Social Software for Modeling Business Processes

Agnes Koschmider[1], Minseok Song[2], and Hajo A. Reijers[2]

[1] Institute of Applied Informatics and Formal Description Methods
Universität Karlsruhe (TH), Germany
koschmider@aifb.uni-karlsruhe.de
[2] Department of Technology Management
Eindhoven University of Technology, The Netherlands
{m.s.song,h.a.reijers}@tue.nl

Abstract. The aim of this paper is to show how the use of social networks may help users to behave as modelers they trust. Users are guided in this respect within the context of an existing Recommendation-Based Process Modeling Support System to which "social" features are added. Two kinds of social networks are used to this end: (1) a social network from a process model repository and (2) a social network from a recommendation history. The social network from *process models* provides an organizational view of business processes. An example of the information that could be derived from such a network is the average distance between performers who belong to part of business process that is already modeled and the ones who belong to a candidate process. A user can apply this result to complete a process model in a way that is similar to earlier selected solutions. The social network from *recommendation history* shows the relationship among modelers who use the recommendation system. From its usage history, social networks can be generated that express the similarity between its nodes (users). Both approaches are presented as effective ways to exploit social relationships in capturing business processes in conceptual models, one of the key activities in the BPM domain.

1 Introduction

By providing people personalized recommendations and services, recommender systems help people facing the challenge of dealing with today's information overload. Various types of such systems can be distinguished. A *content-based* recommender system [1] suggests an item to a user based upon a description of the item and its interests in the past. A *collaborative-based* recommender system [2] predicts what a user wants based on what she and people with similar preferences liked in the past. A specific system relevant in the BPM area is the recommendation-based modeling support system as proposed in [3]. The system suggests process fragments to modelers taking into account their modeling intention as derived from the user's interest and patterns observed in other peoples' preferences. To be more concrete, this recommendation-based modeling support system offers a query interface and an automatic recommender component, which predict fitting process model fragments suitable for completing an

D. Ardagna et al. (Eds.): BPM 2008 Workshops, LNBIP 17, pp. 642–653, 2009.

edited business process model. The modeling support system works on a repository which may contain business processes of an enterprise or of a group of users working in the same field.

Despite the advanced state of this recommendation-based modeling support system, the current version displays several points that could be further improved. (1) Up to now we did not integrate any trust mechanisms into the recommendation process. Heider's famous "balance theory" [4] suggests that individuals are more prone to interact with friends of friends than unknown peers. Hence, following this line of argument users may also be willing to select recommendations that are based on decisions by known (skilled) persons in contrast to unknown ones. (2) Additionally, there is no functionality for change propagations. The application of the recommendation-based modeling support system is suitable for standard processes (order or claims processing) or standard process parts. Usually, users merge different process parts to one business process model or select one process model and then customize it to their needs. Thus, when a user modified a process part (e.g. due to new security restrictions) it would be helpful that the system informs users whose models may be affected by this modification.

To provide solutions for trust and change propagations we will generate a social network from process models and past preferences of users. Our approach relies on the recorded history of the system's usage, which consists of the name of the selected recommendation and the performer (user that selected a specific recommendation). When a user designs a new process, the design history of others who are close to the user is exploited. In summary, a social network that is derived from existing process models and recommender history may be used to the following benefits:

1. the system gathers persons sharing same interests or supports strategic collaboration,
2. the extended recommendation-based modeling support system encourages user trust behavior and participation by those users who are unskilled, because users gain insight who already selected specific processes,
3. the system propagates process changes to users involved in the network.

The first benefit can be gained from the social network from process models and the second and third can be obtained from the social network from recommender history. The generation of social network structures will be explained in detail in Sections 4 and 5. Before doing this, we first summarize related works and sketch the functionality of the recommendation-based process modeling support system in Section 3. Finally, our paper ends with a conclusion and a reflection.

