Conversational Case-Based Reasoning Support for Business Process Management

Barbara Weber
Department of Computer Science, Univ. of Innsbruck
Technikerstrasse 21a
6020 Innsbruck, Austria
barbara.weber@uibk.ac.at

Werner Wild
Evolution Consulting
Jahnstrasse 26
6020 Innsbruck, Austria
werner.wild@evolution.at

Abstract

Competitive pressures drive companies towards the implementation of process-aware information systems. In order to support a broad spectrum of business processes a process management system (PMS) must be flexible at run-time. This includes the support of ad-hoc deviations from the predefined process model, their memorization and reuse, and the support of changes of the underlying business process itself. To tackle this problem we developed the conversational case-based reasoning (CCBR) component CCBRTool. This paper presents CCBRTools's architecture and its major use cases and discusses some of its applications in the process management domain.

Introduction

Competitive pressures drive companies towards the implementation of process-aware information systems. These systems are characterized by the separation of process logic and application code. The business processes are explicitly described as process models and executed accordingly. However, in order to effectively support a company's business processes, process management systems (PMS) must be flexible at run-time and support deviations from the predefined flow of work (Reichert & Dadam 1998; Jørgensen 2004). When exceptional situations are handled in a too rigid manner, users are forced to bypass the system, resulting in limited traceability and information loss. Some PMS recognize the need for more flexible systems and support the definition of ad-hoc modifications for a single process instance. However, when the same or similar exceptions occur more than once, a new ad-hoc modification has to be defined each time. In addition, the knowledge about exceptions is not kept in the system and thus lost. Due to this, memorization and reuse of exceptional knowledge is highly desirable and should be supported by the PMS. When similar exceptions occur frequently they should no longer be dealt with in an ad-hoc way, but should be covered by updating the process model itself. Frequent similar exceptions indicate that the process model does not adequately reflect the real-world business process. The process engineer should then be supported by the PMS to utilize the collected knowledge for improving the process model at hand.

To tackle this problem we developed the conversational case-based reasoning (CCBR) component CCBRTool. This paper presents CCBRTools's architecture and its major use cases. Further, we briefly describe CBRFlow (Weber, Wild, & Breu 2004), DLE (Dynamic Logic Engine) and ADEPT (Reichert & Dadam 1998), three applications using CCBRTool in the context of process management. CBRFLOW and DLE both help the process user in exceptional situations by invoking CCBRTool to collect relevant data and to support its reuse (Weber, Wild, & Breu 2004). In addition, the adaptive PMS ADEPT uses CCBRTool to derive process improvements and to continuously adapt the process model to the living process (Weber *et al.* 2005b; Rinderle *et al.* 2005).

Overview of CCBRTool

CCBRTool is a CCBR system developed for the process management domain. However, it can be easily adapted to other domains. This section presents CCBRTool's architecture as well as its major use cases.

Architecture

CCBRTool is implemented as an Eclipse Plugin which can also be run as stand-alone application (Eclipse Rich Client Platform). As illustrated in Fig. 1, CCBRTool usually is integrated into a user portal and can be invoked from third party-applications via XML-RPC (XML-RPC 2005) or SOAP (W3C 2005). CCBRTool is thus an interactive Web Service fitting nicely into service-oriented architectures (SOAs).

The third party application must implement a small CCBRTool adapter which handles the communication with the CCBRTool component. An XML request is sent to the user portal (CCBRTool) and, after the interactions with the user, a response message is sent back to the adapter. The adapter further invokes the functions to execute when receiving the response, i.e., calling the corresponding third-party application's functions.

XML Request: The XML request consists of a case-base, which contains a set of predefined actions as well as a set of cases (cf. Fig. 2). As an interactive Web Service CCBR-

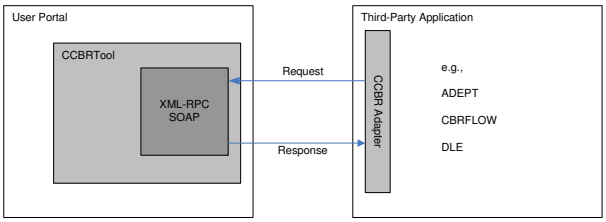

Figure 1: High-level Architecture

Tool receives the XML request and presents its GUI which is initialized according to the case-base in the request.

