IST 597 A – Web and XML Databases

Project Report

Dynamic Workflow Generation for Web based Customer Support Application

Karthikeyan Umapathy
IST PhD Student
[kxu110@psu.edu]

Abstract

In recent research in workflows there is particular emphasis on workflows focusing on how individual input objects should be processed within an organization. Such workflows arise commonly in practice, including insurance claims processing, and many electronic commerce applications, and in the area of web based customer support. Applications like web based customer support are conversation centric. Conversations are the core part of the business process here. In this project I provide a methodology to convert these conversations to workflows, drawing on underlying speech acts and workflow patterns. I describe a mapping between speech acts and workflow patterns which can be used to dynamically specify emerging workflows, which can still meet properties such as correctness and verifiability.

1.0 Introduction

Systems that support and augment work and workflow in organizations must be capable of representing a rational, clear, work process, while still supporting the messy nature of real work. Part of this task is supporting the conversations that coordinate people (or automated systems) in shared activity and allowing them to react collectively to problems as they arise [Kammer 1999]. Human conversation is an important part of business processes. Workflows formalize and support such goal-directed conversations among organizational participants. Existing work on workflow structures heavily favors formalizing these conversations, often ease of expression, understanding and manipulation. This can lead to rejecting workflows enacted with these formalisms [Karsten 2000].

The objective of this project, therefore, is to develop a mapping to bridge the gap between the formalized structures required for workflow enactment, and the everyday language mechanisms used by human developers and users to describe and conduct business conversations. To achieve the goal, I draw on workflow patterns [van der Aalst 2000] as coarse building blocks of workflow structures, speech act [Searle 1969, Bach and Harnish1979] based messages and Petri Net models (PNML [Michael Weber 2002]).

I am using Web based customer support system as my motivating example and as an application area to implement my proposed methodology. All the information required for web based customer support application will be stored in various XML files. Information required for mapping speech act messages to workflow pattern will also be stored in various XML files. Output of the project, which is workflow, will be PNML file, which is XML formatted language.

Rest of the project paper is as follows, in section 2 I will be giving some background information, in section 3 I will explain my mapping methodology, in section 4 I will explain my implementation and in section 5 I give conclusions and future directions.

2.0 Background

2.1 Web based customer support

A web based customer support systems lets user to interact with customer support application through web browser. Using the Web as the mechanism for customer service representative to manage customer support makes a lot of sense, because you don't have client-side software to deal with, user interface and functional changes are instant for all users, and customer service records can be updated from the field if required. In general the an web based customer support application allows customers to register themselves, as well as update and check the status of their own support cases. There is also a product-specific help desk created automatically as support cases are solved. There is also a support-staff Web application that creates, logs and manages support cases; includes case search capabilities, case routing and priority functions; and supports dispatcher status availability.

Web is ideally suited for customer support, because it can provide instant and readily accessible support. Companies have been fast in adapting E-business, which had slow rate

of profitability; companies have been slower in adapting E-support, which will have immediate effect with their business. Companies, that has 24/7 customers needs 24/7 customer support services. Web provides the most cost effective and fastest way of provide answer to customers. Web based customer support can give immediate gratification to customers.

2.2 Workflows

The Workflow Management Coalition defines workflow as: "The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [Rob Allen 2000]. Workflows are the systems used to monitor and control business process. Purpose of a workflow system is to capture business process and coordinate inter and intra activities of enterprise involving both automated and non-automated tasks. The workflow system should allow appropriate personnel to define the process which is to be controlled and monitored, and coordinate the execution of the defined process. The workflow system should also allow the visualization of the defined processes; and allow the monitoring of the current state of the system and of the dynamics of the system. The rationale for the workflow system is to reduce the time taken to carry out a process, improve coordination and control over process and provide accountability.

2.2.1 Workflow patterns

Differences in features supported by the various contemporary commercial workflow management systems, point to different insights of suitability and different levels of expressive power. Many of the more complex requirements identified, recur quite frequently in the analysis phases of workflow projects, however their implementation is uncertain in current products. Requirements for workflow languages are indicated through workflow patterns. In this context, patterns address business requirements in an imperative workflow style expression, but are removed from specific workflow languages. Currently there are about 20 workflow patterns available.

2.3 Speech Acts

Speech Act Theory introduced by Austin [Austin 1962], and further developed and formalized by Searle [Searle 1969]. The main characteristic of the theory is that it considers

the use of language to be a form of rule-governed behavior. Uttering a sentence is the performance of a purposeful act, a so-called speech act. Making a statement may be the paradigmatic use of language, but there are all sorts of other things we can do with words. We can make requests, ask questions, give orders, make promises, give thanks, offer apologies, and so on. Moreover, almost any speech act is really the performance of several acts at once, distinguished by different aspects of the speaker's intention: there is the act of saying something, what one does in saying it, such as requesting or promising, and how one is trying to affect one's audience. In speech act theory speaker's intention is called as illocutionary force.

