Flexible Workflows: Reputation-based Message Routing

Ben Jennings and Anthony Finkelstein

University College London, UK
b.jennings@cs.ucl.ac.uk, a.finkelstein@cs.ucl.ac.uk

Abstract. This paper looks at the integration of human agents into composite workflows. Rather than using existing Worklist based mechanisms, it proposes a reputation recommendation based scheme based on data mining techniques used to analyse both passive and active 'gestures' within a system. The scheme provides a more flexible and effective solution to message routing than that in common use and gives more support to the agents engaged in the process of carrying out a task.

Key words: RMR, SOA, flexible, workflow, BPEL, BPEL4People

1 Introduction

The integration of human agents into composite business applications is an essential part of complex information systems development. As the reach of Service Oriented Architecture (SOA) extends to touch a wider variety of problem domains, the manner in which human agents are integrated and the mechanisms of interaction are becoming increasingly important. During the design and execution phase of a business process, when a human agent is required, a Worklist metaphor is widely used. Whilst this modelling of workflow patterns [1] approach would be perfectly adequate for a problem domain where the human agents were being used in a production style environment, it is neither the optimal solution where dynamic expertise is required, nor does it support the agents whilst they are carrying out such tasks.

This paper proposes a novel approach to the integration of human agents into a composite business process. Rather than a Worklist structure to route the workflow messages to any human agent or programatically assigned group, a reputation metaphor will be presented. This Reputation-based Message Routing (RMR) metaphor is based on the principle of observing passive and active gestures made within a system by the human agents and mining that data. From this information it is possible to make reputation based recommendations for routing of messages in a more flexible manner in order to find the most appropriate human agent at execution time. The other aspect of this analysis would be the ability for RMR to provide support for human agents to create ad hoc sub-process reputation based recommendations which would facilitate a higher quality of problem and exception resolution within the business process.

The paper will look first at the problem of human integration and at existing approaches, specifically the simplistic and proprietary human agent integration approach used most widely. Following this, a novel approach to the integration of human agents into a composite process will be discussed. Finally, some conclusions on the suggested architectural style and points of discussion will be presented.

2 Current Human Agent Metaphor:

This section of the paper will provide an overview of current methodologies for composite applications and human agents as part of a business process. The paper will then look at the BPEL4People proposal and the problem domain that it encompasses. Finally, there will be a more detailed look at the current Worklist metaphor used today and its conceptual limitations.

2.1 Composite Business Processes

Composite Business Processes are multi-step workflows, also known as composite applications, within the Service Oriented Architecture (SOA) context. Many composite workflows involve both human agents and web services. The orchestration of the services is by means of a workflow language and an execution engine. When human agents are integrated into a SOA solution, a new level of complexity is introduced. Rather than the system concerning itself with just modelling data and control messages between disparate services, the choice of *which* human agent and *what* data with which to present them becomes a concern of prime importance.

2.2 Human Agents

The de facto standard in the language to express the design and execution phase of a business process is WS-BPEL (Web Services Business Process Execution Language). BPEL is an XML [2] language and is a notation for defining business processes via Web Services, which has the concept of adapters as a communication mechanism to connect two disparate services. The adaptors act as an abstraction layer between services. IBM and SAP BPEL have put forward an extension to BPEL to try and address the issue of integrating human agents into web service based processes, BPEL4People [3]. This was proposed as an extension to WS-BPEL. Recently, BPEL4People has been reinvigorated by the addition of support from Adobe, via their LiveCycle ES product [4]. At the time of writing, this work has only just been submitted to OASIS and a technical committee formed [5]. This work is expected to take at the minimum eighteen months to get to the point of being a specification. After this, SOA application vendors will need to choose whether or not to adopt the new work and reengineer their execution engines, a non-trivial process, in accordance with their release cycles. As a point of reference, the MicroSoft BizTalk engine does not, as of yet, even have native BPEL support, instead using an import/export mechanism to translate supported BPEL processes into their own older XLANG language [6]. It is therefore reasonable to assert that the BPEL4People work, if completed, will take a significant time to come to market, be deployed and then interoperability problems will need to be addressed.