2 Related Work

The study of social network analysis has attracted the interest of scientists from different fields such as: (1) sociology [5], (2) psychology [6], (3) economy [7] and computer science. In the computer science field the works related to our approach are using social networks as a method for communicating trust or distrust [8], [9], propagating changes [10] or analyzing interpersonal relationships in an organization [11]. Considering the recommender system for process modeling several authors proposed methods for process model reuse [12],[13] but yet with little impact on the modeling context and user requirements. None of these solutions proposes techniques for adequate suggestion of processes to be reused during process modeling regarding user's modeling intentions. In this paper we will generate a social network from process models and past recommendations by retrieving the user history in a recommendation-based modeling support system.

3 Recommendation Based Process Modeling Support

A recommendation-based process modeling support system [3] suggests users fitting processes to achieve a modeling goal regarding the user's modeling intention and modeling history of a community of users. This recommender system implementation embeds two types of modeling support (1)a query interface and 2)a recommender component) in order to achieve the user's modeling goal.

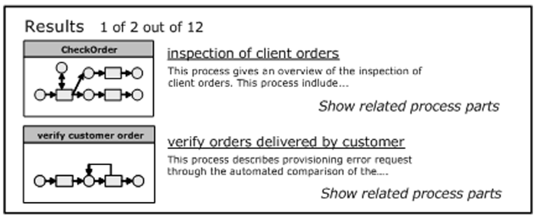
The query interface allows users to request processes or process parts that are of interest to them. The query interface uses Lucene's query parser syntax[1] and users can enter six query arguments. (1) Title: referring to names of process elements, (2, 3) First Element/Last Element: searching for a specific first/last element(s) in the process, (4) Property: filtering the repository by a goal of the process model (*standard* restricts the search to well-established and certified processes), (5) Model: referring to models fulfilling one of the four modeling purposes such as analysis, documentation, execution or reengineering,(6) Objective Description: searching for processes fulfilling an objective. This data is annotated by users.

The recommender component proposes appropriate process parts, which fit to an currently edited business process. The user can invoke the recommender component by highlighting the corresponding element group to be completed by process reuse. The recommender component should be used if the user is not sure how to complete the process. In this case the results from the query can be unsatisfying due to the user's vague intention of the process model.

Processes that meet user's criteria are displayed first in a table-based result list ordered by element name match score, the frequency a process has been selected/reused by another user and the number of average operations (deletion, insertion) made when selecting a recommendation. Next, when double clicking on recommendation(s) in the table-based view a graph-based visualization of the process will be opened as shown in Figure 1.

In Figure 1 the user can preview related process parts for each recommendation (see *Show related process parts*). The idea is that process parts that succeed or precede the part in question and were used in the same modeling domain the user is in at the moment (e.g. Manufacture) can help to estimate the degree of fitness of a recommended part.

[1] `http://lucene.apache.org/java/docs/queryparsersyntax.html`



Results 1 of 2 out of 12

inspection of client orders

This process gives an overview of the inspection of client orders. This process indlude...

Show related process parts

verify orders delivered by customer

This process describes provisioning error request through the automated comparison of the....

Show related process parts

Fig. 1. Graph-based visualization of fitting recommendations

To encourage user trust behavior and participation by those users who are unskilled we extended the table-based result list by a network information. This remark informs the user about the number of persons belonging to a network and selected this recommendation. Table 1 shows our extended table-based representation of fitting suggestions.

Table 1. Extended Table-based Result List with Network Remarks

#	Process Name	Score	Frequency	Operations	Network
1	CheckOrder	95.02	5	15	7 pers.
...	...	...	...	...	...
10	Handle Customer Order	48.85	3	20	2 pers.

By a right mouse click (in the network column) the user can open two network structure, which were generated from process models or recommendation history. The first network structure allows users to view and contact related persons regarding a strategic collaboration. The second network structure enables to propagate changes across "clique" members and support reusing modeling history of "neighborhoods" in order to faster complete an edited process model. In the two next sections we will explain how to generate a social network from process models and the recommendation history.