Request

```
<REQUEST>
      <CASEBASE ID="1">
            <ACTIONS>
                  <!-- This section has to be specified for each
                        application to support automated actions -->
            </ACTIONS>
            <CASES>
                  ...
            </CASES>
      </CASEBASE>
</REQUEST>
```

Response

```
<RESPONSE>
      <CASEBASE ID="1">
            <CASE>
                  ...
            </CASE>
      </CASEBASE>
      <PERFORM_ACTION/>
</RESPONSE>
```

Figure 2: XML Request and Response

Predefined actions are application dependent, they have to be specified for each invocation of CCBRTool and are transmitted in the request. Thereby only actions which are supported in the current context of the third-party application and for which the user is authorized should be included in the request. In addition, CCBRTool allows for *manual actions* to document actions the user performs when bypassing the system (e.g., to deal with exceptions).

XML Response: The XML response consists of the selected/modified case and a meta-action, e.g., PERFORM_ACTION or DELETE_CASE_ACTION (cf. Fig. 2).

Case Representation: Within CCBRTool a case consists of a textual problem description, a set of question-answer pairs and an action to perform (i.e., a manual action or an automated action from the list of predefined actions). In addition, each case keeps its history (cf. Fig. 3) to foster traceability.

Major Use Cases

The major use cases supported by CCBRTool are:

- Retrieve Case
- Show Case
- Add New Case
- Apply Case
- Evaluate Case
- Show Feedback

```
<CASE ID="71">
```

Problem

```
…
<NAME>Dangerous Goods</NAME>
<DESCRIPTION>Shipment must be labeled when articles classified as
      'dangerous goods' are contained in the shipment. </DESCRIPTION>
```

Question-Answer Pairs

```
<QAPAIRS>
      <QAPAIR>
            <QUESTION ID="110">Dangerous goods in shipment?</QUESTION>
            <ANSWER ID="129">Yes</ANSWER>
      </QAPAIR>
      <QAPAIR>
            <QUESTION ID="109">Do dangerous goods exceed legal
                  limits?</QUESTION>
            <ANSWER ID="128">Yes</ANSWER>
      </QAPAIR>
</QAPAIRS>
```

Action

```
<ACTION_TO_PERFORM ID="MANUAL">
      <SHORT_DESCRIPTION>Label as dangerous goods, declare chemicals
      </SHORT_DESCRIPTION>
      <DESCRIPTION></DESCRIPTION>
</ACTION_TO_PERFORM>
```

Case History

```
<HISTORY>
      <HISTORY_ENTRY TYPE="CREATE" ID="11279" >
            <TIMESTAMP>2005-07-10 12:27:06</TIMESTAMP>
            <USER>Tammy Traveller</USER>
      </HISTORY_ENTRY>
      <HISTORY_ENTRY TYPE="EVALUATE" ID="12346">
            <TIMESTAMP>2005-07-11 17:44:00</TIMESTAMP>
            <USER>Tammy Traveller</USER>
            <FEEDBACK>
                  <GRADE>HIGHLY POSITIVE</GRADE>
                  <COMMENT>The shipment of small propane gas containers
                        to Switzerland worked fine by reusing the
                        case</COMMENT>
            </FEEDBACK>
      </HISTORY_ENTRY>
      <HISTORY_ENTRY TYPE="EVALUATE" ID="12935">
            <TIMESTAMP>2005-07-15 11:51:00</TIMESTAMP>
            <USER>John Miller</USER>
            <FEEDBACK>
                  <GRADE>HIGHLY POSITIVE</GRADE>
                  <COMMENT>I applied the case successfully to the
                        shipment of hairspray within the EU.</COMMENT>
            </FEEDBACK>
      </HISTORY_ENTRY>
</HISTORY>
```

```
</CASE>
```

Figure 3: Case Representation

- Edit Case
- Delete Case
- Adapt Case

Retrieve Case: After opening CCBRTool the graphical user interface for retrieving similar cases pops up (cf. Fig 4). For case retrieval the standard CCBR problem-solving process as described in (Aha, Breslow, & Muñoz-Avila 2001) has been slightly adapted. The system presents the user with the set of unanswered questions of all cases in the case-base, ranked by their occurrence frequency. The user can answer any of the displayed questions in arbitrary order and/or filter the case-base by applying a full-text search. The system then calculates the similarity of all cases in the filtered case-base and displays a list of cases ranked by their similarity. Cases which do not match the filter criteria are omitted. Similarity is calculated by dividing the number of correctly answered questions by the total number of questions in the case. The questions are then re-ranked based on the retrieved cases and displayed. The user can continue to answer questions until a suitable case is found or abort the dialog when the system fails to find a similar enough case.