2.3 Worklist Metaphor

In the context of SOA, an agent is defined as an abstraction for a human interacting with a composite application. As most SOA solutions are concerned with the orchestration of computer based web services, a mechanism must be in place for the integration of human agents. The concept of a Worklist presents one unified web service for the interaction of human agent processes [7, 8]. From the perspective of the model, the Worklist is represented as a queue, with a clearly defined service based interface. This approach raises some fundamental conceptual issues. Worklists makes any human interaction with the workflow at a one stage removed level. As the orchestration message is passing to a generic service interface, the Worklist, no information about the individual human agent is accessible. When the workflow message has been passed to the Worklist abstraction, a complex monitoring solution needs to be part of the service in order to allow the parent process to trap any unforeseen errors. As the Worklist queue tends to separates any concept of orchestration of a specific agent from the process itself, compromises must be made. On the outside of the Worklist abstraction, the parent process must assume that any specific human agent will be interchangeable and atomic. From the inside, there is typically no concept of spawning lightweight, ad hoc sub-processes in a manner which facilitates interaction between agents. Dependant on implementation, support is only provided for messages such as start, stop, delegate etc. Typical ad hoc support [9] is built on the premise of creating models of predictable events and programmatically accessible groupings of agents established in design time, effectively abstracting a specific human to a role [10]. This methodology does not take into account the potential rich, quick fire, human agent to human agent disposable processes.

The issue with Worklist implementation is one of significance. In current SOA application stacks, the Worklist concept has been implemented in entirely different and incompatible manners: in Oracle Fusion Human Worklist Services (HWS) [11], WebSphere Human Task Manager (HTM) [12], in MicroSoft BizTalk Human Workflow Services (HWS) [13] and in the one reference implementation of YAWL Inbox [14]. In WebSphere, IBM have extended the Worklist concept even further by building on their BPELJ [15] proposed extensions with Task Extension Language (TEL) [16], using a custom human agent representation. This extension was originally proposed in 2004 and has had no adoption other than in the WebSphere engine [17]. All of these issues are likely to lead towards tightly coupled relationships in human processes which will, in turn, lead to brittle solutions. They also lose one of the main tenants of SOA, that of interoperability and communication. There is also a more fundamental concern, the lack of the concept of the system finding the most appropriate human agent within the system - the problem is outsourced.

The fundamental premise of the Worklist metaphor is predicated on one from which SOA itself is built, that of small, contained, replaceable units that will, from the perspective of the model, execute one atomic unit of work. These model based systems are built on the concept of a priori knowledge of the workflow: model first, then assign a performer [18]. In production style problem domains, such as logistics, the Worklist atomic human agent unit of work would be entirely appropriate. In a more complex domain, the Worklist architecture does not provide a rich enough vocabulary to express the needs of a human centred flexible workflow. The next section of this paper will present a new approach, that of Reputation-based Message Routing, and discusses how this approach may answer the limitations faced by the Worklist metaphor.

3 Reputation as a Metaphor:

In the previous section, this paper outlined the main thrust of current approaches to the integration of human agents into composite services. In this section, a new approach will be presented: that of RMR. First, in order to put some of the fundamental building blocks of this framework in place, the key concept of gesture analysis will be presented. A more detailed look at the cross cutting lifecycle applications will then be discussed. Finally, an overview of an early prototype of this system will be described to put the architectural approach in context.

All example scenarios in this section are based in the problem domain currently being explored experimentally, that of integrating the RMR approach in a software development testing workflow. This domain presents two specific advantages for this research. Depending on the specific project being analysed, there exists a large amount of electronic data, which can be mined. Secondly, the nature of interactions in this context is likely to be complex and unique so will therefore present a rich ground for analysis.

3.1 Gestures

One of the fundamental building blocks of a reputation-based system is that of gesture analysis. Gestures fall into two categories, that of a passive or that of an active gesture. A passive gesture is defined as the observation of human agents' interaction with a system. In the experimental scenarios, a common data artefact is a mailing list. From this pre-existing data set, on the assumption that the development group is not newly formed, a large data set of passive gestures may be mined. One rich source is that of the textual content itself.

Using Natural Language Processing (NLP) techniques [19], the nature of the content may be found. Nature, in this context, will be evinced as indexed keywords which will then be analysed by such techniques as frequency analysis, clustering and cross set correlation. A richer aspect of this data set can be found by the dynamic analysis of the threading and aggregation of such data. From this data, patterns of key terms will cluster around both related data and the human agents who interacted with the system.

