Adaptive Human-to-Human Collaboration via Negotiations of Social Protocols

Willy Picard

The Poznań University of Economics
ul. Mansfelda 4,
60-854 Poznań, Poland
picard@kti.ae.poznan.pl

1. Introduction

Enterprises are increasing constantly their efforts in order to improve their business processes. A main reason for this may be the fact that enterprises are exposed to a highly competitive global market. As a consequence, enterprises improve their business processes to become more competitive and to increase their performances. Among the most visible actions associated with this effort towards better support for better business processes, one may distinguish the current research work concerning 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, it should be notice that 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 much research has to been done to provide both theoretical and practical knowledge to this field.

Among various reasons for the weak support for human-to-human interactions, one may distinguish the following two reasons: first, many *social elements* are involved in the interaction among humans. An example of such a social element may be the roles played by humans during their interactions. Social elements are usually difficult to model, i.e. integrating non-verbal communication to collaboration models. Therefore, their integration to a model of interaction between humans is not easy. A second reason is the *adaptation capabilities* of humans which are not only far more advanced than adaptation capabilities of software entities, but also not taken into account in existing models for collaboration processes.

This paper is a try to provide a model for human-to-human interactions which addresses, at least to some extend, the two characteristics of the interactions between humans. 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 sufficient.

The rest of this paper is organized as follows. In section 2, the concept of *social protocol*, used to model collaboration processes, is presented. Section 3 then expands on the use of *negotiation* as a mean for *adaptation* of social protocols. Next, related work is reviewed. Finally, section 5 concludes this paper.

2. Modeling Collaboration Processes as Social Protocols

A social protocol aims at modeling 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 (Picard 2005b).

Social protocols model collaboration at a group level. The interactions of collaborators are captured by social protocols. Interactions are strongly related with social aspects, such as the role played by collaborators. The proposed model integrates some of these social aspects, which may explain the choice of the term “social protocols”.

2.1. Formal model of social protocols

Before social protocols may be formally defined, others concepts must first be defined, as well as the related notation.

Role. A role r is a label. Let's denote R the set of roles.

In a given group, a set of roles is played by the collaborators, which means that collaborators are labeled, are associated with given roles. The set of roles R_g played in a given group g is a subset of R, i.e. $R_g \subset R$. Collaborators usually play different roles. Roles may be associated with collaborators to specify the way they should interact with the rest of the group. Interactions among collaborators are modeled with the concept of *action*.

Action. An action a is an execution of a software entity. Let's denote A the set of actions.

An action may be for instance the execution of a web service, a commit to a CVS repository, the sending of an email. Within a group, collaborators are interacting by executing actions. The execution of actions is a part of the common knowledge of the group, i.e. all collaborators are aware of the execution of an action by one of the members of the group.

Behavioral Unit. A behavioral unit bu is a pair *(role, action)*. Let's denote BU the set of potential behavioral units. Formally, $BU=R\times A$.

The concept of behavioral unit comes from the idea that the behavior of a collaborator is to a large extend determined by the role he/she plays. Therefore, roles and actions have to be associated to determine the behavior, i.e. the set of actions that a collaborator playing a given role should expose.

By extension, one may say that a behavioral unit is executed. A behavioral unit $bu=(r,a)$ is said to be executed iff a collaborator labeled with role r executes action a. It should be noticed that only collaborators labeled with role r can execute the behavioral $bu=(r,a)$.

State. A state s is a label associated with a given situation in a collaborative process. Let's denote S the set of states.

In a given collaborative process p, the set of states that may occur S_p is a subset of S, i.e. $S_p \subset S$.

Transition. A transition t is a triplet $(bu, s_{source}, s_{destination})$. Let's denote T the set of transitions. Formally, $T=BU\times S\times S$.

Now that all concepts underlying social protocols have been formally presented, the concept of social protocol may be defined.

Social Protocol. A social protocol p is a finite state machine consisting of $\left\{S_p, S_p^{start}, S_p^{end}, R_p, A_p, \Delta_p\right\}$ where $S_p^{start} \subset S_p$ is the set of starting states, $S_p^{end} \subset S_p$ is the set of ending states, $S_p^{start} \cap S_p^{end} = \varnothing$ and Δ_p: $T_p \rightarrow [0,1]$. Let's denote P the set of social protocols.

