Social-PPM: Personal Experience Sharing and Recommendation

Anne H. H. Ngu
Department of Computer Science
Texas State University
San Marcos, Texas, USA
Email: angu@txstate.edu

Shawn Fang
Department of Computer Science
University of Texas, Austin
Austin, Texas, USA
Email:qfun@utexas.edu

Helen Paik
School of Computer Science and Engineering
University of New South Wales
Sydney, NSW, Australia
Email: hapik@cse.unsw.edu.au

***Abstract*—The rise in popularity of various social network applications has brought the opportunities for Internet users to share and reuse a plethora of things like images, videos, datasets, ideas, interests, reviews etc. However, currently there is no effective way of sharing personal experiences such as the process of filing a personal income tax return or applying for a visa. We propose a social-aware process model and its implementation as a social network application that empowers users to create, to execute, and to share personal experiences within a social network at anytime and anywhere. As a social-aware process management system, it is important to have an effective recommender that predicts personal processes that a specific user may not be aware of and yet have the opportunity to enhance his/her life experiences. We adapt an existing collaborative filtering algorithm with emphasis on social context, the interaction history, and the type of interactions in processes for effective process recommendation. The assumption is that if two users have been copying and following same processes then it is likely that those two users have similar life goals, and this would be reflected in their future usage of the system as both of them will engage in similar processes.**

***Keywords*-Personal Process Management; Process Crowd-sourcing; Process Recommendation; Socializing processes**

I. Introduction

The rise in popularity of various social network applications has brought the opportunities for Internet users to share and reuse a plethora of things like images, videos, datasets, interests, ideas, traffic information, reviews etc. There are many different types of social network applications designed to encourage the exchange of ideas and the establishment of a community. However, none of them encourage the tracking and exchange of personal processes. While the rise in the sharing of video clips in Youtube, Snapchat, Facebook and Instagram do compromise personal experiences, those video clips do not document the insight and the context behind the shared personal processes and cannot be reused effectively.

In the increasingly connected digital world, users carry out personal processes on a daily basis using a variety of applications or resources on the web and mobile phones. However, the knowledge gained by users while carrying out the repetitive routines is locked away (in a person's email folder, personal notebooks, etc.) and is inaccessible to others. Even though we have 21st century communication infrastructure, we have no easy way to reuse or share our knowledge about how we do what we do. For example, what do you do normally to apply for admission to a PhD program at Texas State University in USA? Existing mechanisms of sharing personal experiences lack a process model that can document personal experiences in a structured and collective manner. So while personal processes may be available, albeit not in an obvious manner, via existing applications, none appear to be reusable in a collective learning environment.

We define a personal processes as a list of tasks that a user must perform in a certain order to achieve a goal. It can be as simple as obtaining an Internet domain name for a small business or as complicated as applying for a driving license in the state of Texas. Unlike business processes which are structured and schema-based, personal processes are ad-hoc to the point where each process may have its own unique structure. Personal processes are thus much more flexible than business processes in the order of how tasks are scheduled and executed. For example users can add or remove tasks in personal processes as the situation demands. After the completion of a task, users can write a reflection or comment on how the task went, attach pictures, video or documents relevant to the tasks and share that with people in his/her social network. This is different from a business process which is only interested in recording when or whether a task has been executed successfully.

We believe that the lack of a systematic approach for users to create, share, and reuse personal experiences in a collective learning environment is hindering our day to day productivity and unnecessarily preventing the accumulation of "crowd wisdom" that can benefit a virtual community. A recent experiment [1] also strengthens our hypothesis that the ability to share personal processes in a structured manner saves time for subsequent users, as it means that users are not required to conduct their own time consuming research on each process to derive the associated tasks and their contexts. In this paper, we propose a social-aware process model and recommendation that can document personal experiences in a structured way which facilities reuse and collective learning.

