RC 21078 (94190) 16 Jan 98
Computer Science/Mathematics

IBM Research Report

The Internet Knowledge Manager, and its Use for Rights and Billing in Digital Libraries

Adrian Walker

IBM Research Division
T.J. Watson Research Center
Yorktown Heights, New York

IBM Research Division
Almaden · Austin · China · T.J. Watson · Tokyo · Zurich

The Internet Knowledge Manager, and its Use for Rights and Billing in Digital Libraries

Adrian Walker

IBM T.J.Watson Research Center
Yorktown Heights
NY 10598

adwalker@us.ibm.com

Abstract: It is well known that a change in the way human knowledge is represented in software can trigger a major change in business practices. For example, the invention of Hypertext Markup Language (html) has triggered the development of the Web, and of successful Web-based businesses.

It is difficult for a professional in charge of a digital library collection to specify all the future ways in which items from the collection will be assembled and billed for. Even if such a specification is possible, it is not always easy to communicate the spec to programmers, and to check that the program they write in fact implements the spec. Add to this that the spec changes as new items are added to the collection, and as the collection is put to new uses on demand from consumers, and we have a major software engineering problem.

The Internet Knowledge Manager is designed to solve this problem, by opening up a new way of using the Internet. The IKM allows a professional, who has a rights and billing policy for a digital library in mind, to specify an agent that implements the policy, using his or her own English terminology and an ordinary Web browser. Thus, the IKM adds a new way in which human knowledge can be represented in software, and it could trigger new ways of doing business on the Web.

1 Introduction

It is well known that a change in the way human knowledge is represented in software can trigger a major change in business practices. For example, the invention of Hypertext Markup Language (html) has triggered the development of the Web, and of successful Web-based businesses, such as the bookseller amazon.com. Up to now however, the Internet and the Web have mainly partitioned people into *authors* (programmers and Web page designers) and *users*. Some of the successes in Web-based business are starting to blur the distinction between authors and users. For example, amazon.com has pages to which ordinary users contribute opinions about books.

It is difficult for a professional in charge of a digital library collection to specify all the future ways in which items from the collection will be assembled and billed for. Even if such a specification is possible, it is not always easy to communicate the spec to programmers, and to check that the program they write in fact implements the spec. Add to this that the spec changes as new kinds of items are added to the collection, and as the collection is put to new uses on demand from consumers, and we have a major software engineering problem.

The Internet Knowledge Manager is designed to solve this problem by opening up a new way of using the Internet. The IKM allows a professional, who has a rights and billing policy for a digital library in mind, to specify an agent that implements the policy, using his or her own terminology and an ordinary Web browser. In this approach, English words take their meaning from their context, rather than from a separately maintained dictionary and grammar. This makes it easy to write down knowledge with specialized words and phrases, such as "the ISBN in OPAC", and then to use the knowledge directly. The resulting software agent can be run, modified, and tested directly from the Web browser, and can provide step by step explanations of its decisions. The agent can then be integrated into a buy server and Cryptolope [1] architecture. Thus, the IKM adds a new way in which human knowledge can be represented in software, and it could trigger new ways of doing business on the Web.

The next section describes some of the requirements for rights and billing in digital libraries. Section 3 describes the design of the IKM, its nonstandard approach to computing with natural languages, and how it could be used in a buy server and Cryptolope architecture. Section 4 shows an author-user example session, and Section 5 is a summary.

2 Requirements

The requirements for rights and billing software for Digital Libraries include

- billing for multilevel compound documents with component documents belonging to several suppliers,
- dealing with complex international regulations about intellectual property, and
- dealing with specifications of rights and billing policies that are written in different natural languages, and
- dealing at 'fine grain' with large numbers of low cost Digital Library transactions, and
- ease of understanding and accountability for the decisions that a policy agent makes.

Beyond this, the requirements are open ended. As mentioned, this amounts to a major software engineering problem. The next Section describes how the design of the IKM solves the problem.

3 Design

The Internet Knowledge Manager opens up a new way of using the Internet. The IKM is a server program containing new technology that

- allows an end-author client to use a standard Web browser, such as Netscape or the Internet Explorer, to write a rights and billing policy agent using his or her own terminology and grammar,
- avoids the linguistic maintenance difficulties that have defeated several major attempts to commercialize prototype natural language front ends
- supports end-authoring and end-use in French, German, and most natural languages that use an alphabet similar to English
- provides accountability, by means of step by step explanations of the way in which an agent makes decisions.
- includes safeguards to prevent unauthorized users from changing billing policies, and
- contains detailed, automatic, hypertexted, context sensitive help and checking screens for authors.

In the current state of the art in natural language processing, different techniques are used for different tasks. For example, some speech recognition systems use statistics plus a dictionary, but no grammar of English. Machine translation programs tend to use a grammar plus a dictionary. The IKM uses yet another set of techniques:

- *a theory of declarative knowledge*
- *some theory based technology for executing rules*
- *Direct English Specification of Agents*

The first two techniques are described in [2, 3] respectively. The third technique, *Direct English Specification of Agents*, has the following advantages: it is is simple and robust, it needs no maintenance of a dictionary or grammar, and it allows the author to use his/her own words and phrases, including jargon. The technique allows agents to be written in English, so they are self documenting. In particular, the IKM can explain a decision made by an Agent, step by step, in English. Further, the IKM can easily be adapted to Spanish, French, German, and so on. Section 4 shows how the *Direct English Specification of Agents* technique is used.

The IKM could be used in a buy server and Cryptolope architecture as shown in Figure 1. In the Figure, a professional in charge of a digital library collection uses the **Content Provider's Web browser** client to write **policy agents**. The actual writing is assisted by the **Internet Knowledge Manager** server. Each Content Provider has an ID, an author password, and a user password for the IKM. Logging in to the IKM with author password allows agents to be written, maintained, and run. Logging in with the user password only allows agents to be run.

The architecture will allow a customer, using the **Customer's Web browser** on the right of the Figure, to combine the items in the Provider's collection into an individually self-customized compound document. The customer will also be able to see computed rights and billing decisions for the customized document, plus explanations for the decisions. When the customer is satisfied with the customization and its rights and billing decisions, he or she can request the actual document from the **Multimedia Document Server**.