4 Social Network from Process Models

This section discusses the use of social networks from a process repository. The social network from process models provides an organizational view of business processes. When an user queries possible candidate process models, the result could show the related social network and analysis results.

In this section, we will explain the method of generating a social network from process models in Section 4.1. Then we will explain how to use the social network for the recommendations in Section 4.2.

4.1 Deriving Social Network from Process Models

To derive social networks from process models, we can consider three types of metrics defined in [14]. They are *transfer of work*, *subcontracting*, and *cooperation.*

The transfer of work metrics and the subcontracting metrics take into account causal dependencies (i.e., based on ordering of activities) among performers. The causal dependency can be defined as follows. Note that, we first define a process model including performers.

Definition 4.1. (Process model) A process model, PM, is a 5-tuple (P, T, F, R, π) where

(i) (P,T,F) is a WF-net [15],

(ii) R is a set of performers,

(iii) $\pi : T \rightarrow R$, i.e. performer (R) assignment for a task (T).

Process model extends Workflow nets(P,T,F) with performers (R). The models in the process repository are represented in terms of a Petri net. Activities are modeled by transitions and casual dependencies are modeled by places and arcs. Performers related to activities are specified above transitions.

From the process model, we can define the causal dependency as follows.

Definition 4.2. (*causal dependency*, $\Rightarrow$) Let $PM = (P, T, F, R, \pi)$ be a process model. For $t_1, t_2 \in T$, $t_1 \Rightarrow t_2$ if and only if *path*$(t_1 \rightarrow t_2)$ is elementary and $(t_1, t_2) \in F^2$. [2]

If t_1 and t_2 have a causal dependency, t_1 is followed by t_2 in the process model.

Within a process model there is a transfer of work from performer i to performer j if there are two subsequent activities where the first activity is assigned to i and the second activity to j. Considering the frequency of transfers, we can calculate the weight values on arcs between performers. For example, we can define the following formula.

- $r_1 \triangleright_{PM} r_2 = \sum_{p \in PM}(\mid r_1 \ \Rightarrow_p \ r_2 \mid) / \sum_{p \in PM} \sum_{i,j \in p} \mid i \ \Rightarrow_p \ j \mid$

$r_1 \triangleright_{PM} r_2$ denotes the total number of transfers from performer r_1 to r_2 in process models divided by the total number of transfers in the models. Using such formula, we can derive a social network among performers, which shows the transfer of works between them.

We can also derive social networks considering *subcontracting* and *cooperation.* Subcontracting considers the number of times performer j executes an activity in-between two activities executed by performer i. This may indicate that a work was subcontracted from i to j. The cooperation ignores causal dependencies and simply counts how frequently two performers participate in activities of the same models. The more often two organizational units work together, the stronger their relation is.

4.2 Application of Social Network

We assume the process repository contains five process models as shown in Figure 2. The user is not sure how to continue her edited business process. She

[2] *path*$(x \rightarrow y)$ if and only if there is a path of nodes in the graph corresponding to $(P \cup T, F)$. If R is a relation, then $R^n = \{(a_1, a_3) \in A \times A \mid \exists_{a_2 \in A}(a_1, a_2) \in R^{n-1} \wedge (a_2, a_3) \in R\}$ and R^* is the transitive closure.

invoked the recommender component in order to be supported by the recommender system. In the extended table-based result list (see Table 1) she right-clicked on the first network structure alternative and thus the social network opens as depicted in Figure 2.

