Representing Cases for CBR in XML[1]

Lorcan Coyle, Conor Hayes, Pádraig Cunningham

Department of Computer Science
Trinity College Dublin
{Lorcan.Coyle, Conor.Hayes, Padraig.Cunningham}@cs.tcd.ie

Abstract. Case Based Reasoning has found increasing application on the Internet as a shopping assistant for e-commerce stores. The strength of CBR in this area stems from its reuse of the knowledge base associated with a particular application, thus providing an ideal way to make personalised configuration or technical information available to the Internet user. Since case data may be one aspect of a company's entire knowledge system, it is important to integrate case data easily within a company's IT infrastructure, providing in effect a case-based *view* on relevant portions of the company knowledge base. We describe CBML, an XML-based Case Mark-Up Language we have developed to facilitate such integration. We will detail the benefits of our system for industry in general in terms of extensibility, ease of reuse and interoperability. The language allows us to make the formal definition of the structure of our cases completely independent of the application code. In this way we allow the structure and definition of our cases to be described and modified easily. Such a language would also allow cases to be exchanged between heterogeneous CBR systems. As an example of how CBML might be used we describe our research on a wireless Case Based assistant for the travel market. In this application user profiles are marked up as sets of cases in CBML.

Keyword. CBR, XML, Case Representation, Personal Travel Assistant

1 Introduction

Increasingly Case Based Reasoning (CBR) has found a role adding intelligent sales support in the realm of e-commerce. Generally, e-commerce shops set out to sell products without the intervention of a sales assistant. In the absence of human sales assistants there is a need for intelligent software assistants to facilitate the sales process. Since what are available are catalogue data and data on user behaviour and preferences CBR is an obvious technology to create these sales assistants. In this scenario, the obvious cases are descriptions of the commodities on sale and the task is to identify the case configuration that meets the user's requirements. These cases might describe package holidays, hardware configuration or as we will describe in this paper, flight arrangements.

However, despite the fact that Case Base Reasoning is a knowledge-rich methodology there is no standard means of representing the case data in a case base. It is up to the e-commerce developer to shape the case data from the company knowledge base. The manipulation of case data is dependent on the representation format chosen by the developer and this limits transformation of the data into a format suitable for the presentation layer, or its movement to another back-end component. Kitano and Shimazu have proposed that CBR applications have been too narrowly focused on domain specific problems [8]. They suggested that a CBR system should be viewed as a *medium* to be used in conjunction with the mainstream corporate information system. We would share this perspective, and we anticipate that a standard way of marking up cases will facilitate this. The standard proposed for marking up structured, knowledge-rich data is XML, the eXtensible Mark-up Language [12]. XML is a description language that supports meta-data descriptions for particular domains and these meta-descriptions allow applications to interpret data marked up according to this format. Using XML has many advantages: it allows for the application-independent exchange of structured rich data over existing network protocols. XML data may be easily mapped from one schematic to another using XSLT, the XML transformation language and it can be displayed in a customisable manner using the Extensible Style Language (XSL).

[1] The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged

Several XML-based CBR system have appeared [6,4] over the past few years. However, as Wilson has pointed out, the benefits that accrue to XML in general will not be fully passed on to the CBR community until a *standard* means of representing case data in XML is developed [16]. With such a standard, case data can be passed from one CBR system to another, irrespective of the complexity of the case definition. As part of the application described in Section 4 we have developed a set of interfaces that will facilitate the use of CBML data in any CBR system.

Developing such a standard is not a trivial task. A standard case representation format would have to be flexible enough to cater for complex case representations, such as hierarchical cases, as well as the commonly used 'flat' vector format. Our first attempt at developing a standard XML representation for CBR catered for the latter type of representation [6]. In Section 2 we will describe the drawbacks of our first implementation and the development of version 2.0 which caters for complex case representations. We will then describe how this version of CBML is used in a generic CBR application. In Section 3 we will describe a CBR-based travel assistant currently under development which utilises CBML version 2.0. We will discuss the advantages of using this format for complex travel data. In Section 4 we demonstrate how CBML is a key component of the information infrastructure of this application, and give examples of how it is used in marking up cases and then transforming them for the presentation layer. In Section 5 we conclude the paper and discuss possible enhancements to our schema such as tags that include similarity or adaptation information. Finally, in Appendix A we provide a list of online resources relating to CBML.