An active gesture is defined as a directed input, or assertion, by a human agent to a system. These forms of active gesture are domain specific. The RMR architecture described in the following section provides a lightweight, low impact and broadly applicable mechanism for capturing both passive and active gestures. Rather than requiring significant upfront retooling work [20], the RMR system builds the mechanism for data capture into one of its own subsystems. Using a wrapper and parser modular approach, the RMR system can monitor user agent generated text content via multiple input methods, such as RFC2822 or ticketing systems. The text content can then be analysed by the system and subsequently modelled. This approach facilitates adoption, as no changes are required from an infrastructure perspective, only in the human agents themselves. Two examples of such active gestures are tagging and directed messages. By combining both active and passive gesture data, a richer quality of information can be captured and subsequently modelled. From this analysis, more knowledge about the human agents within the system can be elicited and a reputation generated.

Tagging and Directed Messages An approach, which has gained quick adoption in the Web 2.0 community, is that of tagging. A tag is a one or multi word descriptor to add metadata to a data object. With tagging, rather than having a predetermined ontology, the users of the resource add metadata, in the form of tags, to each data item. This user generated metadata is know as a folksonomy [21], an amalgam of the words folk and taxonomy. Rather than having a rigid, predetermined categorisation hierarchy, there is a flat structure.

The second active gesture concept is that of directed messages, popularised by the micro-blogging company Twitter [22]. Within a body of text, by using a specific lightweight syntax, a human agent may make an assertion to another human agent that they should be contacted about this context of data, as they would either be interested or have previous experience in the subject area. This deferral can provide information on a network of relationships of knowledge and from this, a more detailed graph of interactions can be leveraged.

Multi-Aspect Flexibility The RMR metaphor has many cross cutting applications and cross lifecycle uses. The analysis of both passive and active gestures, from a combination of data sources, provides a deep and wide data set from which insight into the executions of flexible workflows can be gleaned. Using new name disambiguation algorithms building on previous work [23, 24], within the RMR system and the correlation of disparate data sets, the patterns of human agent to human agent and human agent to web service interaction will create a reputation profile for each agent. The human agent reputation profile has many potential applications. The most obvious is that of message routing as part of a larger workflow, i.e. selecting the most appropriate agent for the job via a reputation algorithm. There are two more subtle areas of potential application, that of agent support and agent interaction.

One of the shortcomings of the Worklist metaphor, presented in the previous section, was the representation of human agents as singular atomic units of work from the perspective of the workflow model. The RMR metaphor allows and encourages real world scenarios of complex lightweight ad hoc dynamic subprocesses that occur between human agents. By utilising well understood data mining techniques [25, 26], analysis of ad hoc sub-processes is possible. This technique will enable greater flexibility in human agent methodologies. By encouraging actively the interaction between human agents and the system itself, via active gestures, a rich layer of reputation data will be gathered. Blending this data into the reputation system as a whole will enable the system to be more effective in its first reputation recommendation phase [27, 28]. This is a fundamentally different paradigm to that of traditional workflow modelling, suggesting a human centric solution, building on the knowledge and expertise of those carrying out the work, rather than relying solely on administrators creating a workflow model.

3.2 Prototype Architectural Overview

To test fully the ideas presented so far, a prototype RMR based system is currently under construction. The prototype is being constructed from a wide application reuse perspective, rather than relying on direct access to a database. This means that the prototype will be a general-purpose framework within the constraints of the system. The prototype is a client server architecture, based on a LAMP stack. The interface for the client will be based upon REST APIs so will support multiple client type interactions and interoperability.

Currently, three data types are supported: RFC 2822 (mailing list mbox format), Subversion (SVN) and the Trac ticketing system. The choice of these three data sources was based on significant popularity in open source development and therefore a widespread adoption.