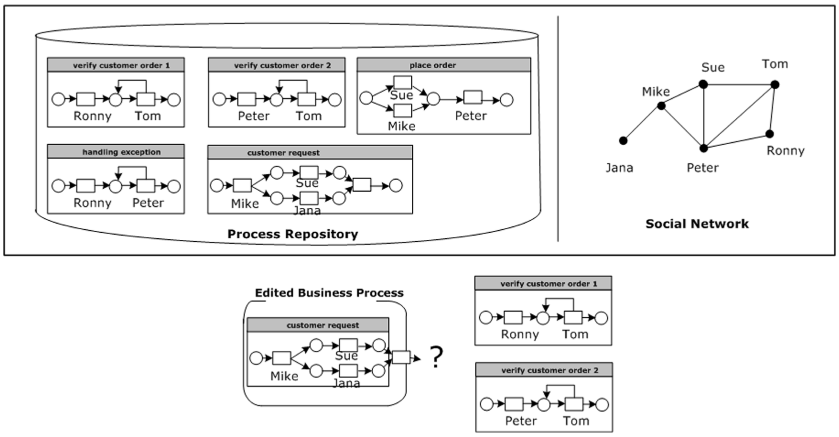


Fig. 2. Social Network from process models

In this figure each process model in the repository contains information on performers. We derived the network depicted in the figure, by considering the transfer of work between performers. In the network, the nodes refer to the performers in the process models and the arcs show the handover of work between the performers. For example, since the work is transferred from "Ronny" to "Tom" at the "verify customer order 1" process, we add an arc between "Ronny" to "Tom" in the social network.

If the user is not sure which recommendation to select after customer request ("verify customer order 1" vs. "verify customer order 2" which have the same process structure, but are performed by different performers) she can consult the social network. If we take into account the connections between the performers (i.e. "Mike", "Sue", "Jana") in the existing process (i.e. "customer request") and the two performers (i.e. "Ronny", "Peter"), there is no direct connection to "Ronny", but two connections to "Peter" (i.e. "Sue" to "Peter" and "Mike" to "Peter"). Thus "verify customer order 2" process could be more attractive to the user, since the performers are more tightly connected. The user can combine the edited business process with "verify customer order 2" and save it as a new business process. Note that, we use the name of performers in the process models and the social network in the example to make it easy to understand. Of course, in real process models, a group of people such as roles, department, etc. is assigned to a task. And nodes in a social network also represent groups of people.

5 Social Network from Recommendations

This section discusses the use of social networks from a recommendation history, which consists of the name of the selected recommendation and the performer[3]. In Figure 3 the user selected the process part *customer request* (edited and selected by Jane and Peter) and the direct succeeding process part *verify customer order* (edited and selected by Anna, John and Jim). When right-clicking on the second alternative in the extended table-based result list (Table 1) the user can choose between three variants of the social network visualization in order to faster select a fitting recommendation. (1) According to the modeling purpose. (2) According to process names, which has been chosen in this figure. (3) Regarding preceding and subsequent processes. In this section, we will explain the method of generating a social network from a recommendation history in Section 5.1. Then we will discuss the application of the social network for the recommendations in Section 5.2.

