is building ontologies, in other words, determining the common vocabularies for enterprise models. Ontologies are the basis of knowledge representation used for constructing information systems. The detailed description of ontologies is beyond the scope of this article. Suffice to say, that I adopted a subset of ontology terms for tags.

Tags are of two types, ontology tags, based on the mentioned above enterprise ontologies (for example, part, parameter, feature, requirement, constraint, function), and organizational tags (for example, objective, meeting, to-do, milestone, action item, proposal). Each tag has a form associated with it which can be opened for viewing and editing by tapping the tag button. I describe the usage of forms in the current implementation in more detail later (see Section: Information Transfer).

The next organizational element which I employ in the NewtEEN are annotated hyperlinks. Links are unidirectional and can be entered between any two notebook pages. They create an annotated, directed graph with the notebooks pages as nodes. Links are implemented in a similar way to tags. They appear as text buttons with text being now the title of the destination page. The associated annotation can be opened by tapping the link button. An example of the link button connected to the page entitled “frontBrake” is shown in Figure 2.
Tags and hyperlinks are the means for the more sophisticated organizing and structuring of the design information recorded by engineers during the design process. Together all the organization and structuring capabilities provide the necessary basis for browsing and navigation.

Information Transfer
NewtEEN has a range of communication capabilities. It can receive e-mail notifications, for example, related to the design changes, and it can communicate with the knowledge base. Communication with the knowledge base is possible in two modes. First, the knowledge base can be queried and its content changed by executing Prolog predicates. Generic (see Figure 5), as well as application specific predicates (see Figure 6) can be entered. This is a very powerful feature, but I have not yet designed the specialized user interfaces for the presentation of results, thus its usefulness remains limited. Second, and more interesting, is transmitting of forms.
As described in the previous section, each tag has a form associated with it. The structure of the form (see Figure 7) corresponds directly to the attribute and relation slots contained in the appropriate frame in the knowledge base. Tags are entered during the concept time while engineers are writing and sketching in their notebooks. At the review time, after the information has been recorded in the NewtEEN, users fill out the forms and then submit them to the knowledge base. In this way post-hoc tagging, which may lead to the loss of information, is avoided. At the same time, by transmitting to the knowledge network only post-processed information, we avoid the “garbage-in/garbage-out” syndrome. Tags and forms play an essential role in making the NewtEEN content understandable, in capturing the design information, and in sharing the recorded information.
CONCLUSIONS AND FUTURE DIRECTIONS
I presented NewtEEN – the prototype implementation of an electronic engineering notebook based on the Newton. In this article I described the NewtEEN from the functionality point of view, the most interesting implementation-level details will be described in the next article.

NewtEEN can be obviously further improved and better integrated with the Newton’s built-in programs. The original application was written under Newton’s operating system 1.3, porting it to Newton 2.0 is on my current agenda.

The main disadvantage of the NewtEEN is due to the Newton’s hardware – the screen is too small. I hope that in the near future tablet-like devices based on the Newton operating system will appear on the market. Success of our project, like of many others, is ultimately dependent on the user’s acceptance. Again, display size is the most crucial factor here.

If you are aware of a large screen Newton please let me know!

Finally, I believe that some of the presented ideas could be used to further improve the Newton’s NotePad. This is very important as the NotePad is normally the always present background application.

ACKNOWLEDGMENT
I would like to thank Celso Mialo of The Newton Store in Toronto for providing the radio modems for testing.