Use Case To put the prototype in context, a brief use case within the test scenario will be described. This use case will demonstrate active gesture analysis, combined with historical passive analysis, to recommend an authoritative voice in the problem domain. Due to writing constraints only a short overview of the use case is possible. The primary actors in this use case are:

- Admin: the administrator of the project
- Developer A: the developer currently working on an issue
- Developer B: a developer with previous experience in the domain

Scenario: Developer A has just been hired to work on the project and has no previous knowledge of the expertise of other developers working within the project. Developer A does, however, have expertise in the problem domain. The administrator assigns Developer A a new task. Developer A, after careful analysis of the problem, asserts some classification tags in a mailing list post discussing the issue. From this, the Reputation-based Message Routing system makes a suggestion to Developer A of another developer in the project who has previous experience in this type of problem (Developer B). Developer A in their next post to the mail thread sends a direct message to Developer B, an active assertion. Developer B then participates in the thread and the issue is now resolved in a more expedient and flexible manner.

Preconditions: In order for passive analysis to occur, there must be a pre-existing body of data to analyse. These data sources must conform to open standards, or have custom parsing and pre-processing modules constructed for them. For this event to occur, a new task must be assigned by the administrator. Also, more than one developer must be working on the project.

Postconditions: The task must be closed in order for this process to terminate successfully. From the new gesture data generated in this process, the RMR system will act upon this to build further profile data for future instantiations of the submitting procedure.

State of Prototype Currently, the prototype is working towards building an aggregated view on the three data sources which can be gathered, via custom parsers, into the RMR system. From this data and the subsequent data mining, disambiguation and NLP techniques, a wide set of passive analysis is being undertaken. The system is also being fitted with active gesture input via the pre-processing modules, which will enable human agents to add further gesture data to the system. The results of this data will be in two phases. First, for the system to be able to make Reputation-based Message Routing suggestions which will improve the quality of the selection of human agents. The second phase will be enabling the human agents to feedback information directly into the system, via active gestures, which will make a more effective relationship between the two phases of the lifecycle of the workflow.

The reputation calculations that the prototype will make raises some interesting questions. As there are three passive gesture data sets and subsequently three active ones on a per human agent basis, different aspects of such data could be emphasised. Once a data event has been classified in the RMR system, the nature of both the originating human agent and the subsequent relations of others to such generated data can be calculated. This calculation will be in relation to the other data already in the system, both from the perspective of the human agent and each data object. The importance or weight put on each data set can be *tuned* within the RMR system by the administrator. This is necessary as some data will have different significance from others. For example, the Subversion commit log data provides interesting hierarchical structure to the group of human agents. As only some of the agents will have been granted commit rights, this reveals an initial structure to the group. From this, commit frequency then shows another aspect to this structure. The *tuning* in this case can emphasise the importance of gestures from those human agents in relation to the group as a whole.

Two further aspects of *tuning* are that of cost and forecasting. When considering the reputation of a specific human agent within a system, both the hourly rate and current load could be considered. Using the RMR mechanism, the most appropriate human agent could be selected for a task, rather than the one with the highest reputation score. Forecasting of requirements could also lead to interesting *tuning* within the RMR system. From the perspective of the administrator, determining the type and skill set of human agents moving forward has always proved challenging. As the reputation mechanism within the system would show emerging patterns of use, the *tuning* could be adjusted to a human agent with a lower score in order to increase their skill base in that area. This holistic view would help the group as a whole.

The requirements of the chief architects of the system can lead the direction of such *tunings* based on observational data and future planning of the flexible system. This cross cutting approach will increase the quality of interaction between the human agents and the system, providing previously unavailable metrics of the social interaction in ad hoc sub-processes and enabling a more flexible holistic workflow execution.

4 Conclusions and Future Work:

This paper has presented the current state of integrating human agents into workflow systems. It has also shown the potential limitations of the Worklist metaphor, the brittle interactions that can ensue and the fundamental lack of support for selecting the most appropriate human agent. The RMR architecture provides a novel approach to this problem space and enables a more natural human style of interaction to a composite workflow with potentially wide ranging implications. The dual approach of combining both passive and active gestures and the subsequent analysis has shown that human agent integration into a complex workflow can be significantly improved. The initial work from the prototype provides a firm foundation from which to build a larger set of results that will validate this approach. The prototype also provides evidence that a generalised, reusable architecture is viable. The prototype is a clearly bounded problem domain, so many of the algorithmic approaches would need to be modified to apply to different situations. The fundamental approach would be applicable to many domains where complex human agent interaction is required.