In general, speech acts are acts of communication. To communicate is to express a certain attitude (illocutionary force), and the type of speech act being performed corresponds to the type of attitude being expressed. For example, a statement expresses a belief, a request expresses a desire, and an apology expresses regret. As an act of communication, a speech act succeeds if the audience identifies, in accordance with the speaker's intention, the attitude being expressed.

2.4 Petri net markup language (PNML)

The Petri Net Markup Language (PNML) is an XML-based interchange format for Petri nets [Michael Weber 2002]. In order to support different versions of Petri nets and, in particular, future versions of Petri nets, PNML allows the definition of Petri net types. Due to this flexibility, PNML is a starting point for a standard interchange format for Petri nets. One of the most required features of Petri net tools are functions for exporting Petri nets to other tools and for importing nets from other tools. The problem with this apparently simple and purely technical feature is the multitude of different Petri net types and the multitude of different tools and file formats for these different net types.

PNML would allow Petri Net tool users in geographically distributed locations to take advantage of newly developed facilities on other tools, for example, for analysis, simulation or implementation. The Petri net community would be able to exchange Petri net models that are of mutual interest, perhaps for teaching a course, or in a global development project where teams in different countries exchange design information. It would allow a library of Petri net models to be created that could be accessed worldwide via the Internet and edited,

simulated and analyzed on different tools. This idea can be extended to the transfer of analysis results.

3.0 Conversations to workflow

Conversations are important part of business process. Workflow systems should be support such a goal directed conversations. Workflows formed from conversations are adhoc workflow. Adhoc workflows typically involve human coordination, collaboration or co-decision. Thus the ordering and coordination of tasks (activities) in an ad-hoc workflow are not automated but are instead controlled by humans. Process of converting conversations to workflow should be automated and has to be dynamic, emergent and adaptable.

3.1 Mapping speech acts to workflows

Methodology to convert speech acts to workflows is very simple. First the messages are converted into speech act based message. Then illocutionary force of speech act is mapped to workflow pattern. As conversation proceeds the workflow patterns are combined to form workflow. Figure 1 shows my mapping methodology. There are two most crucial parts with this methodology, they are extracting illocutionary force from messages, and mapping illocutionary force to workflow patterns. Issues that require addressing while mapping speech acts to workflow patterns include:

- Mapping the illocutionary force of speech acts to workflow patterns,
- Degrees of freedom between speech acts and workflow patterns, and
- Prediction of workflow pattern to be followed a speech act.

To address these issues, my research approach is incremental development of the workflow, with a one-to-many mapping between speech acts and workflow patterns. The mapping provides constraints, which can be used to support the construction of an emergent ad-hoc workflow.

Operationalizing the approach requires that each workflow pattern can be divided into head, body, and tail. Figure 2 shows the head, body and tail part for the patterns ‘Sequence’ and ‘Parallel Split’. The illocutionary force of the speech act can then be mapped to the head, body or tail of the workflow pattern. Therefore, a single illocutionary force can be mapped to N possible workflow patterns.