A Description of how CBML Works

The current CBML elaborates our earlier work [5,6] in developing an XML-based case representation language. The objective was that this language would provide similar functionality to the CASUEL [7] case representation language.

There were a number of weaknesses with CBMLv1.0, which the current version (2.0) overcomes. It allows better and more complete validation; new simple feature types, hierarchical case structures, and feature inheritance. There is also a significant difference in the syntax of the language. CBML version 1.0 required us to 'shoe-horn' data into a specific case format which was not easy to re-transform into other schema, a key advantage of XML representation. Furthermore, version 1 could only cater for flat case structures. Part of the drawbacks of the early version of CBML were due to the limited scope afforded by the XML document type declaration (DTD), the part of the XML document which defines the structure of the elements within the XML document. The early specification of the DTD was concerned primarily with structure and did not allow for a definition of data typing in any form. This was a major drawback for data that required type validation as well as structural validation. To overcome this we defined a separate case structure file for each CBML domain which allowed us to define some simple types. The XML community have since recognised the limitations of the early DTD model and have developed an alternative which allows for structural and type validation. This alternative is the W3C XML Schema specification [13]. XML Schema allows us to create much tighter restrictions. The CBML specification document is written in XML Schema and all case structure documents must follow this schema exactly to be considered valid CBML. This document is static and globally available (see Appendix A for details of where this is located).

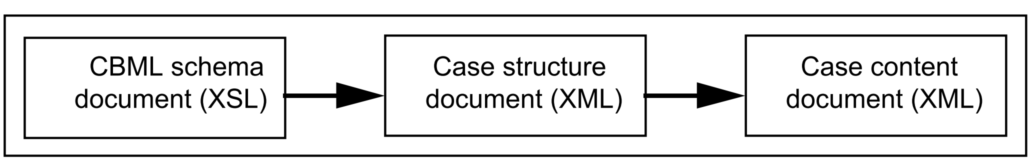

Fig. 1. The CBML document hierarchy. As the case structure document is parsed into memory it is validated against the CBML schema. Since cases are in turn validated against this structure definition there is a logical hierarchy between the three documents; the CBML schema validates the case structure document which in turn validates case content documents.

Two documents are needed to represent a case in CBML, the structure document describes the internal format of a case, and the case document contains the content of the case. The case structure document describes the hierarchy and cardinality of the features that can exist in a case. There are two types of features, simple and complex. A simple feature is an attribute-value pair. A complex feature is a container for sub-features (simple and complex). Use of the complex feature type allows us to create cases with hierarchal structures. The feature definition contains the feature name and its type. Simple features can be further specified to be of a certain type, e.g. numeric, symbolic or taxonomic. This typing can be further constrained, e.g. a numeric feature's value can be constrained to lie within a specified range. Cases must follow their case specification exactly or they will be considered invalid. This ensures that bad case data or structure cannot corrupt the CBR system.

The syntax of a case is as follows: feature values are encased by a pair of XML tags with the feature name (an example case is shown in Fig. 3). This makes them easy to read and easy to translate into other formats since their tags do not contain attributes. This also serves to reduce the size of a case content document, which is important in a distributed CBR application.

The Generic CBML Module

We have developed a generic CBML module (in Java) that should be possible to plug into any CBR application. The main elements of this module are the case structure parser and the case content parser. The first reads the structure document and creates a structure definition in memory. The second parser uses this structure definition to validate and load cases from the case document into memory. We have created a number of interfaces that will enable a CBR system to accept the output from these parsers. However it is not enough for a CBR system to be able to load cases into memory. It also needs to know how to treat them and needs access to suitable similarity and adaptation rules to use these cases effectively. CBML currently specifies an optional `similaritymeasure` tag in each feature definition that allows a preferred similarity measure to be identified. This tag can be used to identify the Java class name of the similarity measure (that must implement our similarity measure interface) to be used with each feature. This is viewed as an extension to another attribute, the `discriminant` attribute, which tells the CBR system that a feature is for informational purposes only. A similar feature attribute could be used to identify adaptation functions. Developers can create packages of similarity measures for each domain in use. These functions are loaded as the case structure document parsed into memory.

