WEBSELL: Intelligent Sales Assistants for the World Wide Web

Pádraig Cunningham, Ralph Bergmann, Sascha Schmitt, Ralph Traphöner,
Sean Breen, Barry Smyth

Trinity College Dublin, Computer Science, Technical Report

TCD-CS-2000-42

Abstract

Searching for and selecting complex products on the World-Wide-Web is a difficult task for consumers as well as for business professionals. The main reason for this well-known deficit in e-commerce is that, unlike normal business situations, there is no intelligent support or assistance for the user on the Web. This is particularly evident in the selection of products/services or when navigating through the complex space of available product information. Current product-oriented database search facilities are widely used on the Internet but recognised as limited in capability for sales support. In this paper we describe the objectives and achievements of WEB**SELL** an ESPRIT IV project (7/1998 - 1/2000) that set out to develop the next generation of intelligent sales support technology. WEB**SELL** draws on techniques from case-based reasoning, decision tree based protocol systems, user profiling and collaborative recommendation to produce virtual sales assistants that elicit users' requirements and identify products or services to meet these requirements.

1. Introduction

E-commerce offers many advantages and opportunities; however these benefits come at the cost of not having a human sales-person in the sales process. The sales-person facilitates the sales process by helping customers identify what their requirements are and then by identifying what products best meet these requirements. In the absence of this human support for the sales process in e-commerce there is a need to develop intelligent virtual sales agents that will fill these roles. This was the objective of the ESPRIT-IV project WEB**SELL** that completed in January 2000. The WEB**SELL** consortium includes four commercial companies and two Universities from Germany, Switzerland, and Ireland. WEB**SELL** has developed new intelligent sales support technology that supports Web shoppers in two aspects of the sales process which have been previously neglected:

- helping the user identify and articulate their requirements and
- identifying the products best suited to those requirements.

To enable maximum flexibility, the WEB**SELL** software has been designed as a modular architecture (see Figure 1) that includes components for:

- representation of product data and knowledge,
- dialogue with the customer (Pathways),
- product search through case-based retrieval,
- personalization and collaborative recommendation,
- product presentation,
- product customisation.

These components communicate with each other through specialised XML-based protocols and can be combined in a flexible manner. The different components draw on techniques from case-based reasoning, decision tree based protocol systems, user profiling, and collaborative recommendation.

The WEB**SELL** architecture is appropriate for a single business that is selling a range of products (e.g. technical equipment) or for a middleman that is brokering a range of products (e.g. hotel bookings). Building a solution for a specific client then corresponds to configuring a set of components and integrating the resulting system into the environment at the client's side. Whenever additional functionality is needed, new services may be added. Likewise, existing systems can easily be integrated by implementing a specific component organising the necessary communication between the external system and the affected WEB**SELL** components.

Today, the WEB**SELL** architecture has been used in a large number of applications that are in daily use, some of which are sketched in section 6 of this paper. Each of the following sections 2-5 gives a brief overview of one of the WEB**SELL** components.