Shawn Fang is an REU participant in the Texas State REUIOT program

Our goal is to develop an innovative platform for personal process documentation and sharing that is accessible via any mobile devices. Through automatic documentation and sharing of personal experiences, we can avoid costly or inefficient steps in executing routine personal processes. Our vision for managing personal process requires the development of the following components:

- A mobile client that supports the highly exploratory process creation and execution (e.g. tasks can be added, deleted, re-ordered, changed and executed with selected resources)
- An intuitive way to query and search personal process that allows users to maximize their productivity (e.g. tasks that must be done in the same location, tasks that need to be completed at certain time-frame, tasks that can done with minimal cost at certain time)
- Sharing of intermediate process execution results with others and receive feedback from them while carrying out the process.
- A personal process recommendation that leverages contextual knowledge and social relationships between users
- A 21st century mechanism for passing accumulated personal experiences to others without explicit manual documentation.

Our earlier paper [2] describes the general architecture of our social aware personal process management system that supports creation and execution of personal processes. In this paper, we discuss the need for a structured personal process model and the robust process recommendation system that encourages reuse of processes from users with similar "behavior" (i.e, followed or copied similar processes). This is known as collaborative recommendation.

The remainder of this paper is organized as follows. Section II describes the related work, Section III provides the overview of the system, Section IV outlines the personal process model, SectionV describes the application, Section VI discuss the recommendation algorithm, Section VII outlines the experimental results. Finally, Section VIII provides some concluding remarks and also discusses various open issues.

II. Related Work

Relevant work in personal process management can be divided into three types. The first type is the existing mobile applications available from Google Play and AppStore. The second type is the various online forums such as wikiHow and eHow and the third type is the published literatures in this area.

We downloaded and tested highly rated free mobile apps on the market that allowed users to organize tasks. Mobile apps that we tested include: MyLifeOrganized[1], Wunderlist[2], GTasks[3], Mindjet Tasks[4] and Todoist[5]. All of those apps, which usually identified themselves as to-do list managers, are simply fancy text editors where the user can write down tasks she needs to do. In the best case, few of those apps offered a reminder functionality and allowed the user to keep track of her progress in her to-do list. Mindjet Tasks extended the functionalities to also manage to-do list for a team of collaborators. None of those apps offered the possibility of reusing and sharing the process/experience of completing the to-do list. In other words, each user is at her own and there is no way of transferring knowledge about common processes from one user to another without tedious documentation.

wikiHow [3] is a web service that sources the crowd to contribute advices on how to do certain things in the same spirit of wikipedia. The topics addressed in wikiHow can range from how to drink more water everyday to how to become more social. Each topic is a textual description on how to accomplish a certain thing written by a volunteer. It does not describe any specific personal experiences in achieving a goal. In particular, the contextual information associated with processes, such as personal reflection or comments on whether a task has a time, location, cultural context, organization, or specific resource dependency is not described. It is not possible for other users interested in the similar process to glean from the wikiHow description on why a particular way of achieving a goal is better. eHow uses a blogging paradigm rather than a mass collaboration approach to achieve similar goals.

The earliest academic research work on personal process management is described in [4]. The authors recognized the need to have a personal process model which is not constrained by rigid control and structural rules as defined in business process management. They proposed a set of algebraic operators that can be used to query a repository of processes for recommendation of suitable processes targeted to a task at hand or a specific input/output artifact. However, the users is restricted to process templates pre-created by the system. Moreover, there is no notion of sharing, recommendation, reuse, following, and notification.

A series of work produced by the authors in [5], [6], [7] defined and implemented the personal process management concepts from Web service composition view points where the focus is on formally and concretely representing sequential and conditional constraints in a personal process so that it is ultimately executable. In particular, [7] presented a language named PPML (Personal Process Modelling Language), and used it to formally describe personal processes with the required input, and the expected output. Again, personal processes in PPML-based system must be gen-

[1] www.mylifeorganized.net
[2] www.wunderlist.com
[3] GTasks: To-Do List & Task List, Google Play
[4] www.mindjet.com/products/mobile
[5] www.todoist.com

erated from templates created by expert process designers before others could use it. It is thus not a tool that can empower end-users to create and share variety of personal processes in a large scale. The similar work described in [6] aims to solve the end-user process creation, execution and sharing, but the applications remain restricted to a domain expert, not designed for everyday processes. [8] described a simplified Business Process Management (BPM) model that paid special attention to the role of social aspects of the process management such as sharing and assigning tasks. However, this work remain at preliminary level and it did not address the importance of recommendation to learn about new processes that could be valuable for achieving goals that users are not aware of. [9] proposes a graph-based personal process model which aims to provide a generic description language for personal 'how-to' instructions. The work does not yet support the integration of the model to the execution or sharing environment.