At this point, the IKM will use the knowledge in the **Content Provider's Policy Agent** to pack the customized document into a **Cryptolope** that the **Multimedia Document Server** can transmit to the **Customer's Web Browser**. The IKM will also prepare a **Decrypt License** to allow a special **Helper Application** to show the customized document. In addition, the IKM will update the **Inter-Provider Payments** and **Customer Profile and Billing** databases. The IKM will also allow the **Content Provider** to specify agents that use the accumulated **Customer Profile and Billing** information to make marketing surveys, and to advertise items of interest to individual customers.

The next Section shows an Author-User Example Session.

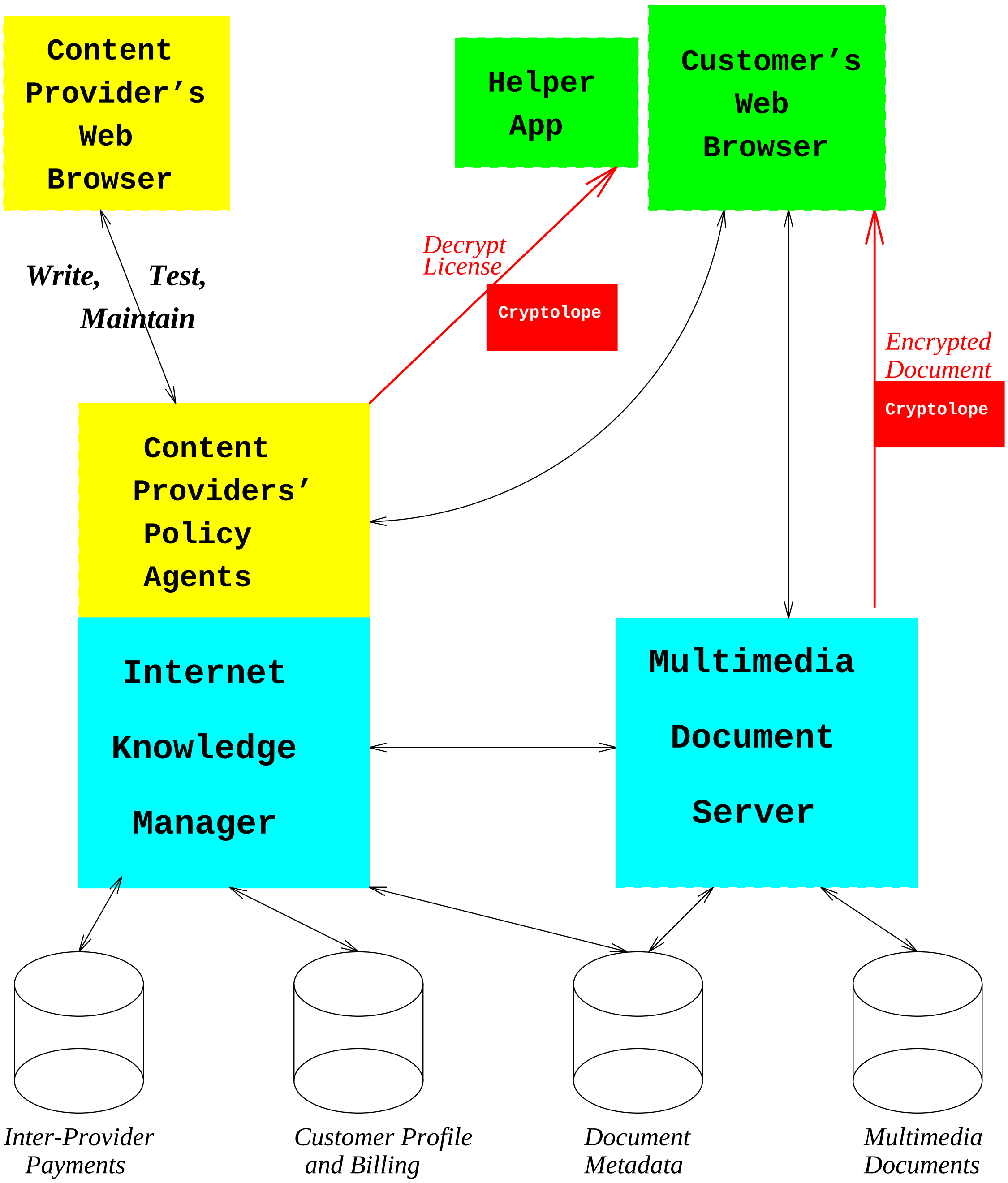
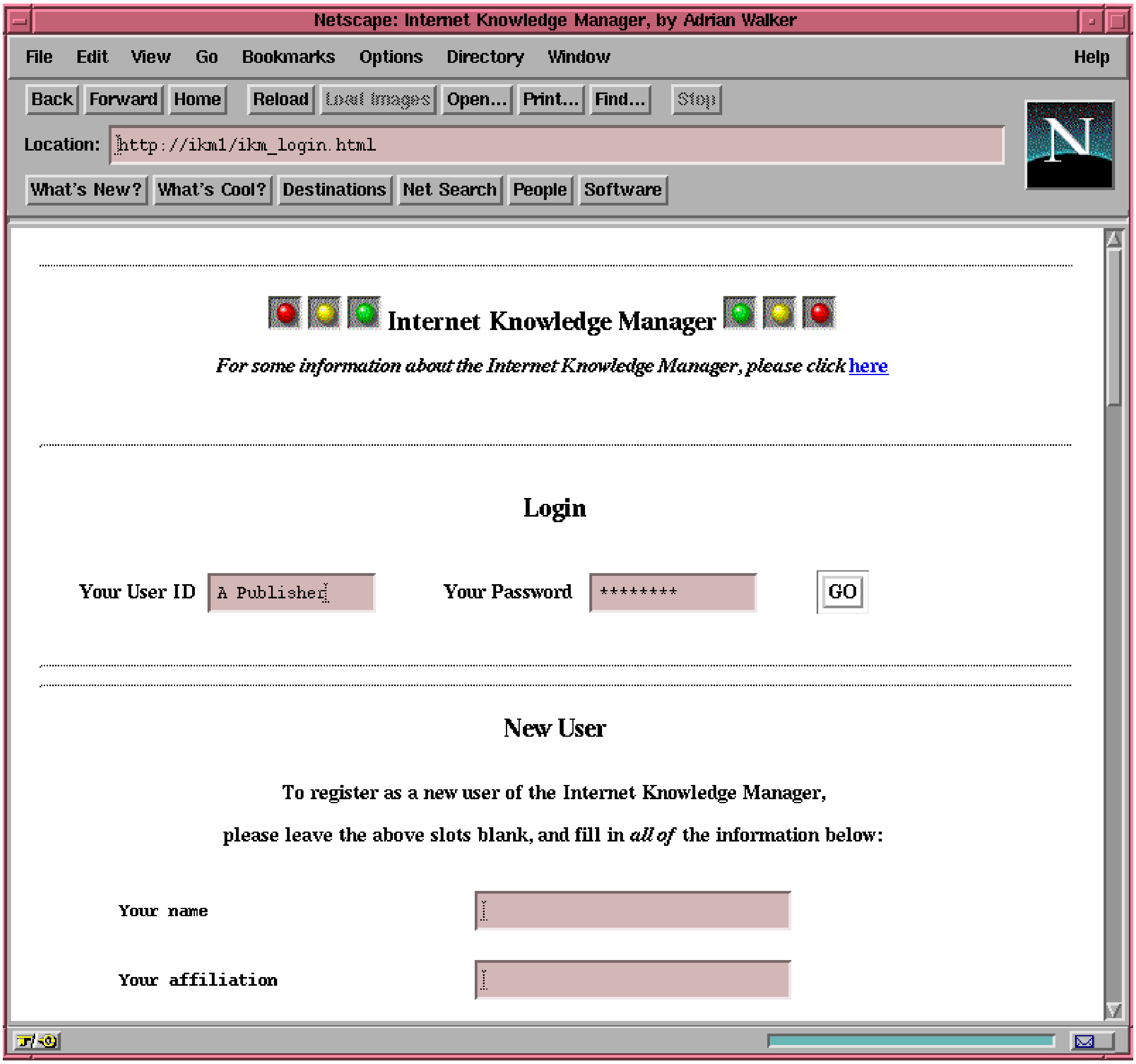


