35

MODELLING MULTITHREADED SOCIAL PROTOCOLS WITH COLOURED PETRI NETS

Willy Picard
Department of Information Technologies
Poznań University of Economics
ul. Mansfelda 4, 60-854 Poznań, POLAND
picard@kti.ae.poznan.pl

Support for human-to-human interactions over a network is still insufficient, while important in the context of Web 2.0. Many works have to be done to provide both theoretical and practical knowledge to this field. The model of social protocols has been formerly presented as a model of human-to-human interactions. In this paper, an enhanced version of social protocols supporting multithreaded collaboration and based on coloured Petri nets is presented.

1. INTRODUCTION

The development and the wide adoption of Wikis, blogs and the concept of folksonomies have led to a generation of Web-based communities (also known as *virtual communities*) and hosted services referred by the term *Web 2.0*. The goal of the Web 2.0 hosted services is to support creation and sharing of contents among members of a given community.

The Web 2.0 is based on the idea that the Web should be not only a set of isolated information sources, but also a network of interrelated *services* providing users with various but complementary features. Examples of such services are Google Maps (Google Maps, 2008) for maps-based services, Del.icio.us (Del.icio.us, 2008) for storing, sharing and discovering Web bookmarks, and Flickr (Flickr, 2008) for photo sharing services.

The Web 2.0 is *participatory* as it involves the users in the production of the contents, instead of being based on the classical separation of roles between producers and consumers of contents. The Web 2.0 is *collaborative* as the contents are produced by a community of users instead of isolated users. Probably the most famous participatory and collaborative Web 2.0 site is the encyclopaedia Wikipedia[5] in which anyone may create or edit articles..

To summarize, the definition of the Web 2.0 that we are using in this paper is the following one:

> *The Web 2.0 is a network of interrelated services, used mainly by a large number of small communities which create contents in a participatory and collaborative way.*

In the Web 2.0, collaboration is a mean for community members to create *contents*. Interactions among community members are limited by the *collaboration protocols* ruling a given service. A collaboration protocol defines various roles and interactions among members playing these roles. As an example, most blog providers, e.g. (Google Blogger, 2008) or (Wordpress.com, 2008), define the roles of author, friend and anonymous. Authors may publish and modify posts on their blog, manage their list of friends, and set their blog as public or private. On a public blog, friends and anonymous users may read and comment any post, while on a private blog, only friends may read the blog and publish comments. Currently, collaboration protocols are not easily modifiable by the service providers nor community members as collaboration protocols are "hard-coded" in the code of the Website providing the service. Following on the previous example, the addition of a new role, such as editor, or the addition of new interactions, such as the possibility for a friend to invite new friends would lead to important changes in the code of the Website providing the service to be modified. A solution to this problem is the separation of collaboration protocols from the rest of the code of Website. Such an approach to collaboration has its root in workflow management systems in which the definition of the workflow is orthogonal to the implementation of the tasks to be performed during the workflow runtime.

Currently, many research works focus on Business Process Modeling and particularly Web services and associated standards: high-level languages such as BPEL or WS-Coordination take the service concept one step further by providing a method of defining and supporting workflows and business processes. However, most of these actions are directed towards interoperable machine-to-machine interactions over a network. Support for human-to-human interactions over a network is still insufficient and many research works have to be done to provide both theoretical and practical knowledge to this field. Emerging standards, such as (BPEL4People, 2007) and (WS-HumanTask, 2007), aim at providing better support for activities performed by humans in the BPEL framework. However, these two standards do not directly address human-to-human interactions but rather propose a formal definition of human activities and potential inclusion of these tasks within a BPEL process.

The insufficient support for human-to-human interactions over a network is a strong limitation for a wide adoption of *professional virtual communities (PVCs)*. As mentioned in (Camarinha-Matos, 2005), "professional virtual community represents the combination of concepts of virtual community and professional community. Virtual communities are defined as social systems of networks of individuals, who use computer technologies to mediate their relationships. Professional communities provide environments for professionals to share the body of knowledge of their professions […]". According to (Chituc, 2005), little attention has been paid to the social perspective on Collaborative Networks (CN) business environment, including obviously professional virtual communities in which social aspects are of high importance.

This paper is an attempt to provide a model for human-to-human interactions within professional virtual communities. The proposed model enhanced the former model of *social protocols* proposed by (Picard, 2005) by allowing members of a given community for a *multithreaded collaboration*. It should however been kept in mind that the results presented here are a work in progress and therefore they are not claimed to be neither sufficient nor exhaustive.