By using the CBML interfaces and parsers it is possible to totally separate case description from the underlying CBR code. Once we change the structure document our case parser will expect the new structure. If we need new similarity or adaptation rules these can easily be imported into the system too. This separation of structure from the underlying application code is the main advantage of using CBML.

2 A CBML Application – The Personal Travel Assistant

In this section we describe a CBR application that uses CBML to represent its cases. This application is called the Personal Travel Assistant (PTA) and is described in greater detail in [1]. Our application follows on from a scenario described by the Foundation for Intelligent Physical Agents (FIPA) [3]. They describe an open marketplace inhabited by intelligent software agents where users come to negotiate trips with travel providers.

Our application makes use of this marketplace. At one end of this market is the user with a travel request and at the other, a number of service agents representing travel providers, guidance service providers and media agents. Brokering agents exist between the two, composing trips and offering value added services. The personal travel assistant we describe is a software agent that provides assistance to users in the planning and execution of such a trip. It also acts as the user's proxy in the marketplace. It takes travel requests from users and negotiates with broker agents for suitable travel solutions. In order to become efficient at its job, it should be able to learn the user's personal travel preferences and apply this knowledge when brokering deals.

The PTA should be able to act on behalf of many users. To be able to offer personalized service to every user it must build a user model or profile for every user. As well as the general user information, e.g. name, address, the profile contains a set of cases representing the user's previous interactions with the system. When a user initiates a new transaction, the travel assistant retrieves a list of cases representing relevant past transactions. This case base represents the user's preferences with respect to the current transaction. To illustrate this, we will now describe how the system uses this personalization knowledge in the evaluation and ranking of offers made by the brokers in response to a travel request.

A Travel-Offer Recommendation - a CBR Solution

The PTA makes a travel request to broker agents in the market who return travel offers. As these offers come from many sources, the PTA may receive a large number of offers with differing suitability. It therefore evaluates all offers and only recommends the most suitable of these.

The user's transaction history will contain all of their past travel requests and the offers that were made in response to these requests. It compiles a list of all offers in the transaction history that resulted from similar requests in the past. Each of these past offers will have a score associated with it that corresponds to the user's level of interest when that offer was made, i.e. accepted offers will have higher scores than rejected ones. Offers are viewed as cases and so, the case base of previous offers can be viewed as being representative of the user's preferences with respect to the current travel request. This case base is then loaded into a similarity retrieval system; in our application *case retrieval nets* [9] are used. Each offer that is received by the PTA is treated as a problem case and the most similar cases are retrieved from the user's previous offers. The recommendation scores of these offers are aggregated to determine the PTA's assessment of the current offer. When all offers have been received, the PTA sends the highest scoring offers to the user. As before, the user scores these cases (either implicitly, by booking one of them over the others, or explicitly by scoring them individually) and the PTA records these scores and adds these offers to the transaction history.