Show Case: A selected case can be displayed in detail, the case (problem description, question-answer pairs and actions) and its full history is shown (cf. Fig. 5).

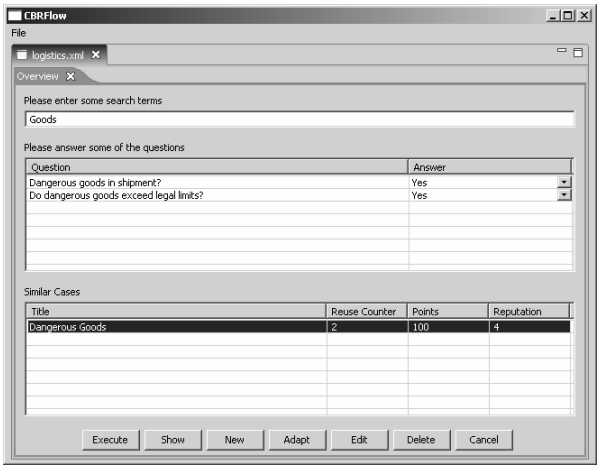

Figure 4: Case Retrieval

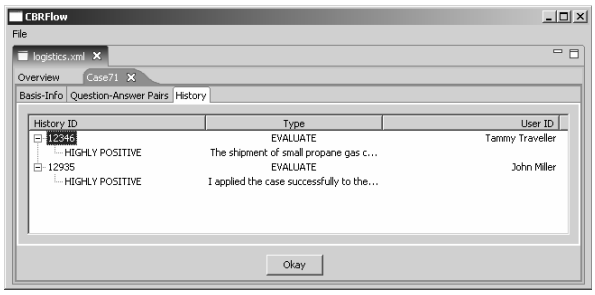

Figure 5: Case History

Add Case: When CCBRTool fails to retrieve a similar enough case, a new case can be added to the system (cf. Fig. 6). For this, the user enters a textual problem description, a set of question-answer pairs and a solution part (i.e., action). Additional administrative attributes like creation date are added by the system.

In order to avoid multiple similar question-answer pairs with the same semantics are entered more than once, users are encouraged to reuse question-answer pairs from existing cases. Question-answer pairs can be entered by either selecting already existing question from a list or, when the system has no suitable question yet, by defining a new question and giving the appropriate answer. CCBRTool intentionally allows the user to specify question-answer pairs using natural language and does not use controlled vocabularies. A given controlled vocabulary only allows to express those things the vocabulary is designed to capture, which is not feasible for handling unexpected exceptions. To prevent uncontrolled growth of the case-base the process engineer should review and consolidate the case-base from time to time.

Actions can be either manual or automated. On the one hand, manual actions simply describe the solution in a textual way, their execution does not include any system supported actions. Manual actions are especially useful when tasks have to be performed which are not (yet) directly sup-

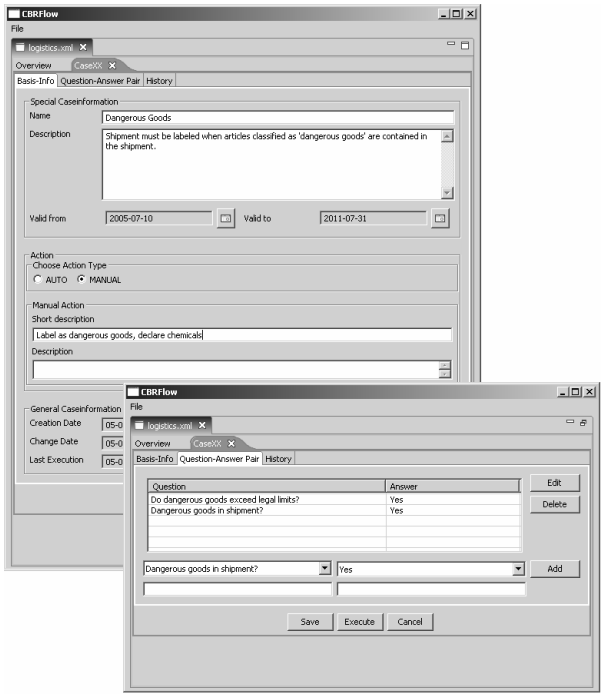

Figure 6: Adding a New Case

ported by the third-party system (e.g., to deal with an exception). On the other hand, automated actions are performed by the third-party application (e.g., skipping an activity in a business process). The application dependent list of actions to select from is sent to CCBRTool in the XML request.