In this paper, the concept of multithreaded social protocols is proposed as an answer to the issues pointed above. In Section 2, the requirements for social protocols for professional virtual communities are presented. Next, coloured Petri nets, on which the proposed model for multithreaded social protocols is built, are briefly introduced. Then, a model for multithreaded social protocols is formally presented. In Section 5, an example illustrates a potential application of the proposed model for the collaborative building of a Frequently Asked Question list (FAQ). Finally, Section 6 concludes the paper.

2. REQUIREMENTS FOR SOCIAL PROTOCOLS FOR PVCS

In the context of professional virtual communities, the concept of social protocols has been proposed (Picard, 2005) as a model of interactions among collaborators within a given sub-community.

The following requirements for social protocols as a model of interaction among collaborators has been identified:

- *reusability:* a given social protocol should be reusable to rule the interactions within various sub-communities; A social protocol aims at modelling a set of collaboration processes, in the same way as a class models a set of objects in object-oriented programming. In other words, a social protocol may be seen as a model which instances are collaboration processes;
- *separation of tasks implementation from social protocols:* a social protocol should model potential interactions among collaborators, however the interactions should decoupled from implementation of the tasks performed by collaborators. As a consequence, tasks of a given social protocol may be implemented in various ways, using various technologies, or various locations/hosts;
- *support for social aspects in collaboration:* interactions are strongly related with social aspects, such as the role played by collaborators;
- *strong mathematical foundations:* social protocols model a complex situation of potential interactions among humans. Therefore, strong mathematical foundations are required as a mean to check properties such as structural validity, reachability, liveness and boundedness;
- *multithreading:* within a given sub-community, it is quite frequent that collaborators work on many topics at the same time. Therefore, social protocols should allow collaborators to work in a multithreaded way, i.e. to work simultaneously in many subprocesses.

The model of social protocols presented in (Picard, 2005) addresses the four first requirements. The fifth requirement is not answered by the formerly proposed model which is single-threaded, as based on Finite State Machines.

In this paper, an enhanced model of social protocols addressing the five requirements presented above is presented. The enhanced model extends the former one by replacing Finite State Machines by Coloured Petri Nets.

3. COLOURED PETRI NETS

Petri Nets were invented by (Petri, 1962) as part of his dissertation, *Kommunikation mit Automaten* (communication with automata) as mathematical representations of discrete distributed systems. Application of Petri Nets to workflow management have been studied by (Aalst, 1998) with the conclusion that “many features of the Petri net formalism are useful in the context of workflow management”.

A classical Petri net consists of *places*, *transitions* and *directed arcs* between places and transitions. Places may contain *tokens*. The set of tokens, potentially spread in many places is called a *marking*. A marking may change when a transition is *fired* (or triggered). The firing of a transition *t* is possible iff all places *p* for which a directed arc from *p* to *t* exists (*input places*) contain at least one token. Where a transition *t* is fired, tokens from the input places are consumed, an action is potentially performed, and new tokens are placed in *output places* (i.e. places to which a directed arc from transition *t* leads).

An example of a classical Petri net is presented on Figure 1. The presented Petri net consists of 5 places (represented by circles) and 2 transitions (represented by rectangles). All input places of the left transition contain at least one token, therefore the left transition may be fired. The state of the Petri net before the firing of the left transition is presented in Figure 1a), while the state resulting from the firing of the left transition is presented in Figure 1b). Tokens in the left places are removed and a token is placed in the central place. Then, the right transition may be fired. After the right transition has been fired, tokens are placed in the output places of the right transition, as presented in Figure 1c). Now the left transition can not be fired because only the top-left place contains a token and the bottom-left place which is required to fire the left transition does not contain any token.

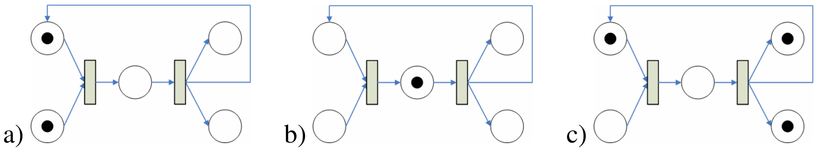

Figure 1 - a) an example of a Petri net; b) after firing the left transition; c) after firing the right transition

Coloured Petri nets (Jensen, 1996) has been proposed as an extension to classical Petri nets. Coloured Petri Nets (CPN) incorporate the concept of data structuring which was missing in Petri’s original works. In a CPN, each token has a value often referred to as ‘*colour*’. Coloured token may be used to represent objects in the modelled system. A coloured token may for instance represent a post or a comment in a blog. The colour of a token may be used in directed arcs to specify 1) the type of data that a given transition requires and processes, 2) the type of data that a transition produces.