There have also been some attempts in recent years to accommodate social features in the Business Processes Management (BPM) environment. Most notably a recommendation-based process modelling support system with social features in [10]; a modelling and execution tool for business processes with collaboration and wiki-like features embedded in [11]; and an ad hoc workflow system focusing on non-intrusive capturing of human interactions in [12]. However, these systems aim at recommendation of task elements of a process during the construction or modeling of a process not prediction of concrete processes that might enhance a user in accomplishing goals that he/she is not aware of. The HelpMeOut [13] is a social recommendation system that aids novices with the debugging of compiler error messages and runtime exceptions by suggesting successful solutions to similar errors that other programmers have encountered. The main hypothesis behind HelpMeOut is the belief that presenting relevant solution examples will make it easier for novices to interpret and correct error messages. The relevant solution examples bring the contextual information how others are approaching the same problem. This is similar to our goal of recommending and sharing of concrete processes rather than just abstract description as presented by systems like wikiHow and eHow.

The work by Matejka et. al. in [14] describe a recommender system that suggests AutoDesk commands to users. They demonstrated that item-based collaborative filtering with domain specific rules is effective in recommending new commands to users. Processes are not characterized by a fixed set of features like a movie or a book. The perceived value of a process depends a lot on the context under which it is being used and the history of interactions on the process. We adapt a collaborative filtering algorithm with special emphasis on social context, the interaction history and the type of interactions in personal processes for effective personal process recommendation.

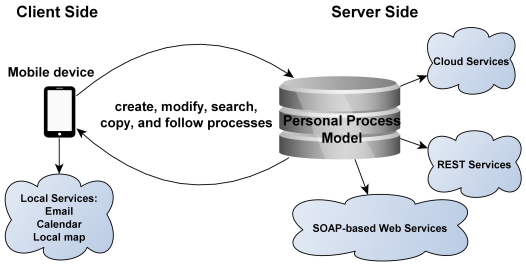

Figure 1. Overview of the System

III. Overview of Social-PPM

Social-PPM is a social network application that runs on a mobile phone with functionalities to (i) document and create personal process using a to-do list model, (ii) query and search personal processes that allows users to reuse the most relevant crowd sourced processes, (iii) leverage contextual knowledge and social relationships between users to recommend other personal processes that might improve user's productivity, (iv) encourage users to share the intermediate results of personal processes with others and receive impromptu feedback that can be used to adjust the current personal processes in real-time.

Figure 1 provides an overall architecture view of the system depicting the flow of data between the end-user, the server process model, and various other services that can be leveraged to complete a process. Upon successful registration in our system, users are given a workspace where they can use the tools provided by the system to create or manage their personal processes. For example, when a user is at a specific place and time, the system can alert users of what tasks can be accomplished at that time and place. If users need to create new processes, they can first perform a search and see whether there are any relevant processes that can be sourced from the community to be used as a template or guidance. If a relevant process is found, creating a new process is just a matter of doing a copy and modifying the copied process for the current situation.

