Human Collaborative Workflow and Business Process and Services Computing

Yoichi Takayama, Ernie Ghiglione, Scott Wilson*, James Dalziel

Macquarie E-Learning Centre Of Excellence (MELCOE)
Macquarie University
E6A-248 Eastern Road
Macquarie University, NSW 2109
Australia
*Centre for Educational Technology & Interoperability Standards (CETIS)
The University of Bolton
Deane Road
Bolton, BL3 5AB
United Kingdom
yoichi@melcoe.mq.edu.au
ernieg@melcoe.mq.edu.au
scott.bradley.wilson@gmail.com
james@melcoe.mq.edu.au

Abstract: Comparing our collaborative human workflow system to existing business process and services computing (BPSC) and workflow models showed that main differences of human workflow are concepts such as human flow, collaborative activity and sub-grouped activity. The possible impact of these findings on the forthcoming implementation of BPSC models that deal with human activities is discussed.

1 Introduction

Workflow has emerged as one of the key paradigms in the design of systems. Over the years, the principle has been implemented differently in various areas, for example, business process management (BPM), document-centric workflow (document flow), data-centric workflow (data flow), and human-centric workflow (human flow), each of which exhibits both common and different characteristics which have diverged from the same route principle.

Although most efforts for research and development, both academic and commercial, have been concentrated on BPSC, there have been few formal representations of human flow to describe its characteristics, particularly compared with other types of workflow manifestations. Also, in spite of the recent recognition of the importance of human activities in BPM area [Ha05], there have been few BPM studies that look at the issue from the experience of a human flow. This paper attempts (1) to clarify the common and different aspects of the human flow compared with BPMN and (2) to consider its applicability to BPSC.

In particular, over the past five years MELCOE has developed a human flow system, LAMS (Learning Activity Management System) [Da03], and we will be comparing the LAMS with BPSC. LAMS is an open source collaborative human workflow system for e-Learning and it uses collaborative human activities. It belongs to a line of human flow technologies developed within the field of education, such as Educational Modelling Language (EML) [Ko01] and IMS Learning Design (IMS LD) specification [Im03]. These technologies can be categorized as *human flow, activityflow [Da06]* or *collaborative human workflow*. Also, recently we are developing RAMS (Research Activity Management System) which applies the concepts of human flow (and a modified version of the LAMS software) to e-Research collaboration [Da06, Wi07].

Collaborative human workflow systems can be characterised by (1) using human flow as the flow model and by (2) using collaborative human activities in all activity nodes in the flow model. So far as we know, these are not seen in conventional business process or workflow modelling outside the collaborative human workflow field.

On the BPSC side, Business Process Modelling Notation 1.0 (BPMN) [Om06] has been selected as a representative technology. BPMN aims to be an implementation agnostic workflow model but also to be executable. It aims to maintain mapping to the Business Process Execution Language (BPEL) [Oa07]. It is also in line with the workflow pattern research [Aa03]. The workflow pattern research seeks to identify and abstract common *patterns* in various workflow situations and it has become particularly a useful resource as a reference point.

Additionally, although the need has been identified, there is no human flow specification for BPM or BPSC yet. A white paper, BPEL4People [IS05], however, will be used as a reference to evaluate issues around human activity implementation in BPSC.

2 Unique features of human collaborative workflow

The following features of human collaborative workflow were used to compare them with BPMN since such attempts will help understand the similarities and differences.

2.1 Human flow

LAMS/RAMS uses human flow. Human flow is a workflow in which the main flow diagram (*control flow*) consists of activities which are all human activities and users *flow* from one activity to next activity when the transition occurs.

In the human workflow, it seems to be a unique problem that the human moved is aware of the transition. For example, in a workflow model which uses data flow as a main control flow, the moved data itself does not have to be aware of the transition. Only the next activity needs to be notified that it now has to execute with the data when the transition occurs. By contract, a human may have awareness of a transition and this may dictate some differences. This may be a case of the second-order cybernetics that described this type of reflective phenomenon [Fo03]. In one way, a human is the observer as well as the observed when using human workflow. The user acts on the workflow but is also moved by the workflow (being acted on by it). This also generates unique awareness between the modeller and the humans who are modeled.

