

# An Expert System Delivery Environment for the World Wide Web

Max Bramer

School of Computing, University of Portsmouth, Lion Terrace,  
Portsmouth PO1 3HE, United Kingdom

[max.bramer@port.ac.uk](mailto:max.bramer@port.ac.uk)

<http://www.tech.port.ac.uk/~bramerm/>

**Abstract.** AdviceWeb is a delivery environment aimed at enabling a wide range of high-quality rule-based advisory expert systems to be created and made available free of charge to users via the world wide web. Applications have a uniform 'look and feel', do not require any special software to use and are fully integrated with other sources of information on the web. A rapid application development facility is provided which automatically converts modules written in a simple scripting language into rules and questions for a production rule interpreter running on the AdviceWeb server.

## 1 Introduction

Rule-based Expert Systems were one of the most visible and most successful manifestations of Artificial Intelligence in the 1980s and 1990s, with many successful applications built, often with substantial commercial value.

The explosive growth of the World Wide Web in the last few years, combined with the much greater availability of personal computers and relatively high-bandwidth communications in both homes and offices has led to a widespread demand for (and expectation of) reliable information on any topic provided free of charge on request, delivered in a uniform way by a standard web browser.

Given this background it may seem surprising that there are very few Expert Systems available on the web. Most of the information available is static, and thus the same for all users. Information tailored to the needs of an individual (if only in a very basic fashion) is provided on sites run by retailers, bookshops, auction companies, travel companies and the like, where basic information provided by the user is matched against large databases of information about available products, holidays etc. The reasoning used by such systems is too shallow and the databases too volatile to justify calling them Expert Systems. However, the popularity of such

---

Please use the following format when citing this chapter:

Bramer, Max, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 129–136

sites with Internet users suggests that there could be a high demand for advisory systems on more complex topics that are personalised to the individual, rather than simply extracts from textbooks or general guidance applicable to everyone.

Even if the Expert Systems developed in past years were still available, were all free of charge and could be ported to a PC environment, they would still suffer from the problems that each one would require a separate download of files to a user's PC, they would all have different interfaces and experience gained from using one system would not make it any easier to use another.

The AdviceWeb project ([www.adviceweb.org](http://www.adviceweb.org)) aims to provide a delivery environment for rule-based advisory expert systems on a wide range of topics to anyone with a standard web browser. Use of the system can be anonymous. There is no need to register, no charge for users and no advertising. There are additional facilities available to registered users, but registration is free and requires only a name and email address to be given, both of which can be anonymised by those concerned about their privacy.

The systems all have a common 'look and feel' designed to appear natural for those already familiar with web-based applications and a standard interface, so that experience gained with using the facilities for one application will automatically transfer to any other application.

A major design aim has been to provide facilities for those who are not computer specialists to develop modules in a straightforward fashion without using any web development tools. Modules are written in a simple scripting language using only a standard text editor and are then uploaded to the AdviceWeb server using a standard web browser. The uploading process automatically invokes a translation program to convert the script into the rules and questions needed for the corresponding expert system module. When the user runs a module a production rule interpreter (written in the server side language PHP) is automatically invoked. This leads to a set of web pages being generated automatically and displayed as a frameset in the user's web browser. The user sees a sequence of such web pages, each with an accompanying question to be answered and supporting information. The sequence of questions is not fixed. Each question depends on the answers previously given.

It is hoped that the relatively little effort involved for the subject specialist to create quite an elaborate rule-based advisory expert system of this kind will encourage accredited module writers, known as *Advice Providers*, to develop not just one system but a suite of systems on a topic of interest. Advice Providers are envisaged as professional bodies or government agencies, rather than commercial companies.

## 2 AdviceWeb: Basic Use

One of the design aims of the AdviceWeb system is that it should be possible for the module writer, known as the Advice Provider, to construct a module with the minimum of effort and prior training. Ideally Advice Providers should be domain experts not programmers or AdviceWeb specialists.

To this end modules are not written as sequences of web pages, although that is the form in which they are presented to users. Modules are written in a simple scripting language called AWSL (AdviceWeb Scripting Language). AWSL was originally based on the scripting language for the SPSS flowcharting package *allClear 3.5* but has been extended with many additional facilities using tags in a similar way to HTML. Scripts are written using a text editor such as NotePad and are uploaded to the AdviceWeb server using a standard web browser.

Figure 1 shows a very basic but complete AWSL script for a simple application concerning a possible tax concession for providing a welfare service in the UK. The example leaves out many features of the scripting language, which will be explained later, but is sufficient for purposes of illustration at this stage.