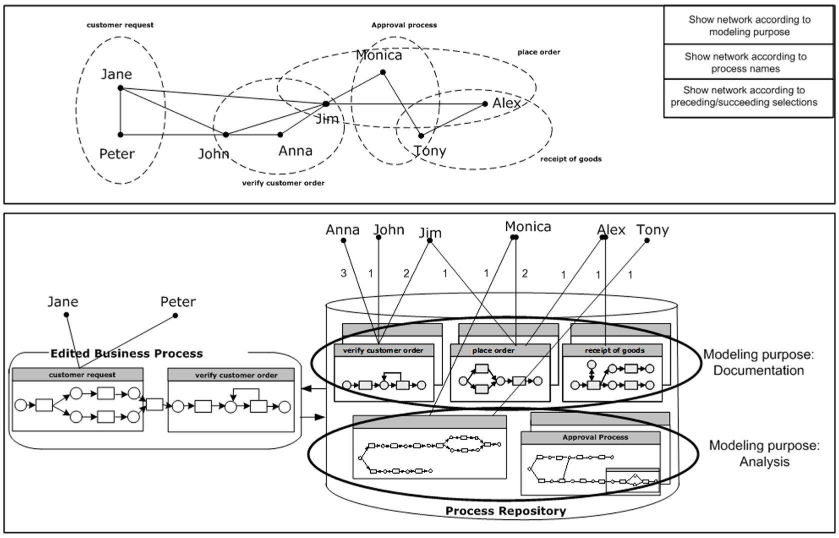


Fig. 3. Social Network according to recommendation names

5.1 Deriving Social Network from Recommendations

To derive social networks from the recommendations, we use the user history. Table 2 shows an example of the user history generated from the modeling history of the community of users. In the table, each row refers to a user (u) and a column corresponds to a process model (p) in the repository. And each

[3] Note the different consideration of users. In the previous section we took into account the transfer of work between performers in order to generate a social network. In this section we use the user's selection for social network generation.

cell (c_{ij}) shows the number of use of the process model (p_j) by the user (u_i). From the matrix, we can measure the distance between two users by comparing the corresponding row vectors. We can use several distance measures such as *Minkowski distance*, *Hamming distance* or *Pearson's correlation coefficient* [11]. Using the distance measure, we can generate the distance matrix among users. The distance matrix can be transformed into a network form. And from the network, by applying a certain threshold value, we can remove unimportant arcs and generate a social network among users. If two users are connected in the network, it means that the two users use a similar set of process models during their previous process designs.

Another way to generating a social network is considering the order of inserted process parts. The recommendation system stores (for each modeling purpose e.g. analysis, documentation or execution) the order of inserted process parts into the workspace of the user. Table 3 shows an example. In the table each row refers to an operation of a user. For example, $user_1$ uses process P_1 and process parts P_2 and P_1 is located before P_2. $User_2$ uses the same process parts (i.e. P_1 and P_2), but the order is different from that of $user_1$ (while the modeling purpose is the same).

From the table we can generate a user by preceding/succeeding table depicted in Table 4. The table shows how frequently users use a certain order of process models. Each cell represents the number of use of an order of processes ($P_j \rightarrow P_k$) by a user (u_i). From the table, we can generate a social network for users by calculating the distance between users. Note that the distance metrics mentioned above can also be used. The third way of generating a social network is taking into account the insertion history of process models. The recommendation system stores the order of inserted process parts into the workspace of a user. A user of the recommendation system can generate a new process model and insert it into the repository. A user can search for suitable process models and generate a new model by combining them. Table 5 shows an example of insertion history. In the table, $user_1$ generates "customer order" process by merging "check order

Table 2. User History

	P_1	P_2	P_3	P_4	P_5	...	P_N
$user_1$	2	1	0	0	0	..	2
:	:	:	:	:	:	:	:
$user_M$	0	0	1	0	0	..	0

Table 3. Preceding/Succeeding History

	Preceding process	Succeeding process
$user_1$	P_1	P_2
$user_2$	P_2	P_1
$user_1$	P_1	P_4
:	:	:

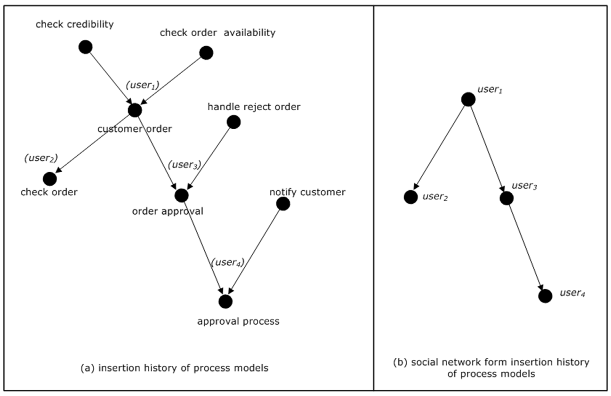
Table 4. User by Preceding/Succeeding Table