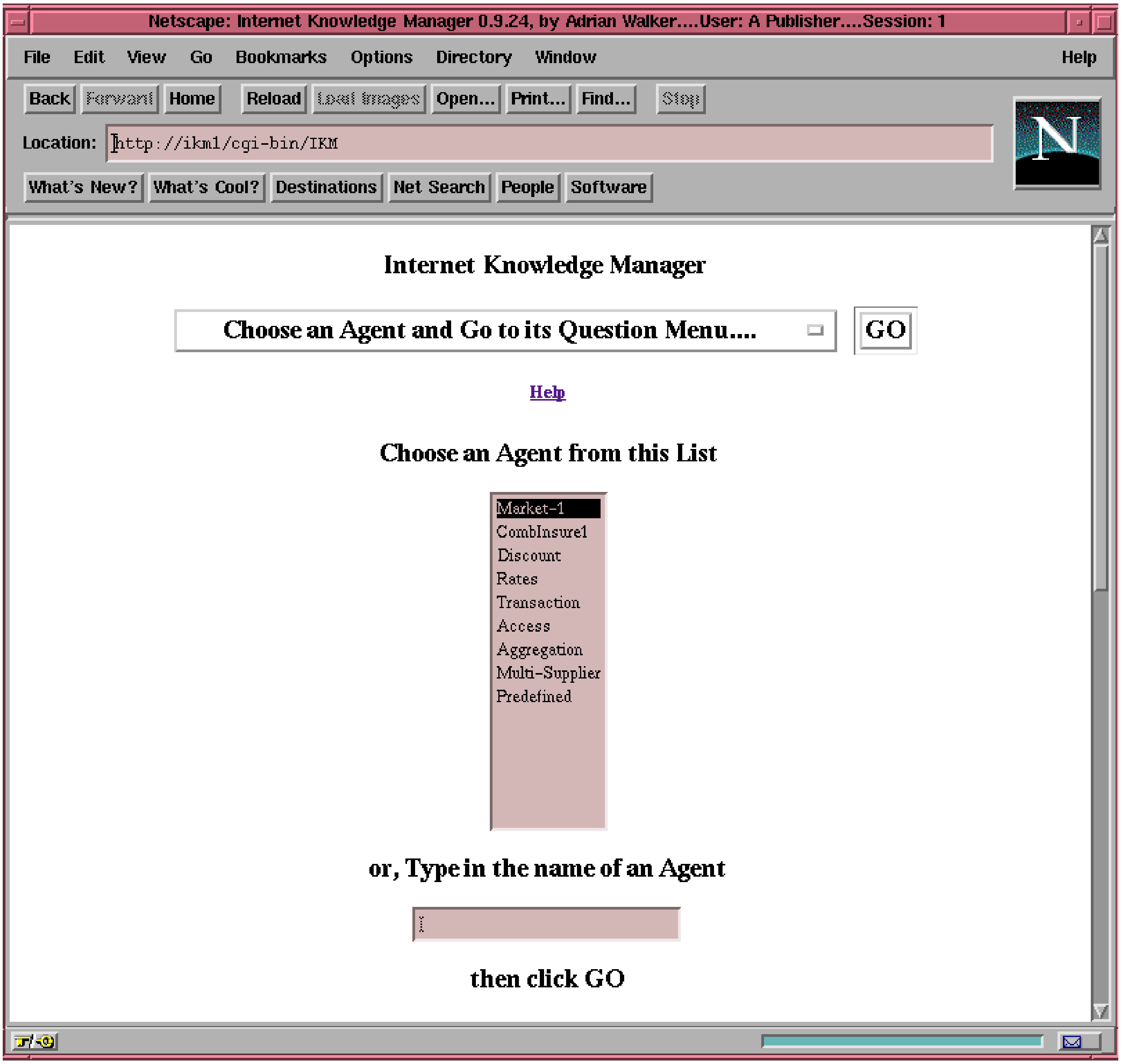
Figure 1. The IKM in a Buy Server and Cryptolope Architecture

4 An Author-User Example Session

A person who writes, maintains, or uses an agent logs into the IKM using a page like this.