---

```

Are you providing welfare?
  (Yes)
    Is service available to all distressed people?
      (Yes)
        Is the service subsidised > 15% below cost?
          (Yes)
            Is it your policy to subsidise the service?
              (Yes)
                Concession applies
              (No)
                -lab1
                Concession does not apply
          (No)
            > lab1
        (No)
          > lab1
      (No)
        > lab1
  
```

---

**Fig. 1.** A Basic AdviceWeb Script

The script consists of a sequence of questions and answers. The AdviceWeb system (and the reader) can tell which answers are associated with each question by the indentation of the lines. As well as questions and answers there are lines of text, such as “Concession applies”, which are called *advice*. The `-lab1` line signifies a label named `lab1`. The `>lab1` lines signify ‘jump to label `lab1`’.

A script file is uploaded to the AdviceWeb server and converted to an AdviceWeb *module* simply by completing an onscreen form in a standard web browser. Completing this form enables the Advice Provider to associate further information with the module, including its name, short and long descriptions and optionally the URLs of a home webpage and a Frequently Asked Questions page.

The uploading process automatically invokes the AdviceWeb script converter, which checks the script for syntactic errors, such as questions with only one answer or jumps with no associated labels. Syntactically valid scripts are automatically converted into rule and question files which together comprise the AdviceWeb

module that is stored on the server. Modules are written in a special production rule language which will not be described here.

When a user runs a module the associated rules and questions, together with the user's answers are processed by a forward chaining production rule interpreter, which automatically generates a set of web pages and displays them as a frameset in the user's web browser.


The user sees a sequence of web pages, each one asking a question and requesting an answer, either from a list of multiple-choice options or by entering a value (name, annual salary etc.). Instead of answering the question the user can choose to undo any number of previously answered questions or to restart the module. As the consultation session progresses a cumulative set of 'Advice Notes' is built up to take away. None of the AdviceWeb screens are pre-stored. They are all generated automatically from the script files and other information provided during uploading. Many of the screens have additional information to help answer questions and/or links to other (pre-stored) web pages, e.g. on a company or advice bureau website. The intention is that the advice given is justified by links to regulations, legislation, codes of practice, guidance notes and the like.

### 3 Associating Further Information With Questions and Advice

The following sample consultation illustrates some of the facilities available in AdviceWeb and how the Advice Provider can make use of them by augmenting the basic kind of script shown in Figure 1 using a markup notation.

The example module chosen is DPA, which gives advice on the disciplinary procedures for a fictitious university.

The module is started by selecting it from a menu of available modules. The startup screen is shown as Figure 2 (all screenshots have been converted to black and white for this paper).

<p><a href="#">Select New Module</a></p>	<p><b>AdviceWeb: Sharing Expertise</b> Disciplinary Procedures Adviser</p> 	
<p><b>Additional Information</b></p> <p>The DPA can give useful advice on a number of topics concerning the disciplinary procedures. These are 1) what type of disciplinary panel should be convened 2) who will sit on the panel 3) the procedure for handling appeals</p> <hr/> <p><a href="#">AdviceWeb Information Page</a> <a href="#">Module Home Page</a></p>	<p>On which aspect of the disciplinary procedures do you need advice?</p> <p><input checked="" type="radio"/> disciplinary panel type  <input type="radio"/> disciplinary panel composition  <input type="radio"/> appeals</p> <p style="text-align: center;"><input type="button" value="Submit"/></p>	<p style="text-align: center;"><b>Advice Notes</b></p> <p>A demonstration module which gives advice on the disciplinary regulations for a fictitious University</p> <p>Consultation continuing...</p>

**Fig. 2.** Startup Screen for Disciplinary Procedures Adviser Module

The module's initial question is "On which aspect of the disciplinary procedures do you need advice?". Selecting the answer 'disciplinary panel type' and pressing the Submit button produces Figure 3.

The figure shows three vertical frames. From left to right these are the *Information Window*, the *Question Window* and the *Advice Notes Window*, respectively. The framesets for each module have a uniform style, layout, colour scheme etc. so familiarity gained by the user with one module is readily transferred to others. All questions and where applicable a multiple-choice list of possible answers appear in the middle window. Any additional information relating to a question is displayed in the Information Window, together with links to the module home page, FAQ page and search engine, if they exist, and a link to the system information page, which contains a list of the modules available to the user.