An example of a coloured Petri net based on the former example of Petri Net is presented on Figure 2. Tokens, as well as directed arcs, are now associated with data types. The left transition requires tokens associated with data types *a* and *c*. When this transition is fired, the token associated with the data types *a* and *c* are removed and a new token associated with data type *d* is placed in the central place.

An important feature of Petri nets, and by extension an important feature of CPNs, is the possibility to model multi-threaded systems as Petri nets. Indeed, an important principle of Petri nets is the principle of locality: the behaviour of a transition exclusively depends on its locality, i.e. its input and output places. As a consequence, various transitions may be simultaneously enabled (i.e. they can be fired) if the marking contains tokens in all input places of these transitions.

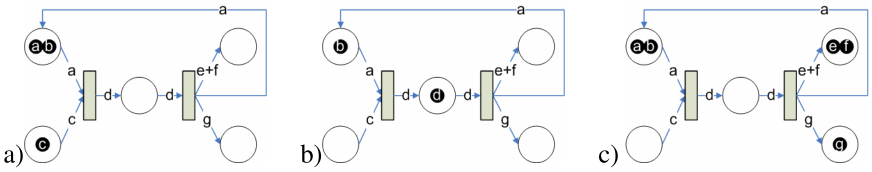

Figure 2 – a) an example of a coloured Petri net; b) after firing the left transition; c) after firing the right transition

4. A MODEL FOR MULTITHREADED SOCIAL PROTOCOLS

4.1 Single-threaded social protocols

The model of multithreaded social protocols proposed in this paper is an extension of the model of social protocols presented in (Picard, 2005). “A social protocol aims at modelling a set of collaboration processes, in the same way as a class models a set of objects in object-oriented programming. In other words, a social protocol may be seen as a model which instances are collaboration processes. Social protocols model collaboration at a group level.”.

The former model consists of a finite state machine (FSM) modelling potential interactions within a group. In these finite state machines, transitions are labelled by roles and are associated with actions. “In a social protocol, collaborators – as a group – move from state to state via the transitions. A transition may be triggered only by a collaborator labelled with the appropriate role. A transition is associated with the execution of an action. Execution of an action means the execution of remote code. SOAP or CORBA are examples of technologies that may be used to such remote code executions.”

The first three requirements presented in Section 2 are clearly addressed by the former model. Mathematical foundations are in the theory of automata. Additionally, an algorithm for structural validation of social protocols has been presented in (Picard, 2007). However, the fifth requirement, i.e. multithreading, is not supported by the former model as finite state machines are not well-adapted to model multithreaded systems. The main contribution of this paper is the replacement of FSMs by CPNs in social protocols to support multithreaded processes.

4.2 Formal model for multithreaded social protocols

A ***multithreaded social protocol*** *Π-net* is a coloured Petri net consisting of (P, T, A, C, I, R, f_A, f_{TI}, f_{TR}, M_0) where

- P is a finite set of places,

- T is a finite set of transitions, such that $P \cap T=\emptyset$,
- A is a finite set of arcs, such that $A \cap T= A \cap P =\emptyset$,
- C is a finite set of data types, also called colour sets,
- R is a finite set of roles,
- I is a finite set of interfaces to actions,
- f_A is the arc function defined from A to $P\times T\times C \cup T\times P\times C$, defining arcs as triples (place, transition, colour set),
- f_{TI} is the interface function from T to I, associating transitions with interfaces,
- f_{TR} it the role function from T to R, associating transitions with roles,
- M_0 is the initial marking.

A multithreaded social protocol consists of places and transitions. Places and transitions are connected with directed arcs which are associated with colour sets. Transitions are associated with an interface of an action to be performed if the transition is triggered by a collaborator playing the associated role. Finally, the initial state of a multithreaded social protocol is defined by its initial marking.

A ***token*** is a pair (p, c) where $p\in P$ and $c\in C$.

A ***marking*** is a set of tokens.