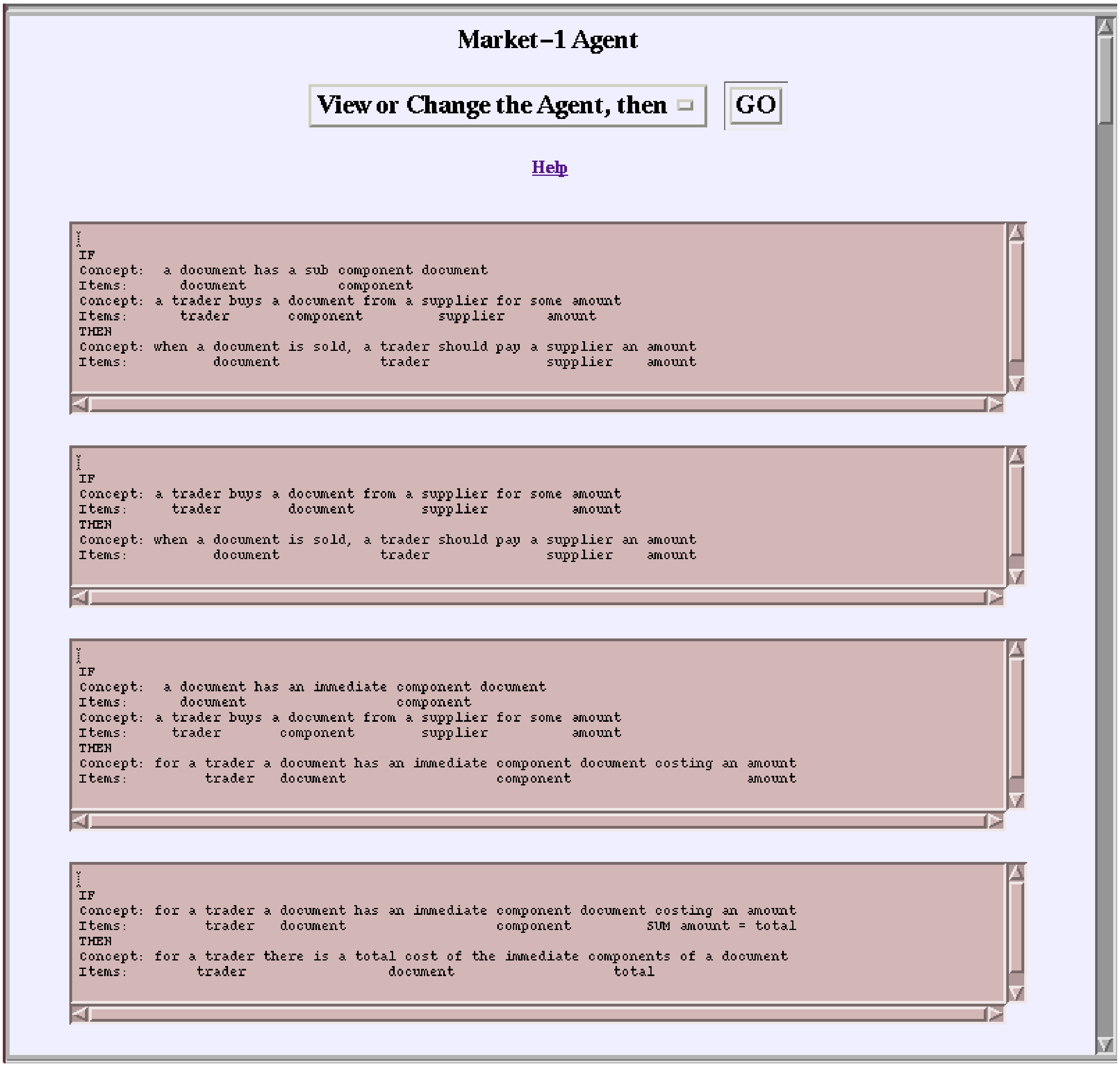


The result is a start page like this, showing the available agents.



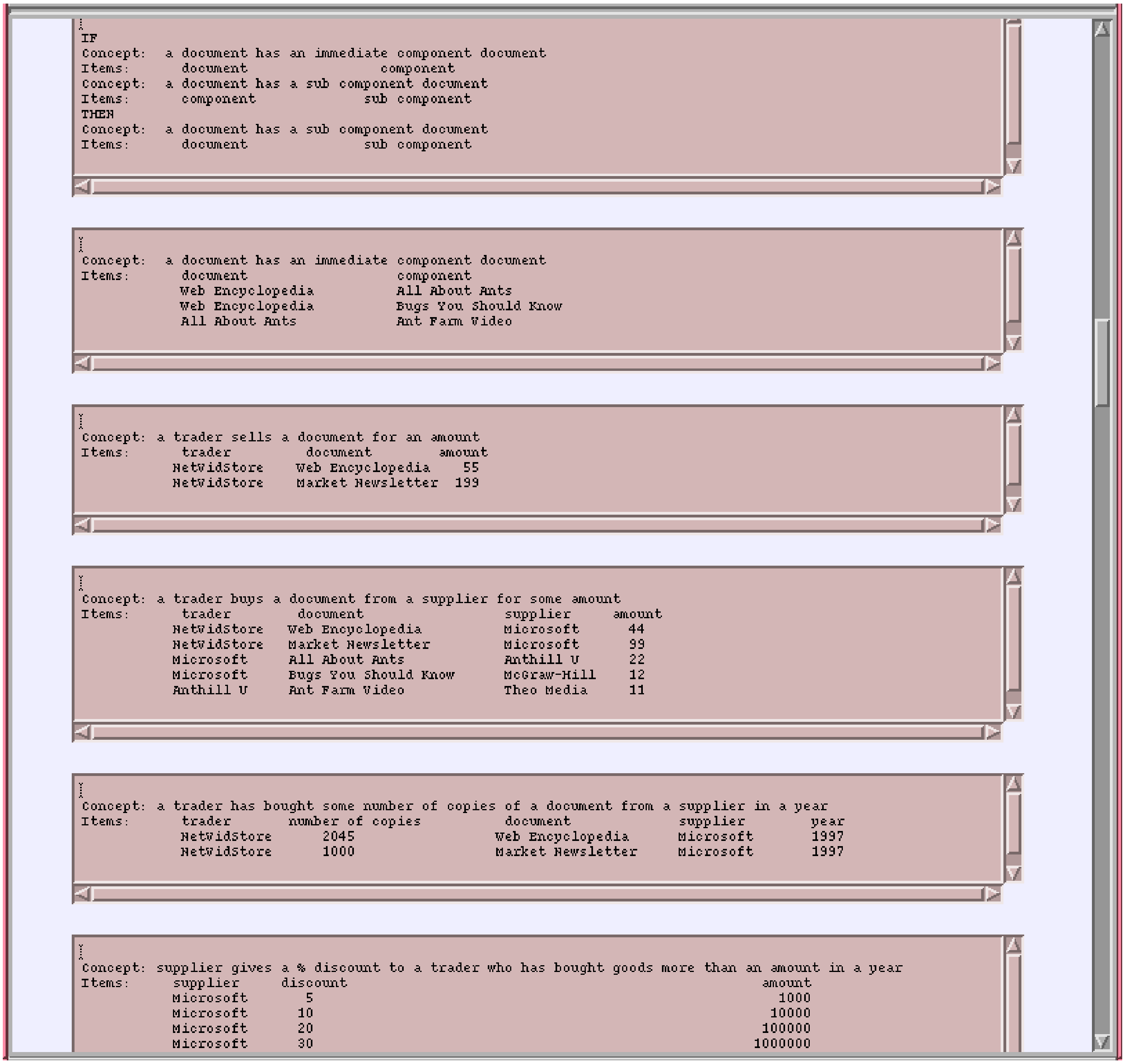
This page, and each subsequent page, shows a suggested action, followed by a **GO** button. Clicking on the suggested action shows a menu of alternative actions. If we leave the suggested action unchanged, and we click on the **GO** button, the result is an agent page.