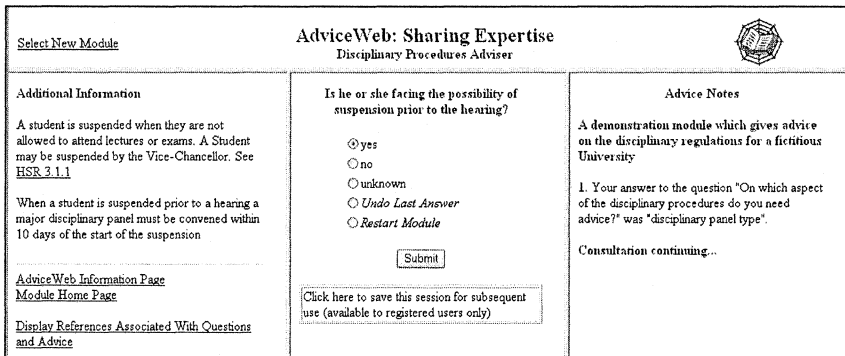


Fig. 3. DPA Example – Illustrating Additional Information and Web Link

This screen again illustrates Additional Information associated with a question. In this case there is also a hot link to 'HSR 3.1.1'.

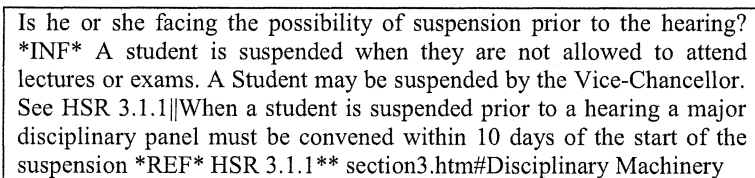


Fig. 4. Example of a Question Augmented with Markup

Figure 4 shows the markup needed in the AWSL script to produce this effect when the question is asked. The question 'Is he or she facing the possibility of suspension prior to the hearing?' is followed by an *information tag* \*INF\*, followed by the information itself (the two vertical bars signify two new lines).

This is followed by a *reference tag* \*REF\*, followed by a *reference string*, in this case HSR 3.1.1, followed by a URL 'section3.htm#Disciplinary Machinery'.

The reference string gives the text of the hot link in the information window. Any number of references can be associated with a question. Although absolute URLs are permitted, relative addresses are generally used. Any URL that does not

begin with http:// is automatically prefixed by a ‘webstem’ chosen by the Advice Provider when the script is uploaded to the AdviceWeb server.

Returning to the example given in Figure 3, answering ‘unknown’ leads to Figure 5. The paragraph numbered 3 in the Advice Notes window gives not only advice but two hot links to sections of the disciplinary regulations. Clicking on these will produce the text of the relevant part of the regulations to justify the advice given.

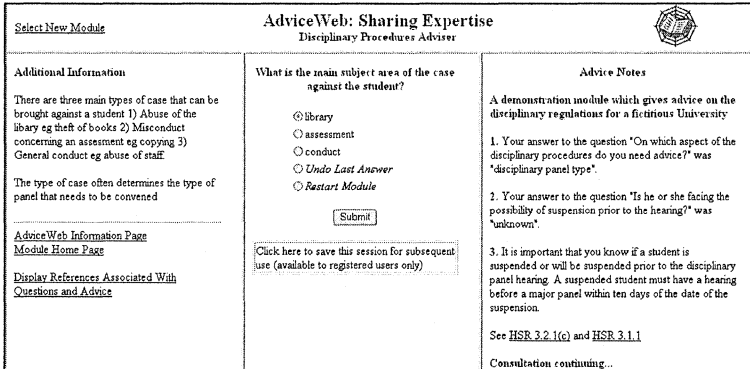


Fig. 5. DPA Example – Advice Notes with Web Links

The marked up advice text from the AWSL script file is shown in Figure 6. The advice text is followed by two reference tag + reference string + URL combinations. Any number of references can be associated with an advice text.

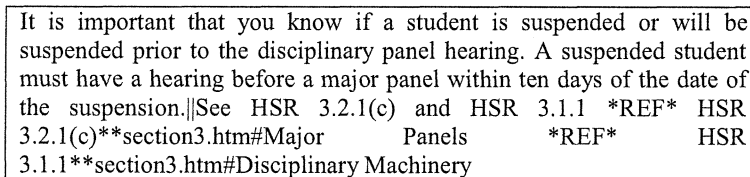


Fig. 6. Example of Advice Text with Markup

As a consultation progresses a references page is generated giving a cumulative list of the references displayed in the information and advice notes windows. Registered users can save an uncompleted consultation session at any point (for example if they have to break off midway or need to find out information from some other source before continuing). Saved sessions are date and time stamped and can be loaded whenever the module is started in future. When a session is completed registered users have the option to have the Advice Notes of their consultation, including all associated web references, sent to them by email. Unregistered users can simply cut and paste the Advice Notes into a word processor and print off a copy.

## 4 Additional Facilities

### 4.1 Using Variables in Advice Modules

An important feature of many Expert Systems is the ability to read in and output variables, as well as performing arithmetic on numerical variables. AdviceWeb has facilities to do all of these. Figure 7 shows a question with a multiple-choice answer augmented so that the answer is assigned to a variable, followed by an instruction to ask the user to input a value and assign it to a variable. A line of the latter kind can be used anywhere a piece of advice would be valid, but is treated as a question.