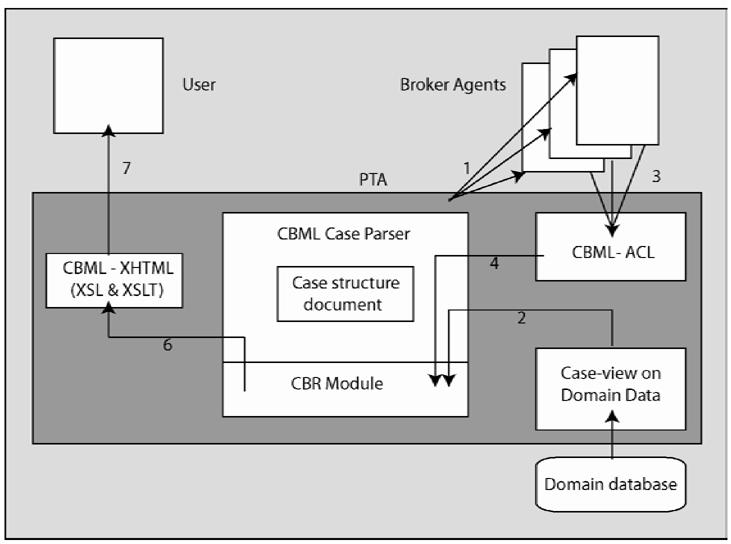

Fig. 2. Offer Recommendation. The PTA sends a travel request to the broker agents on behalf of the user {1}. At the same time the PTA retrieves a CBML case base of previous offers (that are close to satisfying the current request) from the domain database which is passed into the CBR module {2}. The CBR system is then prepared for the offer-recommendation process (the case base is loaded into a case retrieval net, etc). During this time the brokers will have begun to return a series of travel offers {3}. These offers are converted into CBML format and individually passed into the CBR system as problem cases {4}. When all offers have been received and scored, the highest ranking offers are converted into a presentation format (e.g. XHTML) using appropriate stylesheets and transformations {6} and presented to the user for approval {7}.

Case Sharing – Collaborative CBR

CBR is only useful in solving a problem if the case base offers good coverage for that problem. If we note that every user in the system has a different set of cases and some users will have similar travel preferences we could use a collaborative CBR [10] approach. If a user encounters a problem outside their previous experience we could borrow cases from similar users (the user's neighbours). This improves the problem coverage and should increase the probability of finding a good solution. We use this collaboration approach only when the user's own case base provides inadequate coverage. We estimate problem coverage by looking at the degree of similarity between the current problem case and the most similar cases from the user's own case base. A high level of similarity would indicate good coverage for the current problem.

3 How the Personal Travel Assistant uses CBML

Our PTA system consists of a number of modules as described in Fig. 2. This diagram shows the flow of information through the PTA during the offer-recommendation task. As mentioned in Section 3, user profiles contain a set of cases representing the user's previous transactions with the system. This information is actually stored in a database as a series of transactions referenced by user. When we need to compile a case base of suitable past offers we impose a case based view on the data [5] and retrieve the information in CBML format. An example case written in this format is shown in Fig. 3.

```xml
<case name="DUB-OSL #34">
  <features>
    <username>Coyle</username>
    <traveloffer>
      <origin>DUB</origin>
      <destination>OSL</destination>
      <departuretime>Mon, 2 Dec 2002 at 6:45 GMT</departuretime>
      <arrivaltime>Mon, 2 Dec 2002 at 12:00 CET</arrivaltime>
      <distance>1051</distance>
      <flighttime>255</flighttime>
      <hops>
        <numberofhops>2</numberofhops>
        <hop>
          <origin>DUB</origin>
          <destination>AMS</destination>
          <carrier>KLM</carrier>
          <departuretime>Mon, 2 Dec 2002 at 6:45 GMT</departuretime>
          <arrivaltime>Mon, 2 Dec 2002 at 9:20 CET</arrivaltime>
          <class>Coach</class>
        </hop>
        <hop>
          <origin>AMS</origin>
          <destination>OSL</destination>
          <carrier>KLM</carrier>
          <departuretime>Mon, 2 Dec 2002 at 10:10 CET</departuretime>
          <arrivaltime>Mon, 2 Dec 2002 at 12:00 CET</arrivaltime>
          <class>Coach</class>
        </hop>
      </hops>
    </traveloffer>
    <recommendation>5</recommendation>
  </features>
</case>
```

Fig. 3. An example offer-recommendation case (not all detail is shown). This case represents a travel offer made to user *Coyle* in response to a request for a flight from *Dublin* (DUB) to *Oslo* (OSL) leaving Dublin at *8:00am on the 2nd of December 2002*. The details of this offer are contained within the `traveloffer` tags. Finally the level of interest this user gave to this offer is shown between the `recommendation` tags; this offer was given a score of 5, the maximum, i.e. this user accepted and booked this offer. This case, its structure definition and other example documents are available online (details of this in Appendix A).