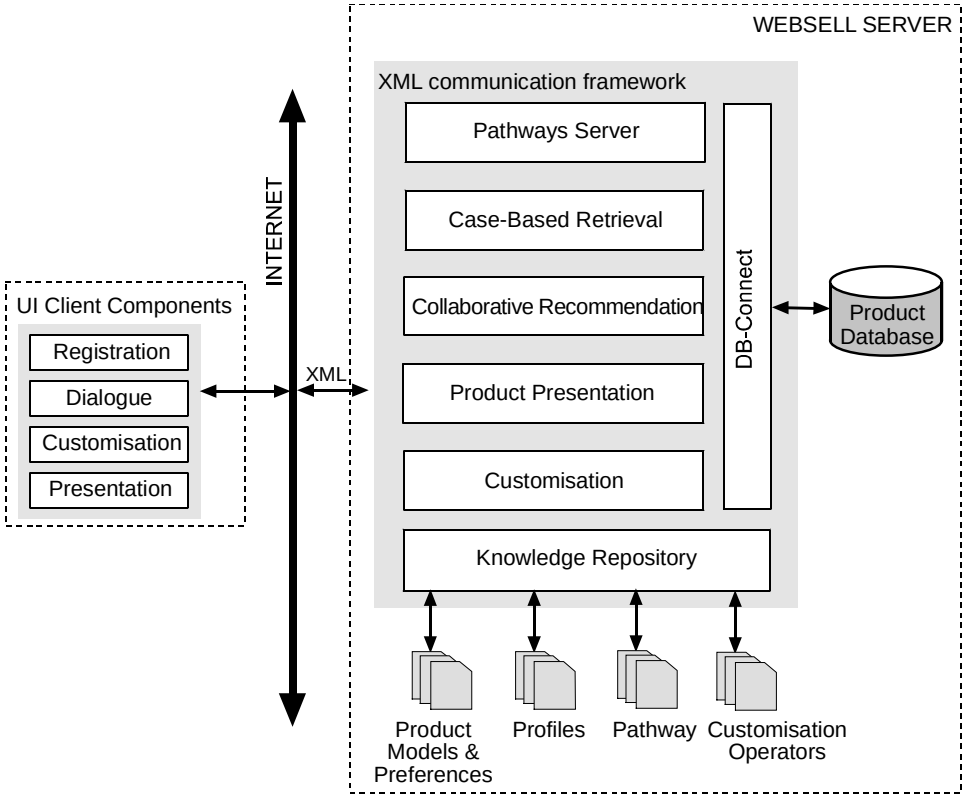

Figure 1. A functional view of the WEB**SELL** components.

2. Pathways: The Customer Dialog Component

A WEB**SELL** pathway is a decision tree that guides the user through questions, information and decision logic to arrive at a description of the user's requirements. Pathways are created using the Pathways Builder tool and executed or navigated using the Pathways Server. The output from the navigation of a pathway is the data gathered or computed during the run; the output is in a customisable format, usually XML.

Pathways allows developers to produce dialogues that will elicit a user's requirements using a combination of standard html pages with platform independent Javascript. The user can navigate forward and backwards through the dialog, changing answers if required.

2.1. Pathways Builder Overview

The Pathways Builder (see Figure 2) supports the developer in creating a web-based agent that gathers customer requirements. This agent, in effect, "interviews" the customer and reacts to the information provided either to directly recommend a product or service or to perform an intelligent search to find the most suitable product or service. A pathway can ask questions, perform calculations, offer information and apply rules at each point in the decision tree. The Pathways Builder is a Visual C++ application that gives drag-and-drop and tab-based features to support the creation of pathways by non-technical users. A single button-press will publish the pathway to the Pathways Server.