The approach opens up interesting questions on cross lifecycle interaction between human agents and web services. It also proposes a novel mechanism for gathering metrics about real world usage. Finally, it proposes in depth mining of information from both passive and active gestures of attention to enable the RMR system to learn and therefore provide a more flexible workflow.

References

1. van der Aalst, W.: Workflow patterns. http://www.workflowpatterns.com/ (2007)

2. Bray, T., J, P., CM, S.M., Maler, E., Yergeau, F.: Extensible markup language (xml) 1.0 (fourth edition). http://www.w3.org/TR/REC-xml/ (8 2006)
3. Kloppmann, M., Koenig, D., Leymann, F., Pfau, G., Rickayzen, A., von Riegen, C., Schmidt, P., Trickovic, I.: WS-BPEL Extension for People–BPEL4People. Joint white paper, IBM and SAP, July (2005)
4. Adobe: Adobe livecycle es. Adobe (2007)
5. Geyer, C.: Members approve web services business process execution language (ws-bpel) as oasis standard. http://www.oasis-open.org/news/oasis-news-2007-04-12.php (April 2007)
6. Juric, M.B., Mather, B., Sarang, P.: Business Process Execution Language for Web Services Second Edition. PACKT Publishing (2006)
7. Clugage, K., Shaffer, D., Nainani, B.: Workflow services in oracle bpel pm 10.1.3. http://tinyurl.com/2sal77 (3 2006)
8. Woodgate, S., Mohr, S., Loesgen, B., Adams, S.: Microsoft BizTalk Server 2004 UNLEASHED. Sams (2004)
9. Iyengar, A., Jessani, V., Chilanti, M.: WebSphere Business Integration Primer Process Server, BPEL, SCA and SOA. IBM Press (2008)
10. Havey, M.: Essential business process modeling. copyright 2005 o'reilly media. Inc **22** (2005) 36
11. Chappell, D.: Understanding bpm servers (2004)
12. Gadiyar, V.: Dynamic configuration of the human task manager in websphere process server. http://www.ibm.com/developerworks/websphere/library/-techarticles/0708_gadiyar/0708_gadiyar.html (August 2007)
13. Microsoft: Human workflow servicew (hws). http://technet.microsoft.com/en-us/library/aa577659.aspx (2008)
14. van der Aalst, W., Aldred, L., Dumas, M., ter Hofstede, A.: Design and implementation of the YAWL system. Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSEÕ04 **3084** (2004) 142–159
15. Blow, M., Goland, Y., Kloppmann, M., Leymann, F., Pfau, G., Roller, D., Rowley, M.: BPELJ: BPEL for Java. White paper.* ftp://www6. software. ibm. com/software/developer/library/ws-bpelj. pdf (2004)
16. Heumann, B., Konig, D., F, N., Pfau, G.: WebSphere Process Server v6.0 Business Process Choreographer Programming Model. IBM Corporation (2006)
17. Akram, A., Meredith, D., Allan, R.: Application of business process execution language to scientific workflows. International Transactions on Systems and Applications **501** (2006) 36535
18. Kirikova, M.: Flexibility of Organizational Structures for Flexible Business Processes. (2005)
19. Manning, C., Schutze, H.: Foundations of Statistical Natural Language Processing. Computational Linguistics **26**(2) (1999)
20. Horvitz, E., Kadie, C., Paek, T., Hovel, D.: Models of attention in computing and communication: from principles to applications. Communications of the ACM **46**(3) (2003) 52–59
21. Vanderwal, T.: Folksonomy. http://www.vanderwal.net/folksonomy.html (2007)
22. Twitter: Directed messages. http://help.twitter.com/index.php?pg=kb.page&id=15 (November 2007)
23. Mann, G., Yarowsky, D.: Unsupervised personal name disambiguation. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4 (2003) 33–40

24. Minkov, E., Wang, R., Cohen, W.: Extracting personal names from emails: Applying named entity recognition to informal text. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (2005) 443–450
25. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)
26. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2006)
27. Resnick, P., Zeckhauser, R.: Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. The Economics of the Internet and E-Commerce **11** (2002) 127–157
28. Heider, F.: Psychology Interpersonal Relations. Lawrence Erlbaum Associates (1982)