IV. The Personal Process Model

In this section, we describe the model we use to represent the processes in Social-PPM. The model shares some commonalities with a related system called ProcessBook [15]. Both systems consider a personal process to be described as a simple task list template (i.e., tasks to be done). The word 'template' is used in the sense that it is intended to be customized. A *TaskListTemplate* is a pair $(\mathtt{G}, \mathtt{T})$, where:

- $\mathtt{G}$ is a statement of the *goal* in natural language,
- $\mathtt{T}$ is a rooted tree where the root is labelled with a unique identifier for the template. Each descendant

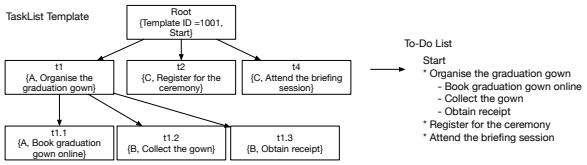

Figure 2. TaslListTemplate and To-Do List

node `t`, which represents an atomic task in the template, is a triple $(\texttt{id}, \texttt{label}, \texttt{properties})$, where `id` is the unique identifier of the task, `label` is a textual description of the task and `property` describes other characteristic of the task such as the person who first created it, the number of like and dislikes on the task as well as the set of available comments on the task.

This simple model allows flexible and agile representation of personal processes, which is necessary to support unstructured or ad-hoc processes. [16] utilizes similar but more restricted form of to-do list model based in a business operation management context to support knowledge intensive processes for the knowledge workers [17]. The idea of supporting flexibility in a process creates a less clear boundary between a process model and its running instances, as the model can be changed by the users during the execution (e.g., users can add a task, or ignore/delete a task). To capture and utilize these kinds of user activities in Social-PPM, we assume the following interactions between models and their instances.

In the beginning, a template is chosen (either by recommendation used in Section VI or by search), then a to-do list is created (see Figure 2) for the user. From here, the system tries to capture the activities of the user by modeling the following change operations on each task. There are two types of change operations: task planning operations, task execution operations.

First, for each task, in terms of its execution life-cycle, it goes through the following states (supported by the task execution operations).

- **Planning** Once a task is created in the To-Do list, by default, it will be placed in the planning state.
- **In-Progress** Once a task is selected and flagged (e.g., by tapping a command) that the task has been started, it is timestamped by the corresponding operation. One a task is in this state, it can be commented and followed by other users in the network.
- **Completed** A task moves to "Completed" state when a user indicates (e.g., by tapping the command again) the task has been done. The operation also creates a timestamp. When a task is in this state, the owner of the task can write a reflection or comments regarding the task.

These distinct states of a task allow for the system to track which task in the to-do list has been completed and in what order (indicated by the timestamps).

Second, the to-do list can also be changed by the user via the following set of explicit change operations. We refer to this as task planning operations as they allow the users to update the to-do list. For simplicity, we omit the operation parameters.

- **Insert()** a new task t can be inserted at a given position
- **Delete()** task t is deleted at a given position
- **Update()** the textual description of task t is updated

Of course, the combination of the insert and delete operations allows for the implementations of high-level change operations such as **Move()** or **Replace()**. These change operations are also logged with timestamps and the location the where the changes occur so that the system can learn how a particular task list template is customized by the user at run-time.

V. The Application

In this section we will briefly explain the key features of the application by describing different use cases in which the user will encounter while using the application. For this purpose, we will be using a hypothetical personal process of a user applying to a PhD program in the United State as an example and show what the user will go through when using this mobile application to create and manage this personal process. When the user logs in to the application, the system will present the list of processes that the user owns, a list of processes that the user is following and a list of recommended processes. The interface also presents three possible options which are: `Manage:` managing an existing process, `Delete:` deleting a current process, and `Search:` searching for a new process as shown in Figure 3.

Creating a New Personal Process: In order to create an "Apply for a PhD" process, the user starts by searching for a similar process in the system that may serve her purpose. She can copy the process if she finds the needed process in the system by clicking on the copy button and then assigning a name and description to it. If the user is not satisfied by the search results then she can design a process from scratch by clicking on the `create a new goal` in the search page. A new view will appear, allowing the user to supply a name, a description, and a category for the process as shown in Figure 3. Once the information is entered, the user can click on the `create` button, and the process will appear in the user's current process list.

Managing a Process: Managing a process is available from the application homepage as shown in Figure 3. The user can select the target process from the current processes list and then click on the `manage` button. The system will react by opening a new view (Figure 4), where the steps of the process will be displayed as a to-do list and the user can manipulate them using several buttons including:

- `up and down` buttons: allows the user to change the position of a task in the list. This re-ordering might be necessary due to some personal constraints.