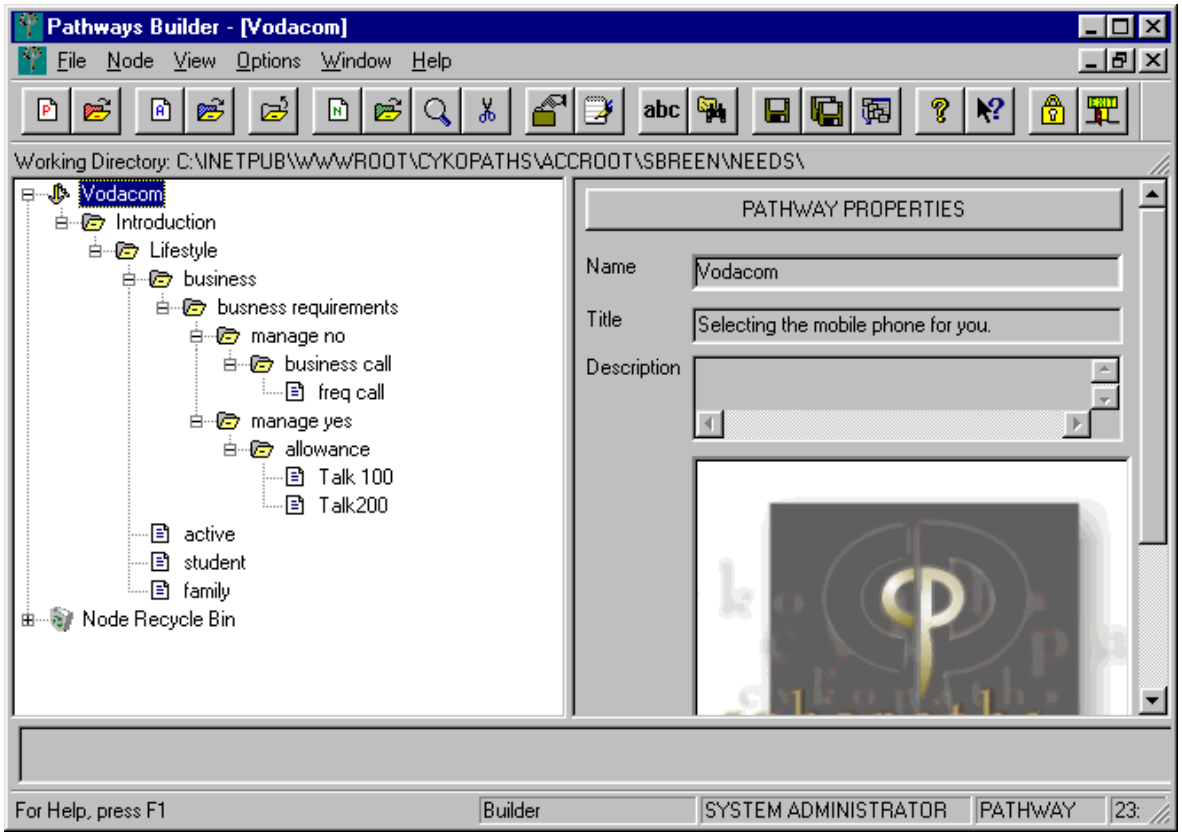

Figure 2. The Pathways Builder.

2.2. Pathways Server Overview

Pathways Server navigates through the decision logic in the pathway and produces the user interface in the customer browser. The pathway branches to different sub-trees depending on the input from the user or on values computed at points on the decision tree. A point (or node) on the decision tree can perform one or more of the following tasks:

- Display information to the user in textual, graphical, or animated format,

- Ask questions to the user with any of a range of standard controls (i.e. text, radio button, combo-box, etc.),
- Calculate equations or perform string additions as required,
- Perform a conditional or unconditional goto to jump to another node of the pathway,
- Produce a report from a designer-specified template and send this to the browser, to a server or to any valid Web address for processing.

The graphical presentation of the contents of a node is fully under the control of the designer. She can specify the fonts to be used, background colours or images as well as the graphics for any and all buttons. The custom report builder allows the designer to format the output for passing on to other components of the WEB**SELL** system.

3. Case-Base Product Retrieval

Retrieval and adaptation techniques from Case-Based Reasoning (CBR) have become very important techniques for realizing intelligent product recommendation agents [1,2,4,10]. The core of such applications is a product database that describes the specific features of each available product. When applying CBR, this product database is treated as a case-base, i.e., each product record in the database is interpreted as a case in a case-base. During the case retrieval phase, product cases are retrieved based on the *similarity* between the product features and the requirements elicited by the Pathways component. The similarity encodes the knowledge to assess whether a product is suitable for the customer's requirements. In the WEB**SELL** retrieval component, similarity is formalized through *similarity measures* that are modelled by combining several parametrizable *local similarity measures* for individual product features with a *global aggregation function*. Thereby global and individual preferences for product selection can be modelled.

The main purpose of the retrieval component is then to select from the product database a set of products with the highest similarity as computed by the similarity measure. The challenge is to realise *efficient* retrieval on a *large* and *highly dynamic* product database. The retrieval component provides different similarity-based retrieval algorithms such as

- complete brute-force search
- case-retrieval nets [4]
- similarity-based retrieval by approximation with SQL queries [7].

Recent applications have demonstrated the efficiency of the WEB**SELL** retrieval component with product databases containing several tens of thousands of products.

4. Profiling & Collaborative Recommendation

In addition to the case-based recommendation approach, WEB**SELL** also provides a facility for collaborative recommendation based on user profiling. In the context of WEB**SELL**, user profiles support the potential personalisation of all aspects of the sales process. A user profile stores the past e-commerce history of an individual user. User profiles are stored and maintained on the server as a profile database (see Figure 3). Each user is associated with a single profile, and each profile contains user information that can be separated into three basic category types:

1. Personal Information: This contains various personal details such as name, age, gender, home address, occupation, credit-card details etc.