In a social protocol, collaborators are moving from state to state via the execution of behavioral units. In other words, the executions of behavioral units are transition conditions. As mentioned before, a behavioral unit may be executed only by a collaborator labeled with the appropriate role.

In the context of social protocols, the Δ_p function puts an additional constraint on the execution of behavioral units. The Δ_p function defines the "desirability" of a transition within the given protocol for the whole group. The highest the value of the Δ_p function for a transition *t*, the highest the desirability of this transition for the group. If the value of the Δ_p function for a transition *t* is zero, then the group does not desire this transition to be executed.

The conditions that protocols have to fulfill to be valid, both structurally and semantically have already been presented in (Picard 2005a).

2.2. Social protocol example

The example of social protocol which is presented in this section is oversimplified for readability reasons. It is obvious that social protocols modeling real-world collaboration processes are usually much more complex.

The chosen collaboration process to be modeled as a social protocol may be described as follows: a set of users are collaborating on the establishment of a "FAQ" document. Some users only ask questions, while others, referred as "experts" may answer the questions. Other users, referred as "managers", are may interrupt the work on the FAQ document. The work on the document may terminate either by a success (the document has been written and the manager estimates that its quality is good enough to be published) or by a failure (the users did not find any way to collaborate and the manager has estimated that the work on the FAQ should be interrupted).

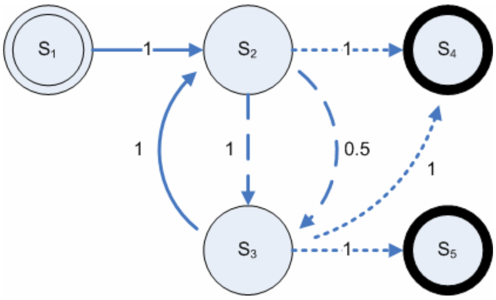

Fig. 1. Example of social protocol

A possible model of this collaboration process as a social protocol is presented in Figure 1. In Figure 1, five states $s_1,...,s_5$ are represented as circles. State s_1 is a starting state, states s_4 and s_5 are ending states. States are named as follows:

- state s_1: *waiting for first question*
- state s_2: *waiting for answer*
- state s_3: *waiting for next question*
- state s_4: *failed termination*
- state s_5: *successful termination*

Transitions are represented as arrows, and the line style is associated with the role of the users that may execute a given transition. Continuous line style is used to represent transitions that may be executed by “normal users”, fine-dashed style for transitions that may be executed by “experts”, and fine-dotted style for transitions that may be executed by “managers”.

The figures closed to the arrows represented the value of the desirability function for the associated transition. Transitions are summarized in Table 1.

Table 1. Transitions for the example of social protocol and their associated desirability values

Source state	Destination state	Role	Action	φ
s_1	s_2	Normal	Ask question	1
s_2	s_3	Expert	Answer question	1
s_2	s_3	Expert	Suppress question	0.5
s_2	s_4	Manager	Failure ending	1
s_3	s_2	Normal	Ask question	1
s_3	s_4	Manager	Failure ending	1
s_3	s_5	Manager	Successful ending	1

2.3. Social protocol filtering

The introduction of the Δ_p function is one of the main innovations presented in this paper. It allows collaborators for presenting various granularity levels of a given social protocol with regards to a *desirability threshold*.

Binary Social Protocol. A social protocol p is a binary social protocol iff its desirability function takes only the values 0 and 1, i.e. Δ_p: $T_p \rightarrow \{0,1\}$. Let's denote P_{01} the set of binary social protocols.

Let's assume that the desirability threshold equals θ, with $0 < \theta \leq 1$. Social protocol filtering consists in transforming a social protocol into a binary social protocol, by “suppressing” all transitions whose desirability is inferior to the desirability threshold. Formally, social protocol filtering may be defined as follows:

Social Protocol Filtering. Given a desirability threshold θ, with $0 < \theta \leq 1$, social protocol filtering is a function $\varphi : P \times [0,1] \rightarrow P_{01}$ such that

$$\varphi(p,\theta) = p' \text{ with } S_p = S_{p'}, S_p^{start} = S_{p'}^{start}, S_p^{end} = S_{p'}^{end}, R_p = R_{p'}, A_p = A_{p'}$$

$$\text{and } \Delta_{p'}(t) = \begin{cases} 0, & \text{if } \Delta_p(t) < \theta \\ 1, & \text{if } \Delta_p(t) \geq \theta \end{cases}$$

An example of social protocol filtering is presented on Figure 2. In this example, the result of the filtering of the protocol presented in section 2.2 for the value of the desirability threshold $\theta = 0.6$. All transitions with a desirability value lowest than the desirability threshold, i.e. the transition allowing experts to suppress a question, have been suppressed.