On this page, the author-user views or changes an agent. The agent's policy is written as IF-THEN rules, and as tables of facts, using a Concept-Items notation.

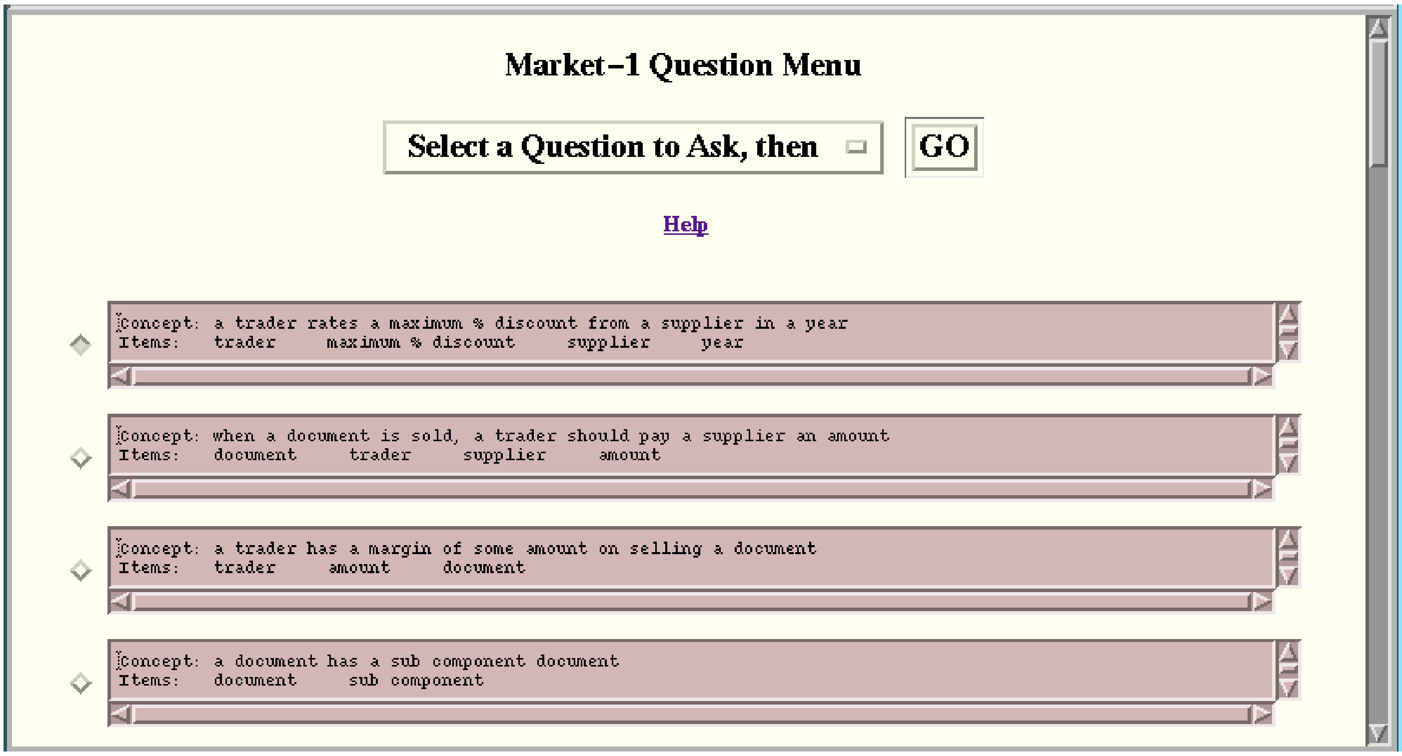


Moving downward on the page shows another rule, and some tables of facts in the Market-1 agent.

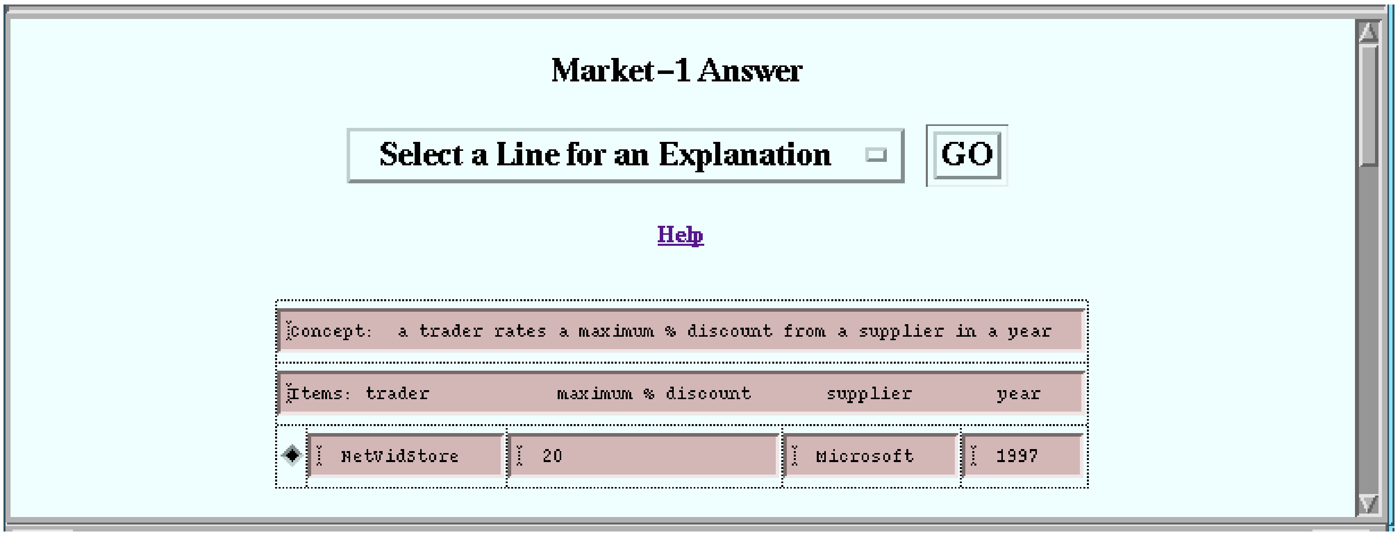
The rule is used to reason about arbitrarily deep nesting of documents. When the IKM uses such a rule, it automatically supplies the necessary control [3].