2. Domain Preferences: This contains user information that is relevant to a particular domain. For example, for an online travel application the domain preferences might include information such as: the type of vacation that the user is interested in (relaxing versus activity, city versus country etc.); their preferred travel arrangements (airline travel with Virgin or British Airways); budget details (the package price should not exceed £2000).

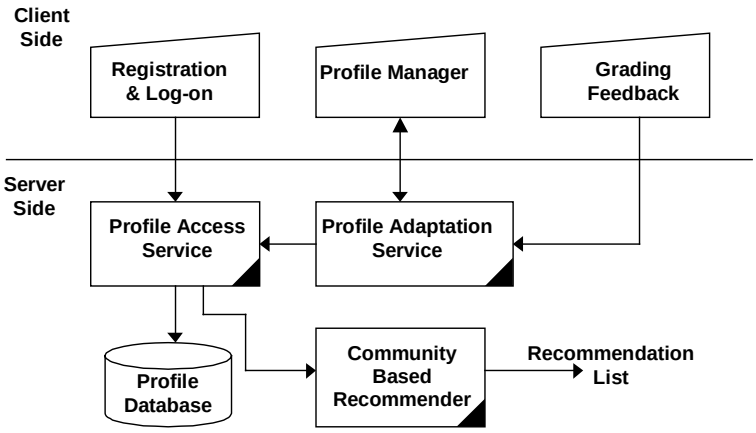

Figure 3. An overview of the WEB**SELL** profiling and collaborative recommendation service.

3. Selection Lists: This is the most important type of profile information from the collaborative recommendation viewpoint. Two selection lists are maintained. The positive selection list (+SL) contains a list of products that the user has expressed an interest in or purchased. The negative selection list (-SL) contains a list of products that the user has explicitly ignored in the past.

The collaborative recommendation service in WEB**SELL** is a recommendation scheme that allows products to be recommended to target users based on their user profile data, and in this sense the recommendations are personalised for the user in question. The detailed operation of the collaborative recommendation process is described in [8] and in [3]. The key to this form of recommendation is the ability to associate a target user with a group of other users that are similar in the sense that their profiles are similar to the target user profile. Typically, profile similarity is a measure of the correlation between the selection lists of two user profiles; users with a high degree of similarity tend to grade the same products in the same way. A group of users that are similar to the target user form a virtual community for the target, and recommendations are drawn from the profiles of the community members. The result is a list of recommendable products, which can be ranked according to, for example, the frequency of the product in community member profiles. Collaborative recommendation is a three-step procedure:

1. Identify the virtual community associated with a given target user.
2. Produce a ranked list of recommendable products. These are products that are listed in the positive selection lists of community members, but that are not contained within the selection list of the target user. The products are ranked according to their frequency of occurrence in the community.
3. Select the top *n* recommendable products as recommendations.

The final output of the collaborative recommendation service is a list of products, and ultimately these can be recommended directly to users or combined with the case-based reasoning recommendation. The collaborative recommendation service is responsible for identifying virtual communities (as groups of user ids) within the WEB**SELL** user population and for associating individual users with the appropriate community.

5. Customisation

One important objective of WEB**SELL** was to be able to support the sale of complex products requiring configuration or customisation. To date, sales success on the Web has been with simple fixed products such as books or music CDs or with very constrained alternatives of configurable products. The WEB**SELL** customisation component allows users to more flexibly and completely configure complex products such as holidays, insurance plans or technical equipment. The developed approaches have their origin in adaptation techniques from CBR. Two different approaches have been developed:

5.1 Operator-based Customisation

The operator-based customisation approach [5,6] supports interactive modification of products by the customer. After a best-matching product has been retrieved and presented to the customer, a set of *customisation operators* is provided, which may be applied to further customise the product. Each customisation operator encodes a particular atomic way of adapting certain products. The description of such an operator contains

- a precondition that specifies under which circumstances the product can be modified,
- a set of parameters to specify the details of the customisation
- an action part specifying how the product is affected by the customisation.

This customisation process is shown in Figure 4 where the retrieved product (left) is transformed into the desired target (right) by a series of operations. The customisation component enables the customer to navigate through the space of possible customised products and takes care of the applicability of operators, validity of parameter values, and the consistency of the adapted products.

In B2C scenarios, the operator-based approach is particularly suited to support products with limited customisation capabilities since otherwise the set of applicable operators overstrains the user. However, in B2B scenarios in which clients have expert knowledge about the products, the operator-based approach can be applied to more complex products as well.