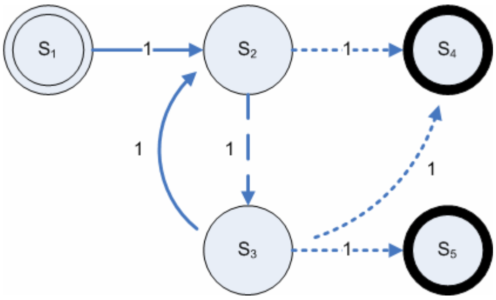

Fig. 2. Filtered social protocol presented in section 2.2 for the value of the desirability threshold *θ = 0.6*

2.4. Social Protocol Design

The proposed model for collaboration processes may be used to design protocol-based collaboration support systems in which social aspects are taken into account. Following the model, specification of a protocol-based collaboration support system involves four areas: action specification, role specifications, states specifications, and, desirability.

The specification of actions focuses on the definition of the functionalities that the collaborators need to achieve their common goal. All software entities needed to achieve this goal should be inventoried and documented.

The specification of roles focuses not only on the identification of the roles that may be played in the related collaboration processes, but also on the definition of behavioral units which are required.

When needed behavioral units have been identified, states may be specified. To do so, all situations that may occur during related collaborative processes should be identified and documented. These situations may then be mapped to states.

Having states and behavioral units, the only missing element for a complete description of a social protocol is the definition of the desirability function. During the design of the desirability function, one should start by identifying transitions which are mandatory (resp. are forbidden or make no sense) and assign them with the value 1 (resp. 0).

3. Adaptation of Social Protocols via Negotiation

While social protocols support, at least to some extend, the integration of some social elements (such as roles) to models of interactions among humans, the adaptation capabilities of humans are not taken into account into social protocols. There is however the need to provide adaptation mechanisms to social protocols. Indeed, interactions among humans are often a context-aware activity. In this paper, context-awareness refers to the capabilities of applications to provide relevant services to their users by sensing and exploring the users' context (Dey et al. 2001; Dockhorn Costa et al. 2005). Context is defined as a “collection of interrelated conditions in which something exists or occurs” (Dockhorn Costa et al. 2005). The users' context often consists of a collection of conditions, such as, e.g., the users' location, environmental aspects (temperature, light intensity, etc.) and activities (Chen et al. 2003). The users' context may change dynamically, and, therefore, a basic requirement for a context-aware system is its ability to sense context and to react to context changes.

Adaptive mechanisms are therefore required as complements to the formerly proposed model for human collaboration processes. The mechanism proposed in this paper is based the idea that social protocols may be negotiated. Two aspects of social protocols may be negotiated independently: first, the desirability function may be negotiated; second, states/behavioral units sets may be negotiated.

3.1. Desirability negotiation

The first element of social elements that could be the object of adaptation may be the desirability function. The values taken by desirability function for various transitions define the desirability of the whole group with regards to single transitions. By modifying the value of the desirability function, the whole group may adapt the social protocol to the situation in which the group is.

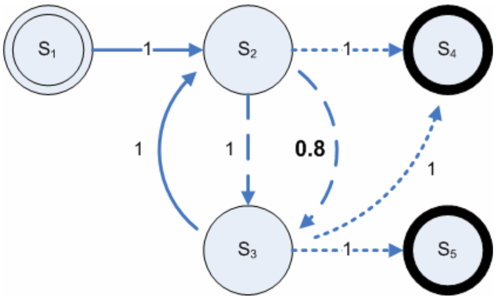

Fig. 3. Filtered social protocol presented in section 2.2 after the desirability value of transition “suppress question” has been increased by 0.3 (desirability threshold $\theta = 0.6$)

By increasing the desirability value of a given transition, a group may decide that a transition is “desirable” for a given desirability threshold, and therefore the transition associated with the modified value will become available. By decreasing the desirability value of a given transition, a group may decide that a transition is not “desirable” any more, and therefore the transition associated with the modified value will become unavailable for a given desirability threshold.