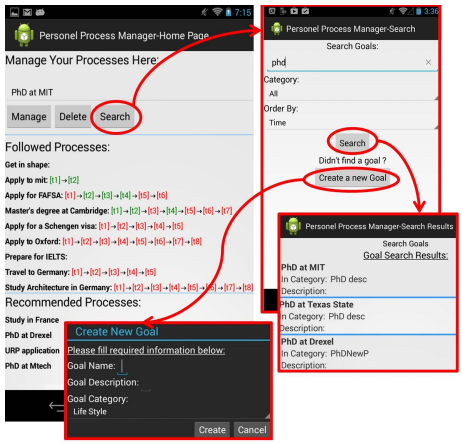

Figure 3. Creating a New Process

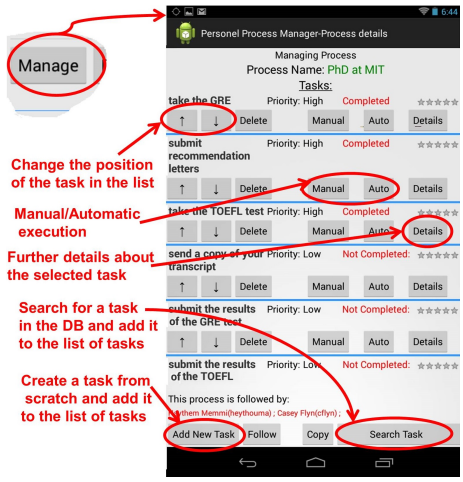

Figure 4. Managing a Process

- `Delete` button: deletes a task from the process. Some tasks in a process might not be relevant due to user's specific context. For example, if the applicant graduated from a university in United State, there is no need to take the TOFEL test.
- `Manual` button: allows the manual execution of a task, when pressed, it will change the state of the task to start. When pressed the second time, it will move the state of the task to 'complete' . This function helps to automatically document information such as the time and location of where a task is completed. User can also post a comment on how the task is completed and how much effort it takes to complete it by going to the `details` button.
- `Auto` button: allows the user to select a resource either locally on the mobile phone or remotely on the cloud in completing the task, and record that information. Currently, the only available local services are sending an email and adding an event to the user's calendar. This function helps to document how a task is being accomplished by software.
- `Details` button: allows the user to make changes in the task's name, description, and priority along with other functionalities such as view the location where the task was executed, add review/comments relative to the task, or comment on other people's reviews if it is a followed process.
- `Add New Task` button: allows the user to create a new task from scratch and add it to the current process.
- `Follow` button: allows the user to follow the displayed process.
- `Search Task` button: allows the user to obtained a list of recommended tasks in the repository and add the top ranked task to the current process's list of existing tasks.
- `Copy` button: allows the user to copy the current process structure for reuse.

VI. The Social Aware Process Recommendation

In a social aware personal process system with unlimited number of users and processes available, effectively and efficiently searching and selecting the most relevant processes is a crucial challenge. It is possible for users to perform a keyword based search to find processes that he/she can reuse. However, the current model of keyword search over the textual descriptions of processes is limited and will returns a set of processes that contains the keywords regardless of the semantics of the process. In the BPM area, many consider BPM models are graphs and apply a graph search algorithm or graph mining techniques to find 'similar' processes in the relevant process repository. Unlike BPM models which have terse task descriptions and highly varying structures, personal process models could contain longer and more descriptive task descriptions and relatively simple structures. So, comparing (graph) structures only may lead to a large number of 'structurally similar' personal processes, but semantically different processes.

Given the limited display of a mobile device, returning a large number of processes will result in information overload for the users. A facet search model [18] for processes can improve the situation but at the expense of taking a much longer time to explore the search space to find the relevant processes. Another solution is to recommend relevant processes to users based on their search goals. In the context of any social based application, the interactions initiated by users are not completely random and carry rich contextual information. We aim to provide a process recommendation service that can predict processes that the user may wish to pursue to improve his/her productivity based on social context. The algorithm for determining what processes might be interesting for a certain user relies on his/her interaction history with existing processes in the system, the types of interactions, and the user profile. For example if two users have been copying, commenting, and following the same

processes then there is a high probability that those two users have similar interests. This would be reflected in the system as both of them will engage in similar personal processes. If this is the case, our system will pro-actively spot processes that only the first user is using and suggest them to the second user or vice versa.