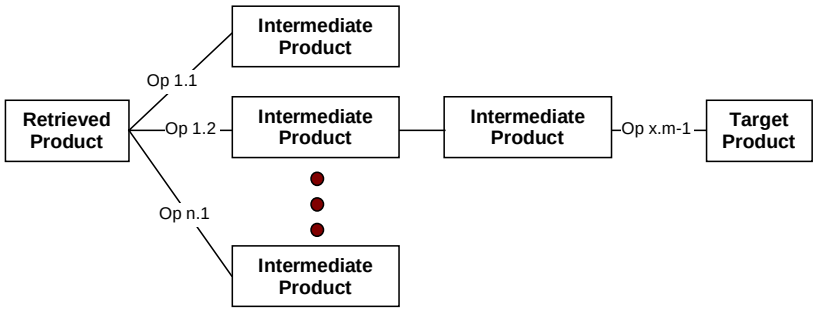

Figure 4. The operator-based customisation process.

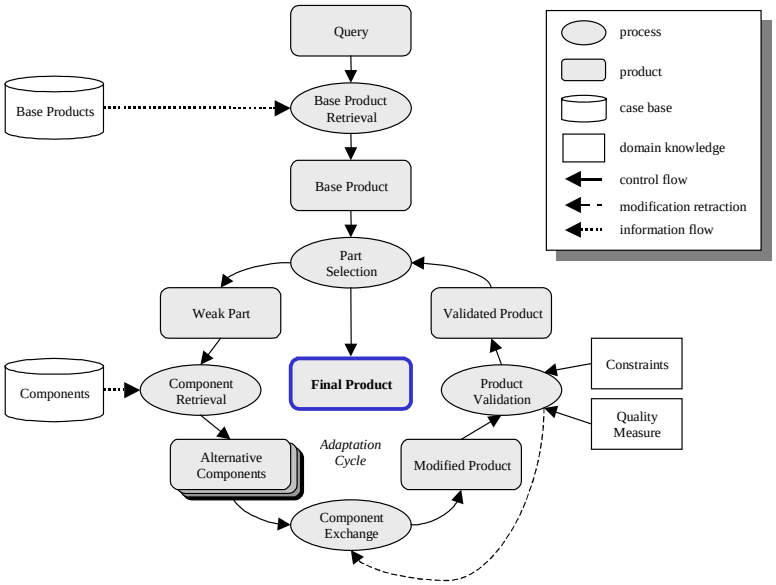

Figure 5. Incremental component replacement cycle.

5.2 Incremental component replacement

The incremental component replacement approach [9] is particularly suited to complex products that require sophisticated customisation. It is assumed that products are structured into sub-components, possibly in a hierarchical manner. Further, product databases with some pre-configured base products and individual sub-components are required. After retrieving the best pre-configured base product with respect to the customer's requirements, the product is customised by incrementally replacing sub-components by more suitable ones. The adaptation cycle depicted in Figure 5 shows the details of this replacement procedure. Components with a low similarity (*weak components)*, i.e., components that do not fulfil the customer's requirements well enough, are candidates for being replaced. By recursively applying CBR at the level of sub-component, alternative components are selected from the product database. Then, the weak component is replaced by an alternative component and the validity of the resulting adapted product is checked. During this validation, constraints that exist between the different components are evaluated. Violation leads to a backtracking to the component replacement step, giving the next best component a chance.

This adaptation cycle is executed several times. In every run the overall suitability of the product is increased. Generally, we can notice that the adaptation cycle implements a hill-climbing search to solve a combination of a constraint satisfaction problem and an optimisation problem for the product suitability (measured by the similarity). On the one hand, it has to find a combination of different components representing a working product, i.e., a product that fulfils all constraints. On the other hand, it has to find an optimal combination that fulfils the customer's demands as well as possible.

6. Demonstration Applications

None of the available demonstration applications exploit all the facilities that WEB**SELL** offers. So in this section we focus on the two key roles for WEB**SELL** emphasised in the introduction, that is in helping the users identify and articulate their requirements and in identifying products best suited to those requirements. The next section describes an application that uses the Sales Server for searching for matching products and section 6.2 describes the use of the Pathways dialogue manager for eliciting customers' requirements.