Another unique feature of human flow in LAMS/RAMS is that often the human user can go back to past activities to review, continue working or retry (Figure 1). In BPMN, normally the workflows are automated non-human processes and a user moving back to past activities is not expected unless explicit looping or some compensation processes are provided. Activities of human flow have design time setting that either allow (or deny) review, continued working or retry of the activities by the user.

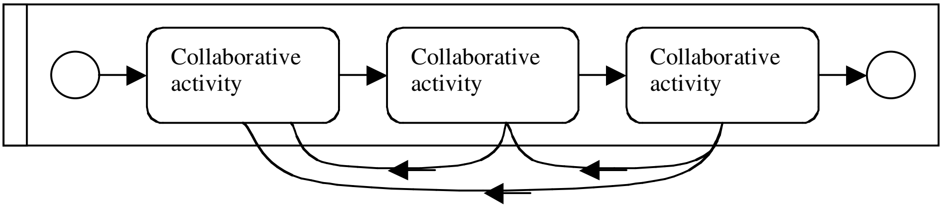

Fig. 1. Human flow with Collaborative activities. Users may be able to go back to activities but cannot jump ahead of activities that are not attempted yet.

2.2 Collaborative activity

As shown in Figure 1, the collaborative human workflow consists of collaborative activities. A collaborative activity is an activity in which human users may work collaboratively. For example, forum, chat, collaborative authoring and whiteboard are collaborative activities. These functionalities may exist as ubiquitously available *tools* in some other systems, which are not workflow systems. In such systems, they are not implemented as activities. In collaborative human workflow systems, however, they are made available as activities so that the functionalities are supplied when and where they are needed.

For example, in LAMS, to allow the collaboration to occur multiple users are grouped in a "Class" and the Class is assigned to a "scheduled" workflow design (a "Lesson"). Because multiple users are "on" the design and flowing together, collaborations can take place on the activities when they reach the activities together.

At the backend, however, multiple instances of the flow and also multiple instances of the activities are created to handle individual users. What is unique in the LAMS modelling is that the collaborations (indicated by dotted lines in Figure 2) take place without explicitly modelling such interactions (Figure 1).

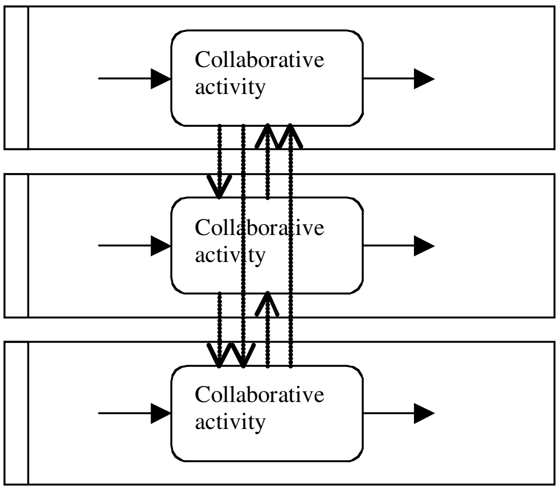

Fig. 2. A Collaborative activity with 3 shared users at runtime. Three runtime instances exist as depicted, but conceptually there is one Collaborative activity for 3 users on the original design. The runtime instances actually talk to just one parental activity on the design (or its service) at the backend. Superficially, however, it appears that all users are talking to one another or together because they receive real time updates of interactions from all members of the group from the service.

2.3 Grouped activity

The human collaborative workflow uses sub-groups of the Class in many situations. Sub-groups are created by Grouping activity, in which the teacher may choose manually put students to the desired number of groups, or allow the system to randomly divide the Class into a number of groups specified by the teacher. The sub-groups are used by either Grouped activity or Group-based branching.