Apply Case: When a similar enough case has been found, or a new case has been added to the case-base, its solution can be applied. For this, a response message including the action to be performed is returned by CCBRTool. In case of a manual action a textual description of the solution is presented to the user who can then resolve the problem by performing the described manual steps. For automated actions the requesting third-party system automatically performs the action specified in the response message.

Evaluate Case: Whenever a case is applied, a work item is generated to evaluate the case's performance later on. As cases are added to the system by end users and not by a process engineer, evaluation mechanisms are needed to keep the quality of the cases in the case-base sufficiently high. Although this can not prevent adding low quality cases to the case-base, it can at least ensure that such cases are not reused by other users. Negative feedback results in an immediate notification of the process engineer, who can then repair the case or deactivate it to prevent its further reuse.

As illustrated in Fig. 7 the user can rate the performance of the respective case either with highly positive (2), positive (1), neutral (0), negative (-1) or highly negative (-2), and may optionally enter textual comments too.

Show Feedback: The detailed feedback of a case can be displayed by right-clicking the reputation score in the display list of cases (cf. Fig. 4). The reputation score in com-

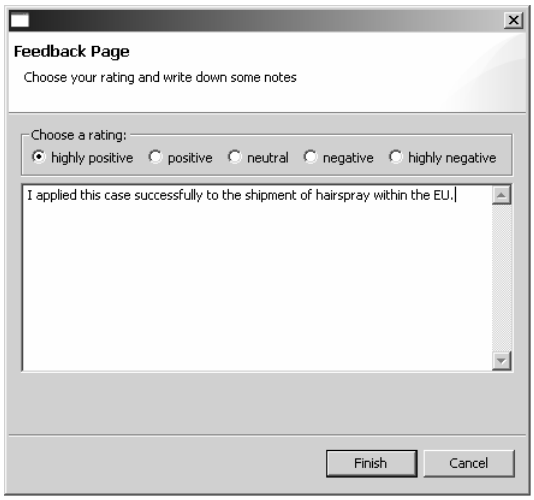

Figure 7: Evaluate Case

bination with the reuse counter indicates how successfully a case has been applied in the past. The reputation score is calculated as the sum of the feedback scores (cf. Fig. 8). In addition to the overall reputation score, plus some simple statistics a table with the scores for the past six month, the past month and the past week is displayed. Upon request the textual comments can be displayed as well.

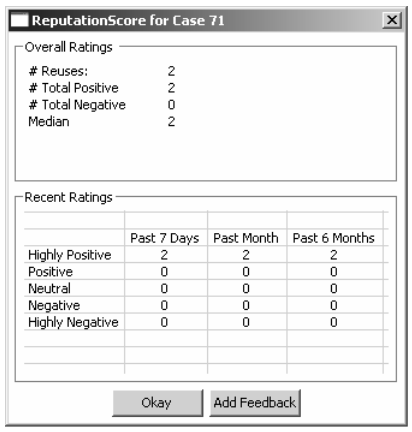

Figure 8: Show Feedback

Edit Case: Editing a case is only allowed for administrators to correct typos or other mistakes. In order to ensure traceability a history entry is written when a case is modified.

Delete Case: Cases which have not proven successful or have become obsolete over time can be deactivated by the system's administrator. Deactivated cases can no longer be applied to solve new problems; however, they are not removed from the case-base to foster traceability and learning from failure.

Adapt Case: When editing a case the original case is modified. In contrast, when adapting a case a copy of the original case is made which can then be modified. This functionality can be used to quickly add cases to the case-base, when they are similar to already existing ones (e.g., same problem description and solution, but different question-answer pairs).

Applications of CCBRTool

This section presents the three systems CBRFlow, DLE and ADEPT, our current applications of CCBRTool in the process management domain.

CBRFlow

CBRFlow (Weber, Wild, & Breu 2004) is a research prototype of a PMS developed at the University of Innsbruck. Within CBRFlow, CCBRTool is used to memorize knowledge about exceptions and to allow process users to deviate from the predefined process model. CBRFlow supports the continuous evolution of the process model by updating the business rules in the process model when necessary. When the process knowledge encoded in cases is reused frequently, it should be extracted by the process engineer and be described directly in the process model to allow for full automation.