Our recommendation system is based on collaborative filtering to recommend a list of processes to a user. The final list of processes is based on a comprehensive score that sums up the score of voting and the score of rating for each neighbor process in a user's network. An action (copy, create, or follow) made by a user to a process represents a voting. For example, when a user copies a process, one full vote is assigned to that action. However, when a user follows a particular process, only half a vote is assigned to that action. We assume that the action of copying a process has more weight than the action of following a process. This is because the action of copying a process made by a user means the user is interested in the copied process to the extend that he/she wants to carry out a similar process by first owning a copy of that process. The overall rating score for a process is calculated based on the average of ratings over all tasks in a process. We will first define some of the terminologies used before we present the recommendation algorithm.

Definition of terms

- Target User is the user whom the recommendation system is targeting
- Target Process is a process that the target user copied/owned/followed
- Target Category is a category that a target process belongs to (e.g. we can have categories such as education, finance, travel etc)
- Neighbor User is a user who shares at least one common process with the target user
- Neighbor Process is a process that a neighbor user copied/owned/followed
- Neighbor Category is a category that a neighbor process belongs to

The recommendation algorithm can be divided into two parts, i.e. determination of the set of processes to be recommended, as well as the scoring and ranking of each process in the set.

Part 1: Determination of recommended processes
The purpose is to find the complete set of processes that the Target User U_i is predicted to copy or follow in the future. This prediction is based on the similarity of users in their actions. The assumption is: if two users have been copying and following the same processes then there is a high probability that those two users have similar interests, which would be reflected as both of them will engage in similar personal processes in the future. The prediction procedure is divided into three steps.

- **Step 1:** Find the set of processes the target user U_i copied, followed or owned, which we call $\{P_\alpha\}_i$, i.e. the set of processes associated with user i. This is known as Target Process set.
- **Step 2:** Find the set of users who owned, copied or followed any process in the Target Process set $\{P_\alpha\}_i$ excluding U_i itself, i.e. $\{U_j\}_i = \{U_j | U_j$ `owned, followed, or copied any process in` $\{P_\alpha\}_i$ `and` $\{U_j \neq U_i\}\}$. This is called the Neighbor User set.
- **Step 3:** Find the set of processes which are owned, copied or followed by any user in Neighbor User set $\{U_j\}_i$, excluding those that are owned, copied or have already been followed by U_i. This set is represented by $\{P_\beta\}_{\{U_j\}_i}$ $\{P_\beta\}_{\{U_j\}_i}$:= $\{P_\beta | P_\beta$ `is owned, copied, or followed by at least one user in` $\{U_j\}_i\} - \{P_\alpha\}_i$. The processes to be recommended to target user U_i all come from the set $\{P_\beta\}_{\{U_j\}_i}$. We call this Neighbor Process set.

Part 2: Scoring and ranking
The records in the Neighbor Process set $\{P_\beta\}_{\{U_j\}_i}$need to be ranked and presented to a target user U_i according to the order of scores which is used as the criteria for ranking. Our scoring system contains two basic schemes, i.e. voting score and rating score, which are combined as an comprehensive score eventually.