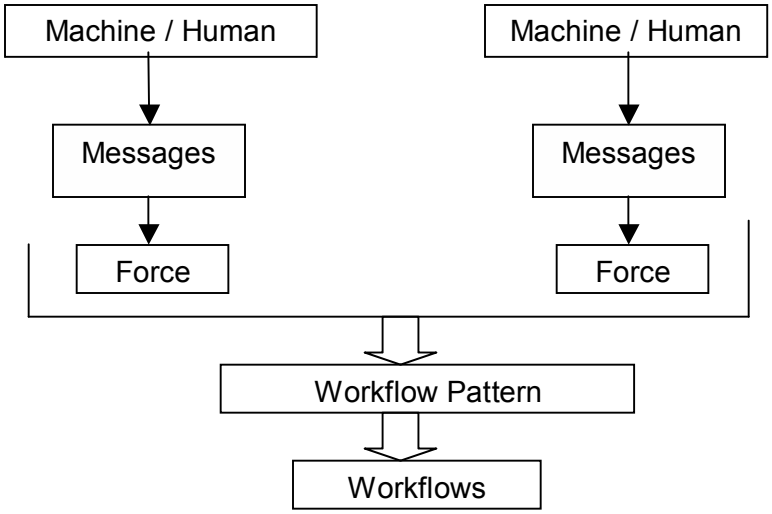

Figure 1: Conversations to workflows

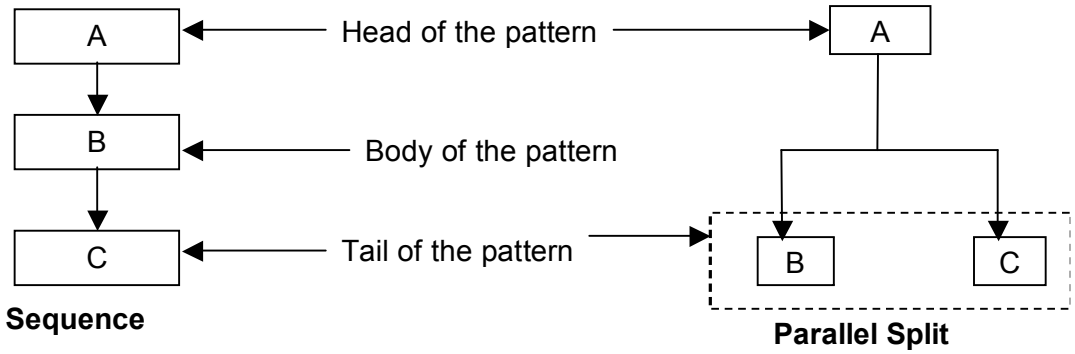

Figure 2: Head, body and tail of workflow pattern

A knowledge base can store this mapping. A walkthrough of how this knowledge-base can be used is described below. For the speech act is from the initiator of conversation, the illocutionary force is checked for head position of the workflow pattern. As the conversation proceeds; the next illocutionary force is checked for tail or body of the workflow patterns. A concatenation of speech acts in this manner can, then, predict the emerging workflow pattern or subset of patterns. Clearly developing the knowledge-base mapping speech acts to workflow patterns is critical.

4.0 Customer support application

Web based customer support application is used here as motivating example and implementation platform to use proposed methodology. In this application, customer asks questions to support system, system will return appropriate responses. Customer support application will have knowledge base, based on which it responses to customer queries. Customer support application will have different responses for different types of questions. Based on the response from the system, customer will provide answers or make choices.

From each message of the customer and of the support system, illocutionary force of the message is extracted and stored. Question from the customer and response of the support system is called transaction. In transaction there should be at least one message from customer and one message from support system, there is no limitation with number messages.

From transaction, list of messages will be retrieved, from the messages illocutionary force will be retrieved, then illocutionary force will be mapped to workflow pattern. Workflow pattern information will be retrieved from workflow pattern knowledge base. As workflow pattern is mapped, it is PNML file structure will be add to single PNML file, which will be the workflow file.

4.1 FAQ based customer support

Due to limited time, I reduced the Web based customer support application to simple FAQ based answering system. I have used Dell's FAQ's for this project. In this customer will enter into the system, if they have log in permission, then customer profile will be shown to the customer. In the profile, list of the products purchased by the customer will be also shown. Customer can select the product for which s/he needs for support. After the product is selected, customer will be provided with problem areas. After customer selects the appropriate product, list of frequently asked question is provided to customer. When customer selects any question, support system provides the answer to the question. While conversation between customer and support system goes on, customer support application shows the entire conversation of the day that took place between customer and support systems.

I have used web client for interaction between customer and support system and convert those interaction to messages and message pair to transactions. I have developed another separate standalone application to convert these messages to workflows.

4.2 Dynamic workflow generation

This dynamic workflow generation application is a standalone application. I assume that this application is run on end of the day. It can be triggered automatically or manually started. This application retrieves messages of the current day generated by the Web client, this easier because message ID has date and time information when message was generated. Even ID's of transaction, pattern, and customer history will have date and time information when they were created.

The messages generated for current date is retrieved and illocutionary force is extracted. In this I am using 10 different illocutionary forces; they are Assentive, Assertive, Conformative, Deny, Disputative, Informative, Question, Requestive, Suggestive, and Suppositive. For each force certain words are checked whether it is present in the message, if they are present, then count for that force is increased. Which ever force count is higher, that force is assigned to the message and it is updated to message XML file. If no force is found, then by default, customer message is assigned question as force, and for system response as suggestive force.

After forces are assigned for the messages, then transaction that took place for the current date is retrieved. From the transaction, message pair information is retrieved and their forces are retrieved from message XML file. This force pair is compare against force pair and workflow pattern information in the pattern XML file. In this project I am using only sequence workflow pattern only. FAQ's also follows sequence workflow pattern. Therefore pattern XML file has force pairs for sequence workflow pattern. Therefore when force pairs is compared with pattern XML files, sequence pattern will be returned as match, then workflow pattern information is updated to transaction XML file.