A ***multithreaded social process*** is an instance of a multithreaded social protocol. A multithreaded social process consists of (*Π-net*, U, f_{UR}, M) where

- *Π-net* is a multithreaded social protocol,
- U is a finite set of users,
- f_{UR} is the role-user mapping function from U to R, associating users with roles,
- M is the current marking.

When a multithreaded social protocol is instantiated, roles have to be attributed to existing users and M is initialized to M_0. Next, users trigger enabled transitions, modifying the current marking.

A ***transition $t\in T$ is enabled*** for (may be triggered by) user $u\in U$ iff:

- $f_{TR}(t) = f_{UR}(u)$, i.e. user u plays the role r associated with transition t,
- $\forall a\in A \mid \exists (p,c)\in P\times C$ such that $f_A(a)= (p,t,c)$, $(p,c)\in M$, i.e. for all arcs leading to transition t, there exists at least one token of the corresponding colour sets in the current marking.

The ***firing of a transition $t\in T$*** is performed in the following way:

- $\forall a\in A \mid \exists (p,c)\in P\times C$ such that $f_A(a)= (p,t,c)$, $M=M-(p,c)$, i.e. for all arcs leading to transition t, tokens of the corresponding colour sets in the current marking are removed from it.
- $f_{TI}(t)$ is executed, potentially using the removed tokens as arguments,
- $\forall a\in A \mid \exists (p,c)\in P\times C$ such that $f_A(a)= (t,p,c)$, $M=M+(p,c)$, i.e. for all arcs coming from transition t, tokens of the corresponding colour sets are added to the current marking.

5. AN EXAMPLE OF MULTITHREADED SOCIAL PROTOCOL

The example of multithreaded social protocol which is presented in this section is oversimplified for readability reasons. It is obvious that social protocols modelling real-world collaboration processes are usually much more complex. The same example has been presented in its single-threaded version in (Picard, 2006).

The chosen collaboration process to be modelled as a multithreaded social protocol is the collaborative authoring of a "FAQ" document. Some users (denoted by the letter "u" in transitions) only ask questions, while others, referred as "experts" (denoted by the letter "e" in transitions) may answer, remove or comment on questions. Other users, referred as "managers" (denoted by the letter "m" in transitions), may interrupt the work on the FAQ document. The work on the document may terminate either by a success (the document is written and the manager estimates that its quality is good enough to be published) or by a failure (the manager estimates that the work on the FAQ should be interrupted). On arcs, the colour "start" refers to the initial token, "q" to a question, "a" to an answer, "r" to a removed question, "c" to a comment, "failure" to the final state of the collaboration when the FAQ is not published, and "success" to the final state of the collaboration when questions, answers and comments are published. One may notice that some colours are prefixed with the negative sign "-": in this case, the token in the input place is removed when the transition is fired, else the transition is enabled iff the token exists but it will not be removed.

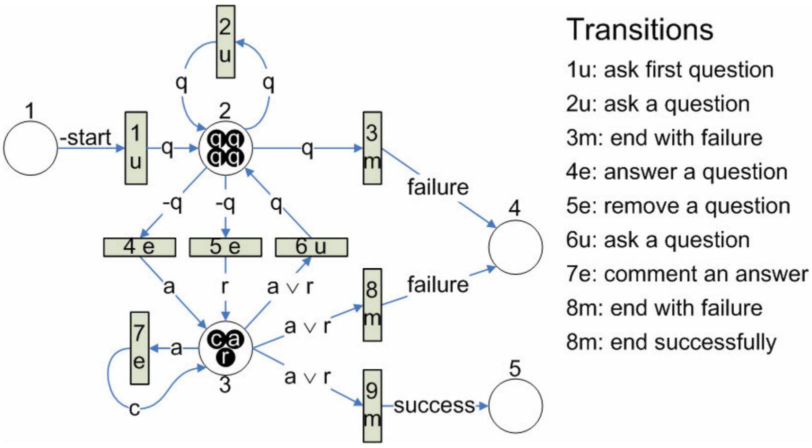

Figure 3 – Example of multithreaded social protocol for FAQ list building

In the situation presented in Figure 3, four questions are waiting for answers or removal, one question has been answered and commented, one question has been removed. The multithreaded aspect of the presented social protocol is visible in the current situation as many transitions, e.g. "2u", "4e", and "7e" may be performed concurrently: a user may ask a new question ("2u"), while an expert answers an existing question ("4e") and a second expert comments on the formerly answered question ("7e").