What is the VAT rate? *VAR* vrate
INPTEXT What is your annual salary? *VAR* y

Fig. 1. Examples of AWSL Script Lines that Refer to Variables

Both questions use the *variable tag* \*VAR\*. Any or all of the variable, information and reference tags can be associated with any question.

A value can also be assigned to a variable by a LET instruction, which can be placed anywhere a piece of advice would be valid. Arithmetic expressions of any complexity can be used. The value of a variable or an arithmetic expression can be output in a piece of advice. Figure 8 illustrates both of these.

LET z = (x+y)/3+z-y-4.2+2*q
Your profit of [x] pounds will incur a tax liability of [(x-allowance)*rate] pounds

Fig. 2. LET Instructions and Advice Using Variables

### 4.2 Fine-Tuning Modules

Another desirable feature of an Expert System delivery environment aimed at Advice Providers who are not computer specialists is the ability to fine-tune a module without having to upload it again for every change. AdviceWeb allows Advice Providers who run their own modules to make changes to them onscreen while they are running.

Whenever a module is run by the Advice Provider who uploaded it, an ‘Edit’ hotlink or button is placed next to the ‘Additional Information’ line on the information window, the text of the question in the question window and every piece of advice in the advice notes window.

Taking the example shown in Figure 5 for illustration, clicking on either of the ‘Edit’ links provided for the Advice Provider in the leftmost two columns produces an ‘edit form’ in the middle column, as shown in Figure 9.

This enables the wording of the question to be adjusted, the additional information to be changed (or deleted) and references to be added if required.

Clicking on the 'Edit' button, which can be seen in the advice window will give another edit screen, which enables the wording of the advice to be modified and the associated references to be changed or deleted.

This onscreen editing facility makes it straightforward to carry out minor maintenance of modules. Advice Providers who prefer not to use the information and reference tag facilities described in Section 3 do not have to do so, as the necessary information and web links can be provided onscreen instead.

AdviceWeb: Sharing Expertise Disciplinary Procedures Adviser		
<p>Select New Module</p> <p><u>Additional Information Edit</u></p> <p>There are three main types of case that can be brought against a student: 1) Abuse of the library eg theft of books 2) Misconduct concerning an assessment eg copying 3) General conduct eg abuse of staff.</p> <p>The type of case often determines the type of panel that needs to be convened.</p> <p><u>AdviceWeb Information Page</u> <u>Module Home Page</u></p> <p><u>Display References Associated With Questions and Advice</u></p>	<p>Question, Additional Information and References Editing Facility For Advice Providers Only</p> <p>Edit question, additional information and/or references below and press submit</p> <p><b>QUESTION</b></p> <p>What is the main subject area of the case against the student?</p> <p><b>ADDITIONAL INFORMATION</b></p> <p>There are three main types of case that can be brought against a student: 1) Abuse of the library eg theft of books 2) Misconduct concerning an assessment eg copying 3) General conduct eg</p> <p><b>REFERENCES</b></p> <p>Submit   Reset</p>	<p><b>Advice Notes</b></p> <p>A demonstration module which gives advice on the disciplinary regulations for a fictitious University</p> <ol style="list-style-type: none"> <li>Your answer to the question "On which aspect of the disciplinary procedures do you need advice?" was "disciplinary panel type".</li> <li>Your answer to the question "Is he or she facing the possibility of suspension prior to the hearing?" was "unknown".</li> <li>It is important that you know if a student is suspended or will be suspended prior to the disciplinary panel hearing. A suspended student must have a hearing before a major panel within ten days of the date of the suspension.</li> </ol> <p>See <a href="#">HSR 3.2.1(c)</a> and <a href="#">HSR 3.1.1</a></p> <p>Edit</p> <p>Consultation continuing...</p>

Fig. 9. Example of Onscreen Editing for a Question (Advice Providers Only)

### 4.3 Linking Modules

Two of the design aims of AdviceWeb are that it should be possible to create substantial systems without significant programming skills and that Advice Providers should find it realistic to develop a suite of modules, not just one.

Both of these aims point to the need for a facility to link modules to each other, thus allowing tasks to be decomposed into smaller subtasks and standard modules to be developed for use in more than one application.

An AWSL script can include a call to another module. This can be placed anywhere a piece of advice would be valid. The AdviceWeb rule interpreter distinguishes between occurrences of the same variable and label names in different modules.

## 5 Current Status

The AdviceWeb system is fully implemented and currently undergoing trials. A range of test modules has successfully been created by automatic conversion from AWSL script files. The largest group of modules links together approximately 20 modules written independently for related tasks. Feedback from developmental testing has been favourable. Further modules are being developed and Advice Providers are currently being sought for the live system.