- **Voting score:** S_C. This score is computed in the following steps. For each process P_β in the Neighbor Process set $\{P_\beta\}_{\{U_j\}_i}$, for each user U_k in the Neighbor User set, $\{U_j\}$, compute the Jaccard similarity $J(U_k, U_i)$ between the users U_k and the target user U_i, which is performed as follows: a) Find the set of processes U_k owned, copied, or followed and the vote for each kind of action on each process i.e. $\{P_{\alpha'}\}_k$; and $\{P_{\alpha'}\}_v$; b) Compute the $J(U_k, U_i) = |\{P_{\alpha'}\}_k \cap \{P_\alpha\}_i|$ / $|\{P_{\alpha'}\}_k \cup \{P_\alpha\}_i|$, in which the "| | " operator means cardinality of. iii) $S_C = \sum_{U_k \in \{U_k\}_{\{P_\beta\}_{\{U_j\}_i}}} J(U_k, U_i)$ weighted by the number of votes.
- **Rating score:** S_R. This score is based on users' ratings on processes. Currently, in our system, there are no direct ratings for a process. However, each task in a process have attached reviews and rating. Therefore, the rating of a process is computed as the mean of available ratings of all tasks in this process.
- **Comprehensive score:** S_S. Both S_C and S_R are non-negative. Therefore, we can normalize the range of each score to [0, 10]. Hence, S_C and S_R becomes $S_C^{norm}$and S_R^{norm}, respectively. The comprehensive score $S_S = S_C^{norm} + S_R^{norm}$

.

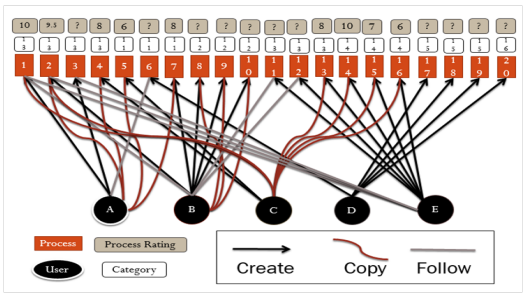

Figure 5. Data and Relationships

.

VII. Evaluation

A thorough user study of our application is our immediate future work. Here, we conducted a simple experiment to test the validity of our recommendation algorithm. We established five user profiles, twenty different processes of varying categories and rating of tasks, and the relationships between various users and processes. This is depicted in Figure 5. Based on the relationships between users and processes, the recommendation system computes a comprehensive score for each neighbor process of a target user and recommends the processes in the correct order to the target user as we anticipated. For example, a target user, person A created process 1 and 3, copied processes 2, 4, 5 and 7, and followed process 6. This implied that user A has a relationship to each of these processes and they should not be recommended to user A in our recommendation system. Since a process stores information about who has a connection to it, from the known processes 1-7 of user A, we find out that user B, C, D and E are the neighbor users with user A. Similar to user A, these other users have their own relationships to the processes. By finding the set of neighbor users of user A, the other unknown processes to user A but known to the other users will be filtered in. Also, based on the ranking algorithm described in Section VI, the processes with ranks greater than certain threshold will be recommended to user A. In this example, ranking of the processes from high to low is process 14, 13, 11, 12, 15, 16, 8, 9, 10, 17,18, 19 and 20. These are the predicted ranked processes in our recommendation system for user A and the actual recommended processes for user A are exactly the same as the predicted recommended processes.

VIII. Conclusion and Open Issues

We have proposed a flexible personal process model and its implementation as a mobile social network application that empowers users to create and to share personal processes at anytime and anywhere. We leveraged the most recent functionalities of social networks, including: a following system allowing users to keep track of the latest progresses and comments of personal processes created by other users, a notification system allowing users to be informed on major changes of followed processes and a recommendation system that predicts processes that a specific user may be interested in. In particular, the recommendation system enables users to learn about new processes that can be used to achieve goals that are oblivious to them. Once a user A shares some common processes with other users in the social context, all processes related to other users but excluding the known processes of the user A can potentially be recommended. However, the current recommendation system in PPM assumes the equal weight of voting (the type of interactions) and rating (the reviews) in the computation of comprehensive score. Further experiment needs to be performed to arrive at an accurate weight for voting and rating. The current recommendation algorithm also suffers from the "cold start" problem. For example, a new process which only has relationship to its creator will never be recommended to other users unless users search for it and the process appears in the search result. A content-based filtering leveraging the category that the process belongs to could solve this problem. Since every process must belong to a particular category, by computing content similarity between the newly created process and the existing processes in the same category, it is possible to derive initially the most likely Neighbor User set and this new process can be recommended to users in the neighborhood. We plan to incorporate this content-based filtering into our recommendation algorithm in the future. A comprehensive field study that includes the usability of the mobile social network application and the accuracy of the recommendation algorithm need to be further investigated.