Returning to the top of the page, and asking for a menu of questions that the agent can answer gives a list of questions like this. The first question is selected.

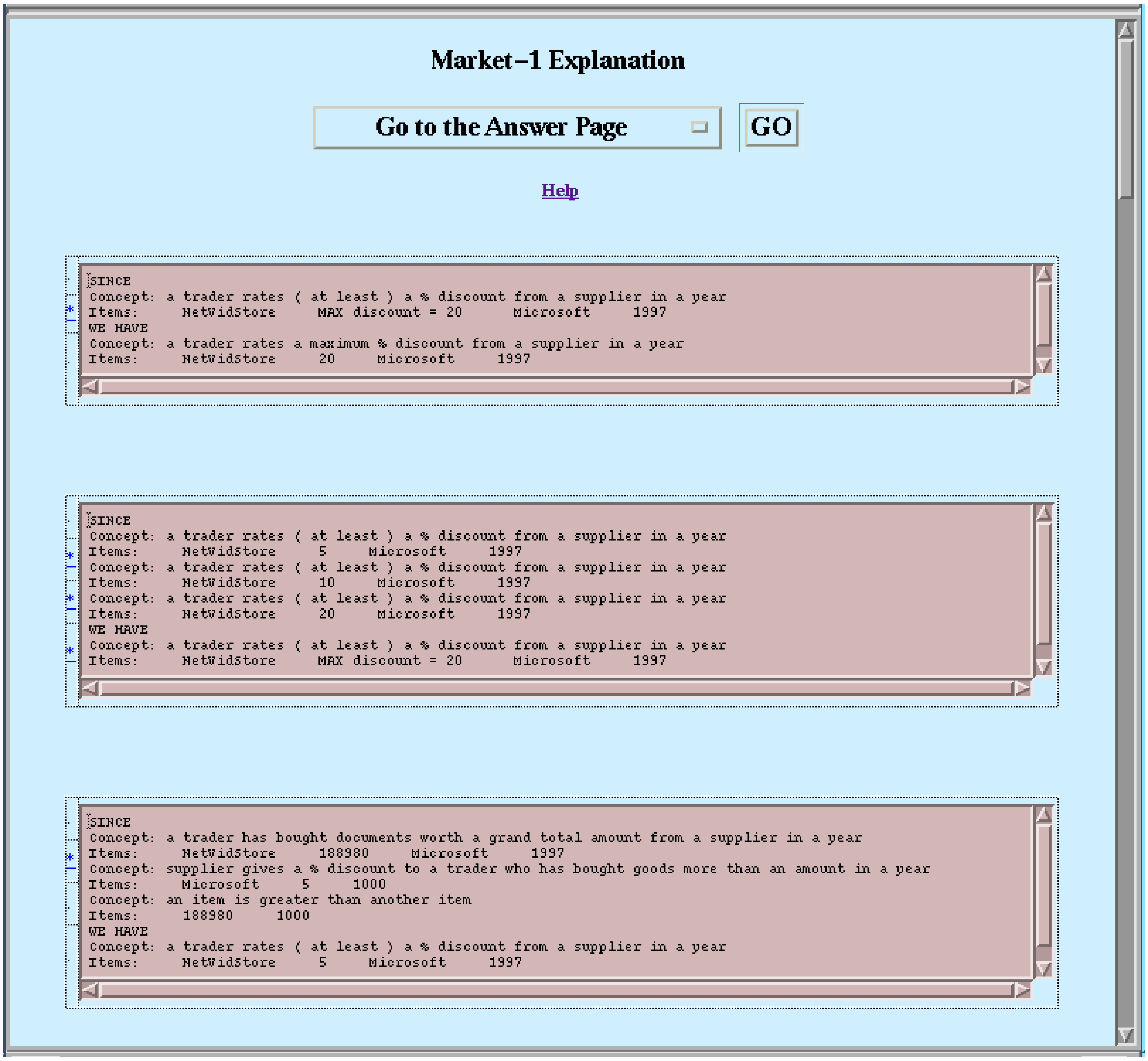


Clicking on the **GO** button gives this answer.