After workflow patterns assigned for the transactions, then workflow file is generated for each customer who had interaction with support system. From the transaction XML file, pattern information of each customer who had interacted with support system. Then PNML file structure of pattern is added to PNML workflow file for the customer. In this project, only

sequence workflow pattern is used, therefore only sequence of PNML format is used. PNML structure of the sequence workflow pattern is given in the appendix. After workflow file is created, then workflow information is updated to workflow history XML file.

Algorithm for the force extraction of messages, pattern matching for transaction, workflow file generation and workflow history update is given in the appendix.

5.0 Conclusion

I have developed Web based customer support system based on FAQ's of Dell. Even though this web client does not look like realistic, but still this is functional system and it is a scalable. This web client stores the messages interacted in the message XML file and message pairs as transactions in the transaction XML file. Stand alone workflow application retrieves information from message XML file and extracts force information for the messages and updates the files, retrieves transaction information from the transaction file and matches the workflow pattern, and updates the pattern information in the transaction XML file, then using the pattern information from the transaction file, creates workflow file for each customer who had interacted with support system, and then updates the workflow file information to the workflow history XML file. Therefore in this project I have developed functional web based customer support system, and developed an application to convert the messages interaction to workflow. When this standalone application linked with web based customer support system it becomes a dynamic system which will help customer to solve the problems as well as act as a system to convert interaction to workflow.

5.1 Future work

Web based customer support system should be made more realistic. Customer should be allowed to ask questions in natural language and support system should process and response to the queries. Need to use linguistics theories, hermeneutics theory and ontologies to extract illocutionary force from message content. Develop knowledge base of illocutionary force to workflow pattern mapping. Provide graphical views of the workflow file. Link the standalone workflow generation application with web based customer support system, so the workflows are generate dynamically.

Appendix

Algorithm for extracting from messages

Load Message.XML to Doc XMLDocument variable
Get all 'msg' elements and store it in 'msg' variable
Get all 'content' elements and store it in 'content' variable
For i = number of 'msg' elements -1 to 0 step -1
    If (date in 'msgID' equal to today's date)
        Force = call forceIdentifier (content[i])
        Create XMLElement of illForce with Force as element value
        Add XMLElement to Node at "/Message/Msg[@msgId='msgID']"
    End If
End For
Save Doc

forceIdentifier function (strContent)

Following are the list of the force and list of their words whose presence in message increases that particular force.

assentive (accept, agree, assent, concur)
assertive (affirm, allege, assert, aver, avow, claim, declare, deny, indicate, maintain, propound, say, state, submit)
confirmative (appraise, assess, bear witness, certify, conclude, confirm, corroborate, diagnose, find, judge, substantiate, testify, validate, verify, vouch for)
deny (deny, no, disagree)
disputative (demur, dispute, object, protest, question)
informative (advise, announce, apprise, disclose, inform, insist, notify, point out, report, reveal, tell, testify)
question (ask, inquire, interrogate, query, question, quiz)
requestive (ask, beg, beseech, implore, insist, invite, petition, plead, pray, request, solicit, summon, supplicate, tell, urge)
suggestive (conjecture, guess, hypothesize, speculate, suggest)
suppositive (assume, hypothesize, postulate, stipulate, suppose, theorize)

Which ever force has higher force count that force is returned.

Algorithm for getting Workflow pattern

Load Pattern.XML to Doc XMLDocument variable
Store all pattern elements to pattern variable
Load Transaction.XML to Doc XMLDocument variable
Get messages pairs and then get force pairs from Message.XML file
Match force pair to force pair in pattern xml and retrieve workflow pattern
Update pattern information to transaction xml at node "/Transaction/Transact[@transId='transID']"
Save Transaction.XML

Algorithm for writing workflow file

Load Transaction.XML to Doc XMLDocument variable
Store all transact elements to transact variable
For i = number of 'transact' elements -1 to 0 step -1
    If (date in 'transactID' equal to today's date)
        Write PNML structure of the workflow pattern
    End if
End for
Save workflow file

Algorithm for updating workflow history

Load WFhistory.XML to Doc XMLDocument variable
Update workflow file name to this node "/WFHistory/Customer[@CustID='customerID']" of WFHistory file
Save WFHistory file

Structure of XML files

Following are XML file structures used in this project

Login details

```
<Login>
    <Customer CustID="C-1">
        <FirstName></FirstName>
        <LastName></LastName>
        <ZipCode></ZipCode>
        <Email></Email>
    </Customer>
</Login>
```