Table 2. Transitions for the example of social protocol and their associated desirability values

Source state	Destination state	Role	Action	φ
s_1	S_2	Normal	Ask question	1
s_2	S_3	Expert	Answer question	1
s_2	**s_3**	**Expert**	**Suppress question**	**0.8**
s_2	s_4	Manager	Failure ending	1
s_3	s_2	Normal	Ask question	1
s_3	s_4	Manager	Failure ending	1
s_3	s_5	Manager	Successful ending	1

Effects of a potential modification of the desirability function of social protocol presented in Section 2.2 are presented in Figure 3. In the presented example, the original social protocol presented in Section 2.2 has been adapted by the whole group via negotiations. The result of the negotiation is the group agreement stating that the desirability value for the transition “suppress question” has to be increased by 0.3. The modified desirability values associated with transitions are presented in Table 2.

After filtering, the adapted version of the social protocol is presented in Figure 3. Comparing with Figure 2, one may notice that the transition “suppress question” is now a desirable transition, at the desirability threshold of 0.5, which was not true before adaptation.

3.2. Structural negotiation

The second element of social elements that could be the object of adaptation may be the set of states and/or the set of behavioral units. The set of states consists of the set of situations that may occur during the life of a collaboration process. The set of behavioral units consists of the set of interactions that collaborators may perform.

By adding/suppressing state(s), the whole group may adapt a social protocol by providing/suppressing situation(s) to the collaboration process. It should be noticed that the addition/suppression of state(s) is related with the addition/suppression of transitions leading and originating from the modified state(s).

By adding/suppressing transition(s), the whole group may adapt a social protocol by providing/suppressing interaction(s) to the collaboration process. It should be noticed that the addition/suppression of transition(s) is usually not related with the addition/suppression of state(s) to/from which the added/suppressed transition(s) lead(s)/originate(s).

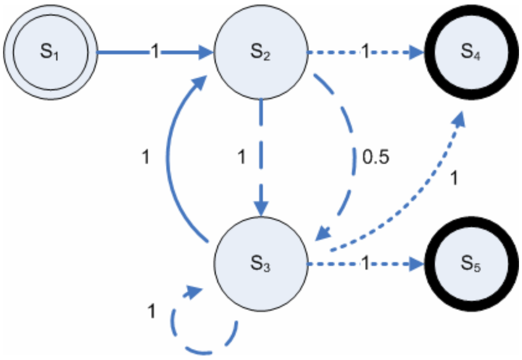

Fig. 4. Social protocol presented in section 2.2 after a transition related with the action “comment a question” has been added from s_3 to s_3.

Effects of the addition of a transition in the social protocol presented in Section 2.2 are presented in Figure 4. In the presented example, the original social protocol presented in Section 2.2 has been adapted by the whole group via negotiations. The result of the negotiation is the group agreement stating that a new transition is needed so that an expert may comment a question many times before answering it. The modified set of transitions is presented in Table 3.

Table 3. Transitions for the example of social protocol and their associated desirability values

Source state	Destination state	Role	Action	φ
s_1	s_2	Normal	Ask question	1
$\mathbf{s_3}$	$\mathbf{s_3}$	**Expert**	**Comment question**	**1**
s_2	s_3	Expert	Answer question	1
s_2	s_3	Expert	Suppress question	0.5
s_2	s_4	Manager	Failure ending	1
s_3	s_2	Normal	Ask question	1
s_3	s_4	Manager	Failure ending	1
s_3	s_5	Manager	Successful ending	1

4. Related Work

As process modeling is concerned, many works have already been conduced in the research field of workflow modeling and workflow management systems. Paul Buhler and Jose M. Vidal (Buhler and Vidal 2005) proposed a mechanism allowing for enacting workflows in an adaptive way using multi-agent systems (MAS). Robert Müller and al. presented in (Müller et al. 2004) various mechanisms for adaptation of workflows to deal with exception occurrences in running workflow instances, with an application to medical treatments. However, to our best knowledge, current works concerning workflow adaptation focus on interactions among software entities. Characteristics of interactions between humans, such as the importance of social aspects, are not or insufficiently taken into account by these works.

Still in the field of workflows, some works (Aalst et al. 2000) have focused on formal models and conditions under which a modification of an existing – and potentially running – workflow retains workflow validity. However, in the case of human interactions, some of these conditions may be relaxed as adaptation of a social protocol may lead to a social protocol which is temporally invalid. Such a case appears when a new state is introduced. The state exists but transitions leading to it have to be defined. The same applies for transitions having the brand-new state as a source.