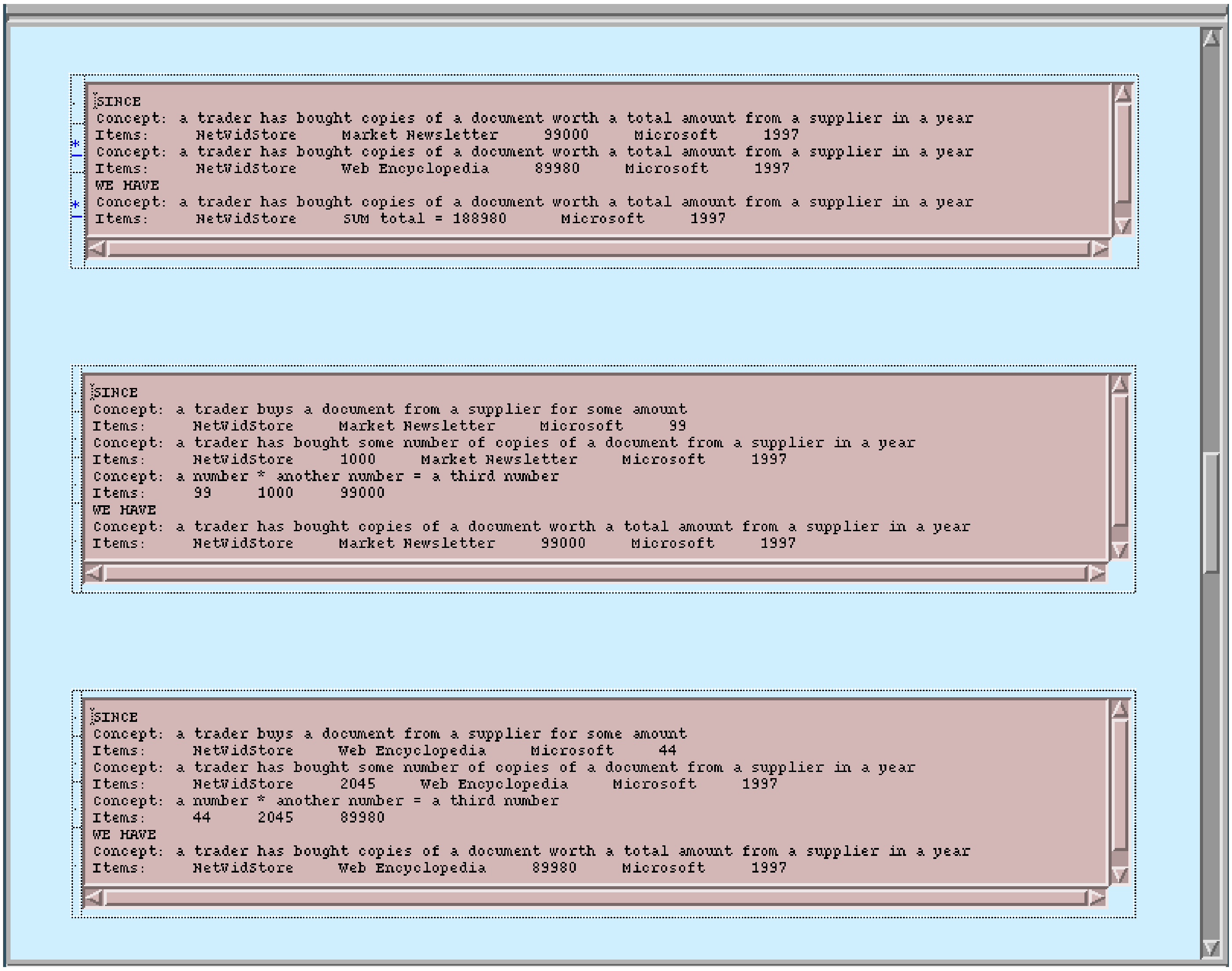


In this simple example, there is only one row in the answer table.

Asking for an explanation of the answer row gives a step by step account of how the agent found the answer.



In the explanation, we can click on a *****-symbol to drill down into more detail.



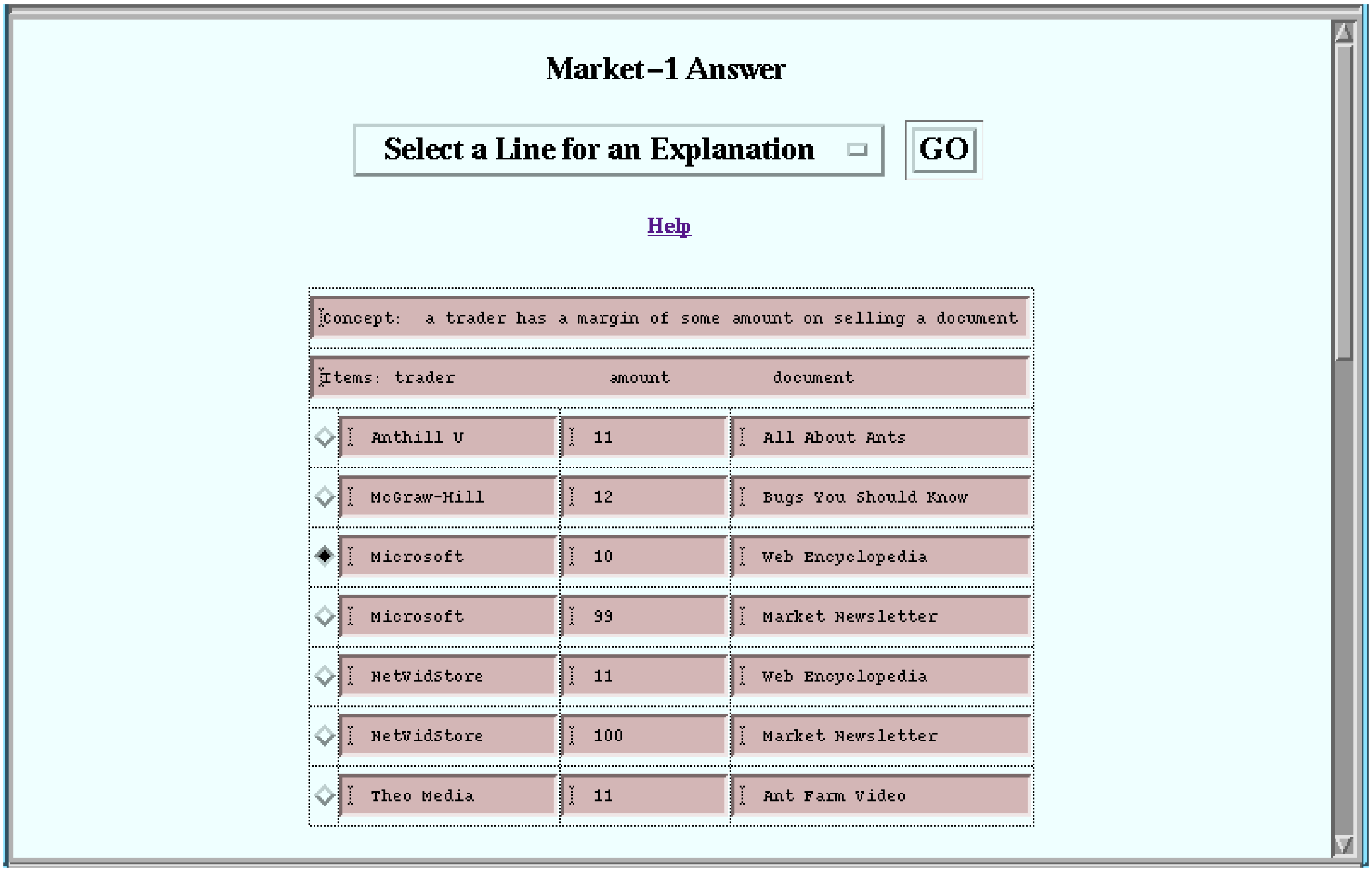
```
SINCE
Concept: a trader has bought copies of a document worth a total amount from a supplier in a year
Items:     NetWidStore     Market Newsletter     99000     Microsoft     1997
Concept: a trader has bought copies of a document worth a total amount from a supplier in a year
Items:     NetWidStore     Web Encyclopedia     89980     Microsoft     1997
WE HAVE
Concept: a trader has bought copies of a document worth a total amount from a supplier in a year
Items:     NetWidStore     SUM total = 188980     Microsoft     1997
```

```
SINCE
Concept: a trader buys a document from a supplier for some amount
Items:     NetWidStore     Market Newsletter     Microsoft     99
Concept: a trader has bought some number of copies of a document from a supplier in a year
Items:     NetWidStore     1000     Market Newsletter     Microsoft     1997
Concept: a number * another number = a third number
Items:     99     1000     99000
WE HAVE
Concept: a trader has bought copies of a document worth a total amount from a supplier in a year
Items:     NetWidStore     Market Newsletter     99000     Microsoft     1997
```