Our case based interpretation is represented by our case structure document. This document is stored as a file and the case structure is loaded from this document when the CBR module is initialised. An illustration of the separation between the case structure definition and the underlying application code in this application is as follows: at present our personal travel assistant deals only with flight data, but should this be extended later to include other options, e.g. train and bus solutions, we can easily extend our case structure to accommodate these. If necessary we could also replace or update the similarity measures for the changed features.

Case Translation

Another advantage of using CBML to represent our cases is the fact that it is XML-based. By following the XML specification we grant ourselves access to the XML toolset. This includes the Extensible Stylesheet Language (XSL) [14] and XML-Transformation (XSLT) [15]. These enable us to translate data from one XML format into another. Our PTA application uses these tools to translate our recommended cases for presentation to the user. This is made possible by the fact that most wireless terminals use WML [17] or XHTML [11] (an XML reformulation of HTML 4) to display content (our mini-PTA system currently uses WML). However using cases in this way leads us to scenarios where the user is interested in viewing features that are not used in the CBR process, e.g. the flight number of a travel offer. We include these features in our case view but we flag these features (using the `discriminant` attribute) so that they are excluded from the CBR process.

We can also develop translators to convert CBML data into other well-structured languages. This will be necessary if the PTA is to communicate with FIPA-compliant broker agents, who use FIPA's Agent Communication Language [2].

4 Conclusions and Future Work

Case Based Reasoning has proven to be a useful methodology for adding intelligence to the process of sales assistance. We have suggested that a standard integrated CBR *view* of an e-commerce information system is an important facility in this respect. We proposed that using a standard XML-based meta-language, CBML, is a key to enabling CBR applications to integrate with existing information systems. Using XML affords many advantages such as ease of transformation, flexibility of presentation and potential to create hybrid systems easily. The maturity of the XML schema definition has allowed us to develop CBML so that it can handle complex hierarchical case structures as well as the commonly used flat vector format.

We illustrated the potential of CBML by describing its use in a CBR-based travel advisor which we have developed. We described a personalisation task that is performed in this application and how CBR is used to solve it. We then outlined the advantages of using CBML in the context of this task. In the concluding paragraphs we outline some possibilities for further work that could be done in relation to both the PTA application and to the CBML specification.

In section 2 we described how similarity measures can be identified within the document structure. We intend to extend this idea to allow adaptation rules to be specified in the same manner. It may be appropriate to include other directives to the CBR system in the case structure file.

In a multi-user system such as the PTA application described in Section 3, users may rate the importance of features differently (e.g. users may have different sensitivity to the price of a travel offer). We are experimenting with a supplemental document (also written in CBML) that describes users' preferences with regard to individual features. The case structure document would contain default feature weights but these would be overruled by the values in the user's profile document. Hence, the CBR module will apply the same similarity retrieval mechanism for every user, except for the feature weighting, which would be done according to the values specified within this user preference document. This will allow a further level of personalisation to the application.

The original goal behind CBML development was to create an open standard to facilitate case based network computing, case base storage and possible interoperability with non-CBR systems [6]. Our first application involved pushing a subset of the case base from the server to the client browser side for further local refinement. This may have relevance in the wireless scenario described in section 3. Currently, most wireless devices are not computationally powerful and may consist only of a micro-browser (a small Internet browser). As these devices get more powerful it may be appropriate to distribute some of the PTA's responsibilities to the mini-PTA. It may be appropriate for the CBR module to be implemented on the client side. The PTA could then be responsible for data maintenance and distributing shared cases and the mini-PTA for carrying out personalization tasks. Those devices that will be capable of performing these tasks will not be homogeneous and it may be necessary to develop tailor-made CBR systems for different devices. By using a standard case representation language, the PTA could send case information to these heterogeneous platforms without information about the architecture of their native CBR system.