Acknowledgment

We thank the National Science Foundation for funding the research under the Research Experiences for Undergraduates Program (CNS-1358939) at Texas State University to perform this piece of work and the infrastructure provided by a NSF-CRI 1305302 award.

References

[1] E. Lie and P. Rajan, "Towards personal process management," Undergraduate Honors Thesis, School of Computer Science and Engineering, University of New South Wales, Sydney, Australia, 11 2012.

[2] A. Ngu, H. Memmi, S. A. Hajimirsadeghi, H.-Y. Paik, and J. Shepherd, "Social-ppm: Social-aware personal process management," in *2nd Asia Pacific Conference on Business Process Management (AP-BPM)*, 2014, short Paper.

[3] "wikiHow: How to do anything," http://www.wikihow.com.

[4] S.-Y. Hwang and Y.-F. Chen, "Personal workflows: Modeling and management," in *Mobile Data Management*, 2003, pp. 141–152.

[5] S. W. Kim, H. Paik, and I. Weber, "Automating form-based processes through annotation," in *Service-Oriented Computing - 10th International Conference, ICSOC 2012, Shanghai, China, November 12-15, 2012. Proceedings*, 2012, pp. 558–565.

[6] I. Weber, H. Paik, and B. Benatallah, "Form-based web service composition for domain experts," *TWEB*, vol. 8, no. 1, p. 2, 2013.

[7] I. Weber, H.-Y. Paik, B. Benatallah, C. Vorwerk, L. Zheng, and S. Kim, "Personal Process Management: Design and Execution for End-Users," UNSW-CSE-TR-1018, Tech. Rep., 2010.

[8] M. Brambilla, "Application and simplification of BPM techniques for personal process management," in *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, 2012, pp. 227–233.

[9] J. Xu, H. Paik, A. H. H. Ngu, and L. Zhan, "Personal process description graph for describing and querying personal processes," in *Databases Theory and Applications - 26th Australasian Database Conference, ADC*, 2015, pp. 91–103.

[10] A. Koschmider, M. Song, and H. A. Reijers, "Social Software for Modeling Business Processes," in *Business Process Management Workshops*, 2008, pp. 666–677.

[11] A. R. Silva, R. Meziani, R. Magalhães, D. Martinho, A. Aguiar, and N. Flores, "AGILIPO: Embedding Social Software Features into Business Process Tools," in *Business Process Management Workshops*, 2009, pp. 219–230.

[12] D. Martinho and A. R. Silva, "Non-intrusive Capture of Business Processes Using Social Software - Capturing the End Users' Tacit Knowledge," in *Business Process Management Workshops (1)*, 2011, pp. 207–218.

[13] B. Hartmann, D. MacDougall, J. Brandt, and S. R. Klemmer, "What would other programmers do: suggesting solutions to error messages," in *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI*, 2010, pp. 1019–1028.

[14] J. Matejka, W. Li, T. Grossman, and G. Fitzmaurice, "Communitycommands: Command recommendations for software applications," in *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*. ACM, 2009, pp. 193–202.

[15] S. A. Hajimirsadeghi, H.-Y. Paik, and J. Shepherd, "Processbook: Towards social network-based personal process management," in *Business Process Management Workshops*, 2012, pp. 268–279.

[16] F. Beuter, "Design and implementation of task management lifecycle concepts based on process mining," Master's thesis, Ulm University, Ulm, Germany, 6 2015.

[17] N. Mundbrod and M. Reichert, "Process-aware task management support for knowledge-intensive business processes: Findings, challenges, requirements," in *18th IEEE International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*, 2014, pp. 116–125.

[18] Y. T. Giovanni Maria Sacco, Ed., *Dynamic Taxonomies and Faceted Search: Theoty, Practice, and Experience*, ser. ISBN 978-3-642-02358-3. Springer, 2009.