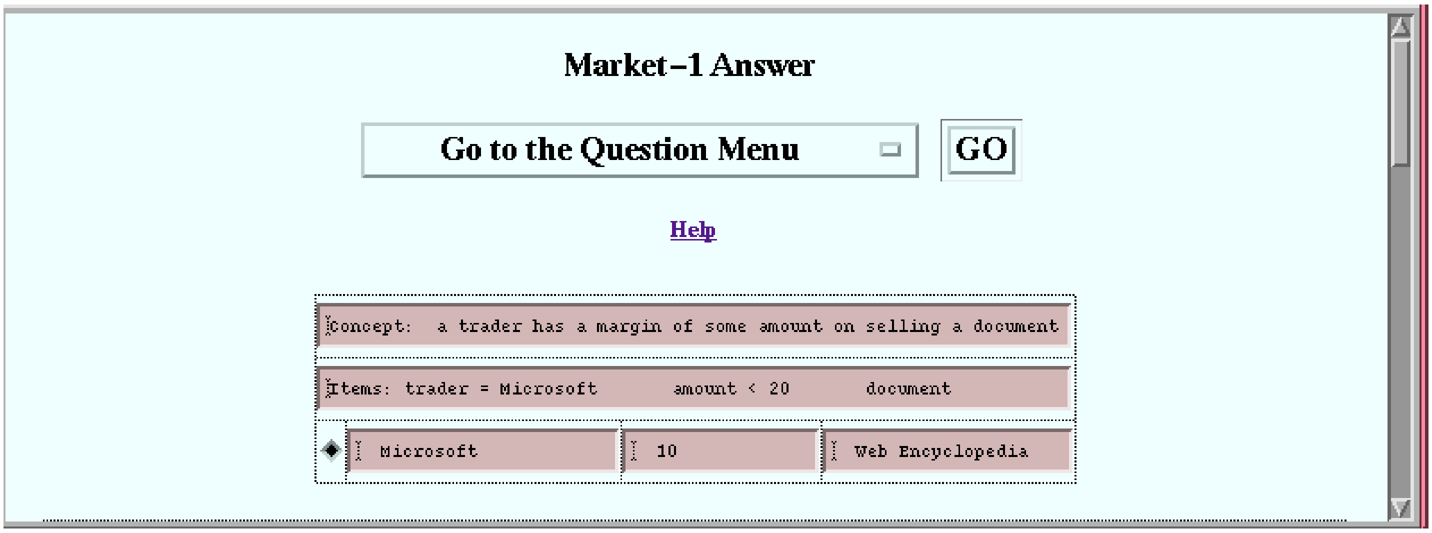
```
SINCE
Concept: a trader buys a document from a supplier for some amount
Items:     NetWidStore     Web Encyclopedia     Microsoft     44
Concept: a trader has bought some number of copies of a document from a supplier in a year
Items:     NetWidStore     2045     Web Encyclopedia     Microsoft     1997
Concept: a number * another number = a third number
Items:     44     2045     89980
WE HAVE
Concept: a trader has bought copies of a document worth a total amount from a supplier in a year
Items:     NetWidStore     Web Encyclopedia     89980     Microsoft     1997
```

Clicking on the browser's **Back** button returns to the *-symbol. Clicking several times on the **Back** button takes us back to the question menu page.

Asking a new question gives the answer



A menu question can be specialized before it is asked, to get for example just



Asking for an explanation gives a step by step account like this

This completes a simple Author-User example session. There are additional pages, in particular a check page, that gives smart help comments on a partly written agent. There is also support for complex transactions, and for getting an explanation before a transaction is made.

5 Summary and Directions

A change in the way human knowledge is represented in software can trigger a major change in business practices. Specifying all the future ways in which items in a digital library will be assembled and billed for amounts to a major software engineering problem. The Internet Knowledge Manager is designed to solve this problem, by opening up a new way of using the Internet. Up to now, the Internet and the Web have mainly partitioned people into *authors* (programmers and Web page designers) and *users*. The IKM allows a library professional to specify an agent that implements a rights and billing policy, using his or her own English terminology and an ordinary Web browser. A nonstandard approach to computing with natural language allows the IKM to explain, in English, the sometimes complex decisions that an agent makes in applying a policy. The IKM can be used in a buy server and Cryptolope architecture.

Future directions for the IKM include knowledge based mining of warehoused data.

References

[1] Safeguarding Digital Library Contents and Users: Assuring Convenient Security and Data Quality. Gladney, H. M. and J. B. Lotspiech. D-lib Magazine, www.dlib.org, May 1997.

[2] Apt, K., H. Blair and A. Walker. Towards a Theory of Declarative Knowledge, In: Foundations of Deductive Databases and Logic Programming, J. Minker [Ed.], Morgan Kaufman 1988.

[3] Walker, A. Backchain Iteration: Towards a Practical Inference Method that is Simple Enough to be Proved Terminating, Sound and Complete. Journal of Automated Reasoning, 11:1-22. 1993.