DLE

DLE (Dynamic Logic Engine) is a user-friendly programming environment for domain experts (e.g., in the logistics industry), developed by the Austrian software company VISION-FLOW. Using DLE, business logic can be extracted from application systems (e.g., ERP packages) into an easy to maintain environment. CCBR enables the interactive selection of predefined executable components in unforeseen situations. In DLE, like in CBRFlow, CCBRTool is used for the memorization and the reuse of knowledge about exceptions. Automated actions as well as manual actions are supported.

ADEPT

ADEPT (Reichert & Dadam 1998) is an adaptive PMS which has been developed at the University of Ulm. It supports ad-hoc modifications (i.e., process instance changes) as well as changes of the underlying business process (i.e., process type changes).

Currently we work on the full integration of CCBRTool into ADEPT to support the whole process lifecycle in an integrated way (cf. Fig 9) (Weber *et al.* 2005a; 2005b; Rinderle *et al.* 2005).

The following describes the process life cycle as supported by ADEPT. At buildtime an initial computerized representation of a company's business processes is created either by business process analysis or by applying process mining techniques (i.e., by observing process and task executions) (1). At run-time new process instances are created from these predefined process models (2). Process instances are then executed according to the process model

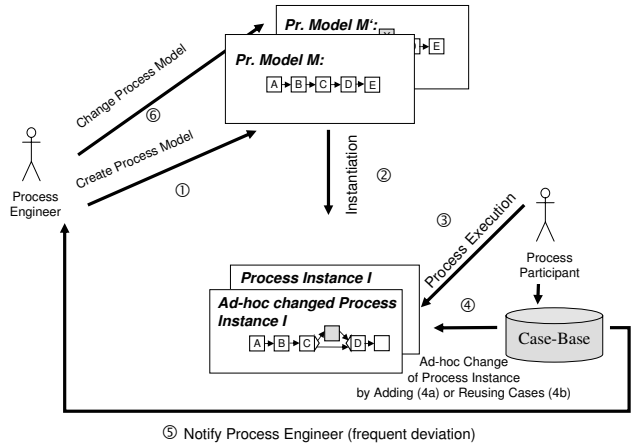

Figure 9: Integrated Process Lifecycle Support

they were derived from and activities are allocated to authorized process participants to perform the respective tasks (3). However, when deviations from the predefined model become necessary at the process instance level (e.g., due to exceptions), process participants must be able to deviate from the model. They can either specify a new ad-hoc deviation and document the reasons for the changes in a case-base (4 a), or reuse a previously specified ad-hoc modification from the case-base (4 b). The PMS monitors how often a particular model is instantiated and how frequently cases are reused. When a particular ad-hoc modification is reused frequently, the process engineer is notified that a process type change should be performed (5). The process engineer can then evolve the process model, and, as far as possible, migrate running instances, with help from ADEPT, to the new model version (6). The associated case-base has to be migrated as well; cases which get directly integrated into the new process model version are no longer needed and can be deactivated in the case-base. All other cases are still kept active in the case-base.

Related Work

Several other approaches combine process management and case-based reasoning. CBR has been used to support process modeling (Kim, Suh, & Lee 2002; Madhusudan & Zhao 2003) and ad-hoc planning (Fremann, Maximini, & Sauer 2005), it has been applied to the configuration of complex core processes (Wargitsch 1998) and to the handling of exceptions (Luo *et al.* 2000). However, all of these approaches apply traditional CBR, to our best knowledge there are no other applications of CCBR to process management.

AI planning, especially mixed-initiative case-based planning (e.g., NaCoDAE/HTN (Muñoz-Avila *et al.* 1999), MI-CBP (Veloso, Mulvehill, & Cox 1997), SiN (Muñoz-Avila *et al.* 2001) and HICAP (Muñoz-Avila *et al.* 2002)) can be seen as complementary to our approach, as we primarily focus on the execution of processes and not on modeling or planning. In contrast to AI planning, a process model is instantiated for a large number of process instances, i.e., many process instance are started according to the same process model (template).

CCBRTool esentially adapts the standard CCBR problem-solving process as described in (Aha, Breslow, & Muñoz-Avila 2001). Taxonomic (Gupta, Aha, & Sandhu 2002; Gupta 2001) or causal CCBR (Gupta & Aha 2002) are extensions of standard CCBR, but are currently not used in CCBRTool.

Conclusion

CCBRTool is a CCBR system currently tailored to process management but can easily be adapted to other domains as well. Within CBRFlow and DLE CCBRTool is used to support the user to deviate from the predefined process model, to describe her deviations and to reuse the knowledge in similar future situations. The process engineer is not supported in abstracting the knowledge encoded in cases and in improving the process model. However, in ADEPT (with CCBRTool) the process engineer will be notified when case usage exceeds a predefined threshold value and will be assisted in migrating both the process model and the corresponding case-base.