	$P_1 \rightarrow P_2$	$P_1 \rightarrow P_3$	$P_1 \rightarrow P_4$	...	$P_{N-1} \rightarrow P_N$
$user_1$	2	0	1	..	2
$user_2$	0	1	0	..	2
:	:	:	:	:	:
$user_M$	0	0	1	..	0

Table 5. Process Insertion History

	used processes	inserted process
$user_1$	check order availability + check credibility	customer order
$user_2$	customer order	check order
:	:	:

availability" and "check credibility". A user can also take one process model and modify it to make a new process model. In the table, $user_2$ modifies "customer order" and generated "check order". From this information we can generate a social network reflecting the history. Figure 4 shows the social network derived from the table. Figure 4(a) shows a graphical view of the insertion history from the table. In the figure, nodes represent process models and arcs show relationships between process models. For example, since the "customer order" is designed with "check credibility" and "check order availability", there are arcs from "check credibility" to "customer order" and from "check order availability" to "customer order". From this figure, by replacing the process model with the user who generated it, we can derive the social network depicted in Figure 4(b) In this section, we explained three ways of deriving social networks from recommendation history. The next section focuses on the application of the generated social networks.



(a) insertion history of process models

(b) social network form insertion history of process models

Fig. 4. Generating social network from insertion history

5.2 Application of Social Network

This section explains application of social networks from recommendation history, which is shown in Table 6. Whenever a user logs in to the recommender system she needs to specify the name, the purpose, the objective, the description and the property of the model. Table 6 shows a recommendation history where 10 users were involved and selected a total of 13 process models. The purpose of the models was documentation. From this history, we derived a social network shown in Figure 5. We applied the threshold value of 0.3 and removed arcs of which weight value is less than it. There are five cliques in the network. They are $\{user_1, user_2\}$, $\{user_3\}$, $\{user_4, user_7, user_9\}$, $\{user_5, user_8\}$, $\{user_6, user_10\}$. The user that is directly connected to a certain user is the neighborhood of the user, who has similar process usage patterns. Thus, when the user designs a new process model, she can use the history of her neighborhoods. For example, in the figure, the history of $user_4$ is used in the recommendation system for $user_7$.

Table 6. Realistic User History (P_1=check Order2, P_2=check Order3, P_3=order approval, P_4=Order Checking, P_5=CheckOrderAvailability, P_6=order_approval, P_7=handle rejected order, P_8=manage order approvals, P_9=customer order, P_{10}=check credibility, P_{11}=checked order availability, P_{12}=validate Order, P_{13}=approve order)

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}
$user_1$	0	0	0	0	1	0	1	2	0	1	0	0	0
$user_2$	0	0	0	1	0	0	1	2	1	0	0	0	0
$user_3$	1	0	0	0	0	0	0	0	0	0	0	0	2
$user_4$	0	0	1	1	0	0	0	0	1	0	1	1	0
$user_5$	0	0	1	0	0	0	0	0	0	1	0	1	1
$user_6$	1	0	0	1	0	0	0	1	0	0	0	1	0
$user_7$	0	0	1	0	0	1	0	0	0	0	1	1	0
$user_8$	0	0	1	0	0	0	0	0	0	1	0	1	1
$user_9$	0	1	1	2	0	0	0	0	0	0	0	1	0
$user_{10}$	1	0	0	0	0	0	0	0	0	0	0	1	0