Some interesting works have been done in the field of electronic negotiations to model electronic negotiations with the help of negotiation protocols. In (Kersten et al. 2004), it is stated in that, in the field of electronic negotiations, “the protocol is a formal model, often represented by a set of rules, which govern software processing, decision-making and communication tasks, and imposes restrictions on activities through the specification of permissible inputs and actions". One may notice the similarity with the concept of social protocol. The reason for this fact is that the model presented in this paper was originally coming from a work on protocols for electronic negotiations (Picard and Huriaux 2006). However, to our knowledge, none of the works concerning negotiation protocols provides mechanisms for protocol adaptation. Moreover, these works are by nature limited to the field of electronic negotiations which is just a subset of the field of human collaboration.

5. Conclusions

While many works are currently done on modeling collaboration processes in which software entities (agents, web services) are involved, modeling collaboration processes in which mainly humans are involved is an area that still requires much attention from the research community. Some of the main issues to be addressed are the social aspects of collaboration and the adaptation capabilities of humans. In this paper both issues are addressed. The concept of social protocol aims at being a start of answer to the question of computer support for social collaboration. The idea of negotiation of social protocol is a try to weaken constraints usually limiting the interaction between collaborators, so that the adaptation capabilities of humans may be integrate in the life of a social protocol.

The main innovations presented in this paper are 1) the introduction of the desirability function as a way to provide filtering functions to social protocols, 2) the idea of negotiation of social protocols, based either on negotiation of the desirability function or on the negotiation of the structure of the protocol. The proposed concepts are currently under implementation as extensions to the *DynG* protocol (Huriaux and Picard 2005), a social protocol-based platform. The next steps will include a refinement of the concept of role, so that relationships between roles, e.g. specialization, compositions, may be integrate to the presented model. Automated support for social negotiation would be an interesting feature for a social adaptive protocol-based framework, but negotiation models supporting contextual and social elements are still to be built.

References

van der Aalst WMP, Basten T, Verbeek HMW, Verkoulen PAC, Voorhoeve M (2000) Adaptive Workflow: On the Interplay between Flexibility and Support. In: Filipe J (ed) Enterprise Information Systems, Kluwer Academic Publishers, Norwell, pp. 63–70

Buhler P, Vidal JM (2005) Towards Adaptive Workflow Enactment Using Multiagent Systems. Information Technology and Management Journal, Special Issue on Universal Enterprise Integration, 6(1):61–87

Chen H, Finin T, Joshi A (2003) An Ontology for Context-Aware Pervasive Computing Environments. Knowledge Engineering Review, Special Issue on Ontologies for Distributed Systems, 18(3):197–207

Dey AK, Salber D, Abowd GD, (2001) A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction, 16(2-4):97–166

Dockhorn Costa P, Ferreira Pires L, van Sinderen M (2005) Designing a Configurable Services Platform for Mobile Context-Aware Applications. International Journal of Pervasive Computing and Communications (JPCC), 1(1):27–37

Huriaux T, Picard W (2005) DynG: a Multi-protocol Collaborative System. In: Funabashi M, Grzech A. (eds) Challenges of Expanding Internet: e-Commerce, e-Business and e-Government, Proceedings of the 5th IFIP International Conference on e-Commerce, e-Business, and e-Government (I3E 2005), Poznan, Poland, Springer, pp. 591–605

Kersten GE, Strecker SE, Lawi KP (2004) Protocols for Electronic Negotiation Systems: Theoretical Foundations and Design Issue. In: Bauknecht K, Bichler M, Pröll B (eds) Proceedings of the 5th Conference on Electronic Commerce and Web Technologies (EC-Web04), Sarragoza, Spain, IEEE Computer Society, pp 106–115

Müller R, Greiner U, Rahm E (2004) AGENTWORK: A Workflow-System Supporting Rule-Based Workflow Adaptation. Data and Knowledge Engineering, Elsevier, 51(2):223–256

Picard W (2005a) Towards Support Systems for Non-Monolithic Electronic Negotiations. The Contract-Group-Message Model". Journal of Decision Systems, Special Issue on Electronic Negotiations - Models, Systems and Agents, 13(4): 423–439

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

Picard W, Huriaux T (2006) DynG: A Protocol-Based Prototype for Non-monolithic Electronic Collaboration. In: Shen W, Chao KM, Lin Z, Barthès JPA, James A, (eds) CSCW in Design 2005, LNCS 3865, pp. 41–50