Appendix A: Online CBML Resources

There are additional resources on CBML available online which include:

- A technical description of CBML's capabilities
- The CBML schema definition documents which must be followed to ensure full compliance with the CBML specification.

- Several example cases, from different domains, and their case structure documents marked up in CBML format.
- Further instructions as to how to incorporate CBML into a CBR system including the interface definitions that were mentioned in this document.

These resources are available online at: www.cs.tcd.ie/Lorcan.Coyle/CBML

References

1. Coyle, Cunningham & Hayes (2002) A Case-Based Personal Travel Assistant for Elaborating User Requirements and Assessing Offers. Proceedings of the 6th European Conference, ECCBR 2002, Susan Craw, Alun Preece (eds.). LNAI Vol. 2416 pp. 505-518, Springer-Verlag
2. Foundation for Intelligent Physical Agents – ACL Message Structure Specification (Document No. XC00061E)
3. Foundation for Intelligent Physical Agents - Personal Travel Assistance Specification (Document No. XC00080). Carried forward from FIPA 1997 Specification 4 V1.0
4. Gardigen D., Watson I. (1998). A Web based Case-Based Reasoning System for HVAC Sales Support. Proceedings of British Expert Systems conference 1998
5. Hayes, C. & Cunningham, P. (1999) Shaping a CBR View with XML. Proceedings of the Third International Conference on Case-Based Reasoning, ICCBR'99, Seeon Monastery, Germany. LNCS Vol. 1650. Althoff, K.-D., Bergmann, R., Branting, L.K. (Eds.) Springer-Verlag Berlin/Heidelberg 1999, pp.468-481
6. Hayes, C., Cunningham, P., & Doyle, M.. Distributed CBR using XML. In Proceedings of the KI-98 Workshop on Intelligent Systems and Electronic Commerce, number LSA-98-03E. University of Kaiserslauten Computer Science Department, 1998
7. INRECA consortium (1994). Casuel: A Common Case Representation Language, available at http://wwwagr.informatik.uni-kl.de/~bergmann/casuel/CASUEL_toc2.04.fm.html
8. Kitano, H. & Shimazu, H. (1996) The Experience Sharing Architecture: A Case Study in Corporate-Wide Case-Based Software Quality Control. In Case-Based Reasoning: Experiences, Lessons & Future Directions. Leake, D.B. (Ed.) pp235-268. AAAI Press/ The MIT Press Menlo Park, Ca, US
9. Lenz, M., Auriol, E. & Manago, M. (1998) Diagnosis and Decision Support. In M. Bartsch-Sporl, H. D. B., and Wess, S., eds., Case-Based Reasoning Technology: From Foundations to Applications, LNAI Vol. 1400, Springer-Verlag
10. McGinty, L. & Smyth, B. (2001) Collaborative CBR for Real-World Route Planning. In Proceedings of the 2001 International Conference on Artificial Intelligence (IC-AI'2001) Las Vegas, Nevada
11. The W3C Recommendation on XHTML 1.0, the Extensible HyperText Markup Language, available online at: http://www.w3.org/TR/xhtml1
12. The W3C Recommendation on Extensible Markup Language (XML) 1.0 (Second Edition), available online at: http://www.w3.org/TR/REC-xml
13. The W3C formal description of XML Schema, available online at: http://www.w3.org/TR/xmlschema-formal/
14. The W3C Recommendation on Extensible Stylesheet Language (XSL) Version 1.0, available online at: http://www.w3.org/TR/xsl
15. The W3C Recommendation on XSL Transformations (XSLT) Version 1.0, available online at: http://www.w3.org/TR/xslt
16. Wilson, D. (2001) Case-Base Maintenance: The Husbandry of Experience. PhD dissertation, Indiana University, 2001
17. The Wireless Forum's WML Specification, available online at: http://www1.wapforum.org/tech/documents/WAP-238-WML-20010911-a.pdf