6.1. Virtual Letting Agent

The Virtual Letting Agent was developed by IMS MAXIMS for Hooke & MacDonald, a Dublin Estate Agency (www.hookemacdonald.ie). It is a system for finding apartments for renting. In this scenario the requirements elicitation process is straightforward so the user's input is captured in a single screen (see Figure 6). This set of requirements is passed to the case-based product retrieval engine on the server and the most suitable apartments available are returned.

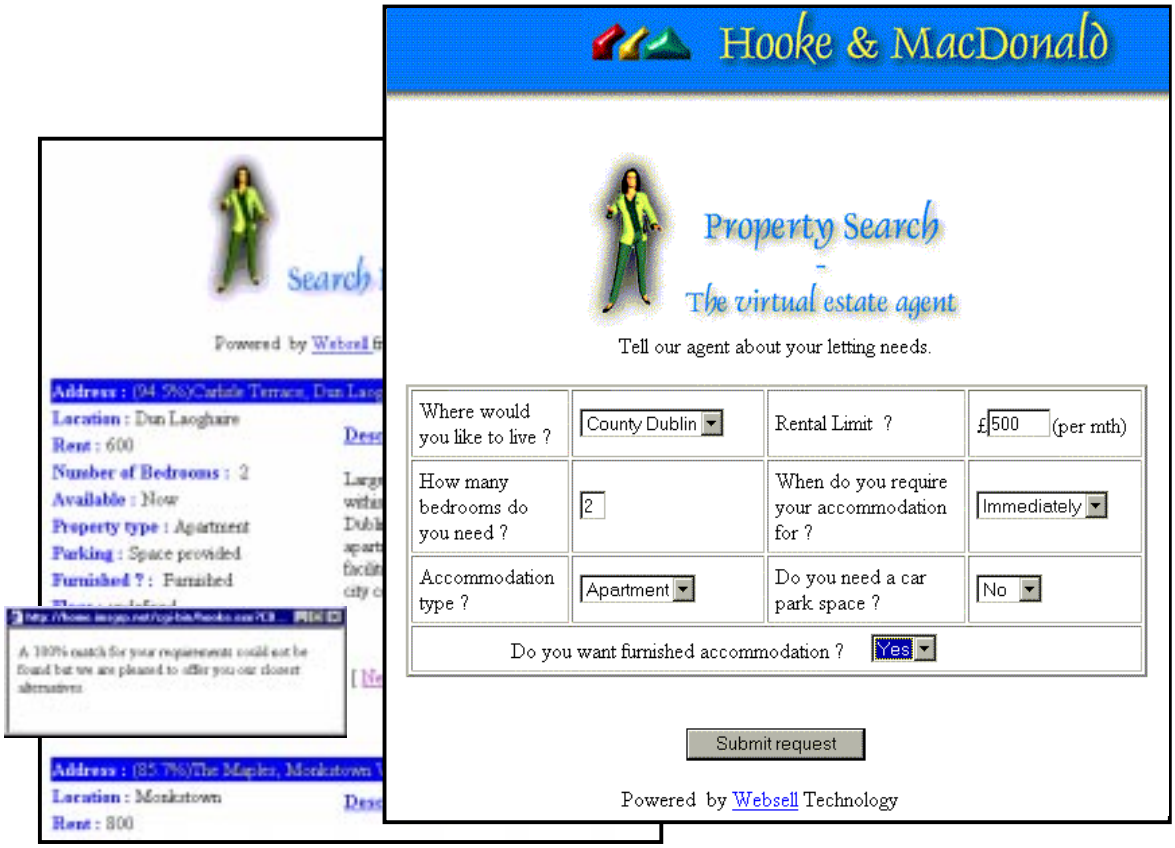

Figure 6. Some screens for the Virtual Letting Agent. The screen in the foreground is the requirements input screen and those in the background are the responses from the system.

The main source of intelligence in this system is the similarity measure (see section 3) that identifies good matches and the score from this metric is used to rank the apartments returned.