Our ongoing research focuses on the integration of adaptive process management technology and CCBR. We currently work on the implementation of a prototype which combines the methods and concepts provided by ADEPT and CCBRTool. We further plan a thorough evaluation of the integrated system in different application settings, including healthcare processes and emergent workflows (e.g., in the automotive sector).

To prevent unauthorized changes to the process model we are also working on the implementation of an access control model (as a Web Service) and its integration into the adaptive PMS ADEPT and CBRFlow.

References

Aha, D. W.; Breslow, L.; and Muñoz-Avila, H. 2001. Conversational case-based reasoning. *Applied Intelligence* 14(1):9–32.

Fremann, A.; Maximini, R.; and Sauer, T. 2005. Towards collaborative agent-based knowledge support for agile projects. In *WM2005: Professional Knowledge Management Experiences and Visions*, 383–388.

Gupta, K., and Aha, D. W. 2002. Causal query elaboration in conversational case-based reasoning. In *Proceedings of the Fifteenth Conference of the Florida AI Research Society*.

Gupta, K.; Aha, D. W.; and Sandhu, N. 2002. Exploiting taxonomic and causal relations in conversational case retrieval. In *Proceedings of the 6th European Conference, ECCBR02*.

Gupta, K. 2001. Taxonomic case-based reasoning. In *Proceedings of the Fourth International Conference of Case-Based Reasoning, ICCBR05*, 219–233.

Jørgensen, H. D. 2004. *Interactive Process Models.* Ph.D.

Dissertation, Norwegian University of Science and Technology, Trondheim, Norway.

Kim, J.; Suh, W.; and Lee, H. 2002. Document-based workflow modeling: a case-based reasoning approach. *Expert Systems with Applications* 23(2):77–93.

Luo, Z.; Sheth, A.; Kochut, K.; and Miller, J. 2000. Exception handling in workflow systems. *Applied Intelligence* 13(2):125–147.

Madhusudan, T., and Zhao, J. 2003. A case-based framework for workflow model management. In *Proc. BPM'03*, 354–369.

Muñoz-Avila, H.; McFarlane, D.; Aha, D.; Ballas, J.; Breslow, L.; and Nau, D. 1999. Using guidelines to constrain interactive case-based htn planning. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 288–302.

Muñoz-Avila, H.; Aha, D.; Nau, D.; Breslow, L.; Weber, R.; and Yamal, F. 2001. Sin: Integrating case-based reasoning with task decomposition. In *Proc. IJCAI-2001*, 99–104.

Muñoz-Avila, H.; Gupta, K.; Aha, D.; and Nau, D. 2002. *Knowledge Management and Organizational Memories.* Kluwer Academic Publishers. chapter Knowledge Based Project Planning.

Reichert, M., and Dadam, P. 1998. ADEPT$_{flex}$ - supporting dynamic changes of workflows without losing control. *JIIS* 10(2):93–129.

Rinderle, S.; Weber, B.; Reichert, M.; and Wild, W. 2005. Integrating process learning and process evolution - a semantics based approach. In *Int. Conference on Business Process Management 2005*.

Veloso, M.; Mulvehill, A.; and Cox, M. 1997. Rationale-supported mixed-initiative case-based planning. In *Proceedings of the Ninth conference on Innovative Applications of Artificial Intelligence*, 1072–1077.

W3C. 2005. http://www.w3.org/tr/soap/.

Wargitsch, C. 1998. *Ein Beitrag zur Integration von Workflow- und Wissensmanagement unter besonderer Berücksichtigung komplexer Geschäftsprozesse.* Ph.D. Dissertation, Erlangen.

Weber, B.; Reichert, M.; Rinderle, S.; and Wild, W. 2005a. Towards a framework for the agile mining of business processes. In *Proc. of Int'l BPI workshop*.

Weber, B.; Rinderle, S.; Wild, W.; and Reichert, M. 2005b. CCBR–driven business process evolution. In *Int. Conference on Case-Based Reasoning (ICCBR'05)*.

Weber, B.; Wild, W.; and Breu, R. 2004. CBRFlow: Enabling adaptive workflow management through conversational case-based reasoning. In *Proc. Eurpean Conf. on Case–based Reasoning (ECCBR'04)*, 434–448.

XML-RPC. 2005. http://www.rml-rpc.com.