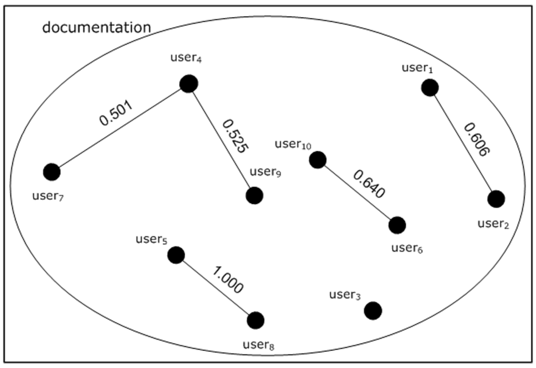


Fig. 5. Social Network from Table6

There are two more ways of generating social networks in Section 5.1. Social networks from the order of process models can be used to find a neighborhood of a user. For example, Table 6 can be extended including the history of the order of process models and a social network can be calculated from the combined table. Social networks from insertion history cannot be used to find neighborhoods, but for change propagation. Whenever one of the members of the clique changes a process the other neighbors, who are affected by this modification, will be informed about this change. This notification is optional and will be performed only if users activated this functionality.

6 Conclusion and Reflection

Process modeling is an important activity that supports all phases of the BPM life-cycle, from the early identification of a process through subsequent phases of diagnosis, re-design and beyond. There is much to gain in supporting users to come up with better models in a shorter time-frame. This is the main motivation for the development and enhancement of the central artifact in this paper, the recommendation-based modeling support system. We described in the previous sections how this system can be enhanced by exploiting the information that can be derived from historic usage patterns. In particular, social networks can be generated from such logs, much in the spirit of our earlier work [11], but this time with the specific intent to guide users towards a selection of process fragments that resembles earlier decisions of their peers.

The proposed approach fits within the broader advancement and development of social software systems and neatly addresses the realms of sociality that are seen as characteristic for social software [16]. In particular, our enhanced system relates to a *realm of sociality* where process modeling is seen as an act that is better performed collaboratively. It supports an *enabling practice* of people building on the modeling experiences of others. User of the system *build on their identity* by helping to extend a repository of high-quality models with elements of their own. Finally, the system may help them in *actualizing themselves* by allowing them to grow in their modeling proficiency and receive recognition for this in terms of reuse of their decisions and models. We believe this to be a promising development of bringing introducing social software to the BPM area.

References

1. Balabanović, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Commun. ACM 40, 66–72 (1997)
2. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22 (2004)
3. Hornung, T., Koschmider, A., Lausen, G.: Recommendation based process modeling support: Method and user experience ((to appear)). In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231. Springer, Heidelberg (2008)
4. Heider, F.: The Psychology of Interpersonal Relations. Wiley, Chichester (1996)

5. Watts, D.J.: Networks, dynamics and the small-world phenomenon. American Journal of Sociology 105, 493–527 (1999)
6. Kenny, D., LaVoie, L.: The social relations model. American Journal of Sociology 105, 141–182 (1984)
7. Tichy, N., Tushman, M., Fombrun, C.: Social network analysis for organizations. Academy of Management Review 4, 507–519 (1979)
8. Ziegler, C.N., Lausen, G.: Propagation models for trust and distrust in social networks. Information Systems Frontiers 7, 337–358 (2005)
9. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proceedings of WWW 2004, pp. 403–412. ACM, New York (2004)
10. Mason, W.A., Jones, A., Goldstone, R.L.: Propagation of innovations in networked groups. Journal of Experimental Psychology (in press)
11. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering social networks from event logs. Computer Supported Cooperative Work 14, 549–593 (2005)
12. Kim, J.H., Suh, W., Lee, H.: Document-based Workflow Modeling: a Case-based Reasoning Approach. Expert Syst. Appl. 23, 77–93 (2002)
13. Lu, R., Sadiq, S.W.: On the discovery of preferred work practice through business process variants. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 165–180. Springer, Heidelberg (2007)
14. Song, M.: Organizational mining in Business Process Management. PhD thesis, Pohang University of Science and Technology, Pohang, South Korea (2006)
15. van der Aalst, W.M.P.: The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers 8, 21–66 (1998)
16. Bouman, W., Hoogenboom, T., Jansen, R., de Bruin, B., Huizing, A.: The realm of sociality: Notes on the design of social software. Working Paper, 1–24 (2008)