6.2. Selecting a Mobile Phone

In other scenarios, expressing the users' requirements will not be so straightforward and Pathways can be used to manage a dialog with the user to identify a requirements description. An

example of this is an application powered by Pathways for selecting a mobile phone demonstrated on www.cykopaths.com. This is a classic application where WEB**SELL** can assist because there is a difficulty in defining the user's requirements and in mapping these requirements to available phones and connection packages. As described in section 2, Pathways allows the requirements elicitation process to be expressed as a tree. The user starts at the root node and is led through a series of questions to elicit a complete requirements description. The questions can change dynamically depending on previous answers; questions from a typical node are shown in Table 1.

Table 1. Questions from an example node in the Pathways application for selecting a mobile phone.

Will you use the phone mainly during business times? (Y/N)
How many minutes do you expect to spend on calls per day?
Do you need to control the cost of your usage?

Complete interaction with Pathways involves traversing from the root to one of the leaf nodes of the tree answering questions and receiving clarification. Each node of the tree represents a decision point and the direction taken depends on the answers to questions or calculations based on data already available. Once a leaf node is reached a complete set of requirements is available and case-based product retrieval is used to select the best product as happens in the previous application.

7. Summary and Future Work

The objective with WEB**SELL** was to produce a set of tools that would extend the range of products and services that can be marketed successfully on the Web. The first challenge was to tackle scenarios where expressing the users' requirements in terms of product descriptions was difficult. Pathways achieves this and several demonstration applications can be seen at www.cykopaths.com.

The second challenge was to help users to find the best products matching their requirements and to provide a flexible mechanism to help the user customise these products. The WEB**SELL** solution to this is described in sections 3 and 5. Finally, the WEB**SELL** suite of tools is completed with a user profiling and collaborative recommendation facility that exploits the available data to bring customers together with products that should interest them (section 4).

In B2C e-commerce the challenge is to bridge the knowledge gap between the customer's world and the technical specifications of products. We feel that the WEB**SELL** tools are only a first step in this direction. In the future there is a need to develop more flexible mechanisms for managing dialogues with the user; at present the Pathways can only select between pre-configured dialogues. More important, we need to recognise the difference between the language of requirements and the language of product descriptions and develop mechanisms for mediating between these two representations.

8. Project Partners

tec:inno GmbH (Germany, Co-ordinating Partner)
Adwired One to One Communication AG (Switzerland)

IMS MAXIMS plc. (Ireland, Contractor)
IWT Magazin Verlag GmbH (Germany)
Trinity College Dublin (Ireland)
University of Kaiserslautern (Germany)

References

[1] Bergmann, R., Breen, S., Göker, M., Manago, M. & Wess, S., *Developing industrial case-based reasoning applications: The INRECA methodology*. Lecture Notes in Artificial Intelligence, LNAI 1612, Springer, 1999.

[2] Cunningham, P. CBR: Strengths and Weaknesses. In Proceedings of *11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, eds A. P. del Pobil, J. Mira & M. Ali, Lecture Notes in Artificial Intelligence 1416, Vol. 2, pp. 517-523, Springer Verlag, 1998.

[3] Hayes, C., Cunningham, P. Smart Radio: Building Music Radio on the Fly. *to be presented at Expert Systems 2000*, Cambridge, UK, December 2000.

[4] Minor, M. & Lenz, M. (This Volume). Textual CBR im E-Commerce, 2000.

[5] Schmitt, S. & Bergmann, R. Applying case-based reasoning technology for product selection and customization in electronic commerce environments. 12th Bled Electronic Commerce Conference, 1999.

[6] Schmitt, S., Maximini, R., Landeck, G., Hohwiller, J. A product customization module based on adaptation operators for CBR systems in e-commerce environments. *5th European Workshop on Case-Based Reasoning*, Springer Verlag, 2000.

[7] Schumacher, J. & Bergmann, R. An effective approach for similarity-based retrieval on top of relational databases. *5th European Workshop on Case-Based Reasoning*, Springer Verlag, 2000.

[8] Smyth, B. & Cotter, P. Surfing the Digital Wave: Generating Personalised TV Programme Guides using Collaborative, Case-Based Recommendation Techniques. ICCBR 99, 1999.

[9] Stahl, A. & Bergmann, R. Applying recursive CBR for the customization of structured products in an electronic shop. *5th European Workshop on Case-Based Reasoning*, Springer Verlag, 2000.

[10] Wilke, W. Knowledge management for intelligent sales support in electronic commerce. DISKI 213, Infix Verlag, 1999.