Customer details

```
<CustomerDetail>
<Customer CustID="CD-1">
    <FirstName></FirstName>
    <LastName></LastName>
    <FirstLineAddress></FirstLineAddress>
    <SecondLineAddress></SecondLineAddress>
    <State>PA</State>
    <ZipCode></ZipCode>
    <Email></Email>
    <Product>
        <ProductName></ProductName>
        <SerialNo></SerialNo>
        <DatePurchased></DatePurchased>
        <WherePurchased/>
```

```
        </Product>
</Customer>
</CustomerDetail>
```

Workflow history

```
<WFHistory>
        <Customer CustID="C-1">
                <PNML/>
        </Customer>
</WFHistory>
```

Solution set

```
<Solution>
        <Set>
                <Question/>
                <Answer/>
        </Set>
</Solution>
```

Message

```
<Message>
        <Msg ID="msg-123">
                <Content/>
                <illForce/>
        </Msg>
</Message>
```

Transaction

```
<Transaction>
        <Trans ID="t-123">
                <Head>
                        <msgID>msg-123</MessageID>
                </Head>
                <Tail>
                        <msgID>msg-124</MessageID>
                </Tail>
                <Pattern></Pattern>
        </Trans>
</Transaction>
```

Workflow patterns

```
<WFPatterns>
        <PatternMatch>
                <Head>
                        <Force/>
                </Head>
```

```
                <Tail>
                        <Force/>
                </Tail>
                <Pattern><Pattern>
        </Pattern>
</WFPatterns>
```

PNML structure of sequence workflow pattern

```
 <pnml WFPID="pC-112032003">
        <net>
                <place id="P10A">
                        <graphics>
                                <position x="0" y="0" />
                        </graphics>
                        <initailmarking>
                                <value>1</value>
                        </initailmarking>
                        <name>
                                <text>mC-112032003083456</text>
                                <graphics>
                                        <position x="0" y="5" />
                                </graphics>
                        </name>
                </place>
                <arc id="a10A" source="P10A" target="t10A">
                        <graphics>
                                <position x="2" y="0" />
                        </graphics>
                        <incriptions>
                                <value>1</value>
                                <graphics>
                                        <offset x="0" y="0" />
                                </graphics>
                        </incriptions>
                </arc>
                <transition id="t10A">
                        <graphics>
                                <position x="5" y="0" />
                        </graphics>
                        <name>
                                <text>Sequence</text>
                                <graphics>
                                        <position x="5" y="5" />
                                </graphics>
                        </name>
                </transition>
                <arc id="a10B" source="t10A" target="P10B">
                        <graphics>
                                <position x="7" y="0" />
```

```
                        </graphics>
                        <incriptions>
                                <value>1</value>
                                <graphics>
                                        <offset x="0" y="0" />
                                </graphics>
                        </incriptions>
                </arc>
                <place id="P10B">
                        <graphics>
                                <position x="12" y="0" />
                        </graphics>
                        <name>
                                <text>mmc12032003083456
                                        <graphics>
                                                <position x="12" y="5" />
                                        </graphics>
                                </text>
                        </name>
                </place>
        </net>
</pnml>
```

Reference:

1. van der Aalst, W.M.P., A.H.M. ter Hofstede, B. Kiepuszewski, en A.P. Barros (2003). "Workflow Patterns". Distributed and Parallel Databases, 14(3), Pg: 5-51.
2. Austin, J.L (1962). “How to do things with words”. Oxford University Press, London.
3. Bach, Kent (2003). "Routledge Encyclopedia of Philosophy entry". http://online.sfsu.edu/~kbach/spchacts.html
4. Bach, Kent and Harnish, Robert (1979). “Linguistic Communication and Speech Acts”. MIT Press, Cambridge, MA.
5. Kammer, P. J. and D. W. McDonald (1999). "Putting Words to Work: Integrating Conversation with Workflow Modeling". Technical Report UCI-ICS-99-30, Information and Computer Science Department, University of California, Irvine.
6. Karsten, H (2000). “Weaving tapestry: Collaborative information technology and organizational change”. Jyvaskyla studies in computing. University of Jyvaskyla, Jyvaskyla, Finland.
7. Rob Allen (2001). “Workflow: An Introduction”; The Workflow Handbook 2001. Workflow Management Coalition (WfMC).
8. Searle J.R. (1969) “Speech Acts: An Essay in the Philosophy of Language”. Cambridge University Press, New York.
9. Weber, Michael and Kindler, Ekkart (2002). "The Petri Net Markup Language". Petri Net Technology for Communication Based Systems within the LNCS series Advances in Petri Nets.