6. CONCLUSION

While many works are currently done on modelling collaboration processes in which software entities (agents, web services) are involved, modelling collaboration processes in which mainly humans are involved is an area that still requires much attention from the research community. The model presented in this paper is an attempt to provide a formalized model of human-to-human interactions.

The main innovation presented in this paper is the extension of the social protocol model by replacing FSMs by CPNs so that multithreaded processes are supported. Additionally, the strong mathematical foundations of Petri nets, with the existence of many analysis techniques, ensure that powerful analysis capabilities may be added to information systems supporting multithreaded social protocols.

The next steps will include modification of our technique for the adaptation of single-threaded social protocols to multithreaded social protocols, allowing collaborators to modify a multithreaded social protocol at runtime.

7. REFERENCES

[Aalst 1998] W.M.P. van der Aalst, "The Application of Petri Nets to Workflow Management", The Journal of Circuits, Systems and Computers, 8(1), pp. 21 – 66, 1998.

[BPEL4People, 2007] WS-BPEL Extension for People (BPEL4People), Version 1.0 June 2007, https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/30c6f5b5-ef02-2a10-c8b5-cc1147f4d58c , accessed on March 2008.

[Camarinha-Matos 2005] L.M. Camarinha-Matos, H. Afsarmanesh and M. Ollus, "ECOLEAD: A Holistic Approach to Creation and Management of Dynamic Virtual Organizations", In L. Camarinha-Matos, H. Afsarmanesh and A. Ortiz, Eds, *Collaborative Networks and their Breeding Environments*, Proceedings of the 6th IFIP Working Conference on Virtual Enterprises (PRO-VE 2005), Valencia, Spain, September 26-28, 2005, Springer, pp. 3 – 16, 2005.

[Chituc 2005] C.M. Chituc, A.L. Azevedo, "Multi-Perspective Challenges on Collaborative Networks Business Environments", In L. Camarinha-Matos, H. Afsarmanesh and A. Ortiz, Eds, *Collaborative Networks and their Breeding Environments*, Proceedings of the 6th IFIP Working Conference on Virtual Enterprises (PRO-VE 2005), Valencia, Spain, September 26-28, 2005, Springer, pp. 25 – 32, 2005.

[Del.icio.us 2008] Del.icio.us, http://del.icio.us/, accessed on March 2008.

[Flickr 2008] Flickr, http://www.flickr.com/, accesses on March 2008.

[Google Maps 2008] Google Maps, http://maps.google.com/, accessed on March 2008.

[Google Blogger 2008] Google Blogger, http://www.blogger.com/, accessed on March 2008.

[Jensen 1996] K. Jensen. *Coloured Petri Nets. Basic concepts, analysis methods and practical use.* EATCS monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1996.

[Petri 1962] Petri, Carl A. "*Kommunikation mit Automaten*". Ph. D. Thesis. University of Bonn. 1962

[Picard 2007] W. Picard, "An Algebraic Algorithm for Structural Validation of Social Protocols", Proceedings of the 10th Int. Conference on Business Information Systems, Lecture Notes in Computer Science, 4439, Springer, pp. 570–583, 2007.

[Picard 2006] W. Picard, "Computer Support for Adaptive Human Collaboration with Negotiable Social Protocols". In A. Witold and H.C. Mayr, Eds, *Technologies for Business Information Systems*, Proceedings of the 9th Int. Conference on Business Information Systems in cooperation with ACMSIGMIS, Klagenfurt, Austria, May 31 – June 2, 2006, Springer Verlag, pp. 193 – 203, 2006.

[Picard 2005] W. Picard, "Modeling Structured Non-monolithic Collaboration Processes", In L. Camarinha-Matos, H. Afsarmanesh and A. Ortiz, Eds, *Collaborative Networks and their Breeding Environments*, Proceedings of the 6th IFIP Working Conference on Virtual Enterprises (PRO-VE 2005), Valencia, Spain, September 26-28, 2005, Springer, pp. 379 – 386, 2006.

[Wordpress.com 2008] Wordpress.com, http://www.wordpress.com/, accessed on March 2008.

[WS-HumanTask 2007] Web Services Human Task (WS-HumanTask), Version 1.0, June 2007, https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/a0c9ce4c-ee02-2a10-4b96-cb205464aa02 , accessed on March 2008.