The Group-based branching is straightforward. Each Group proceeds along different predefined branches according to what group the user belongs (Figure 3).

The Grouped activity requires a little explanation. This is a single activity to which the Grouping is "applied". This is used in a situation in which the teacher would like the same collaborative activity for the entire Class but would like the collaboration to happen only within each Group. For example, the teacher may like to have the Class to discuss Global Warming but in small groups because discussion in a large class tends to silence non-vocal people. All groups discuss the same topic but may experience different discussions or reach different consensus. Comparing the results afterwards may be useful. Any LAMS activity which can accept the Grouping being applied can become a Grouped activity.

The Group-based branching may be modelled easily in BPMN. The Grouped activities (or "applying" Groups to activities) do not exist in BPMN but may be modelled as Multi-instance activities. We will consider the possible modelling in 3.3.

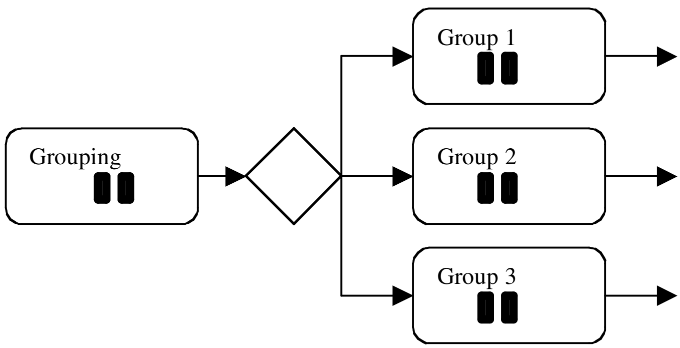

Fig. 3. Group-based branching.

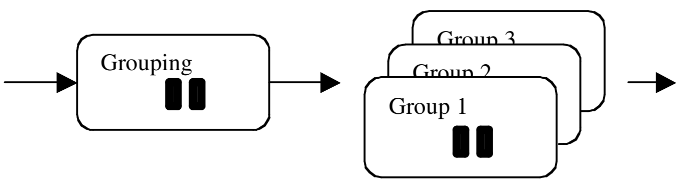

Fig. 4. Grouping and a Grouped activity

2.4 Tool Interface

LAMS/RAMS uses a Tool Interface to access the underlying Tools from Activities. Conversely, Tools use the Tool Interface to access various system service calls or ask the system to present human interface (GUI, graphical user interface). From workflow point view, the Tools are services and the locations are expressed as URLs, but they are not Web Services. Unlike the services of a typical orchestration solution, these services are not faceless, i.e. they have a human interface.

The Tool Interface is subdivided to Authoring, Monitoring, Admin and Learner/Participant components. These sub-services are accessed via 4 corresponding system modules.

At authoring time, the system requests the Tool Authoring sub-system (service) to be activated. The service presents the Authoring GUI of the Tool to the author for creation of the "Contents" of the Tool. The Contents is (1) various properties of the Tool which govern the runtime behaviours and (2) actual textual data which the Tool may use it for calculation or to present to the end users at runtime. When the authoring session ends, the Tool saves the authored Contents and returns to the system. The system then saves the workflow design with the Activity with the toolContentID.

Table 1. Design time Attributes for Activity.

Attribute	Description
toolContentID	System passes it to the Activity.

At runtime, the Learner/Participant Interface passes the Attributes in Table 2 to the Tool's Runtime Service. The service presents the Learner GUI of the Tool for the user to interact. When the Tool task is completed, it returns the Completion message.

Table 2. Runtime Attributes passed to Tools.

Attribute	Description
toolSessionName	Descriptive name for the Tool instance.
toolSessionID	ID of this Activity, includes links to userID, GroupID, etc.
toolContentID	ID of the Contents created at the authoring time.
defineLater	Boolean. Indicates that the content of this activity is not finalized and the users should wait.
runOffline	Boolean. Indicates that it is a "Face to Face" Task (ie, this task is not run using a LAMS/RAMS tool), equivalent to the Manual Task (BPMN).

At monitoring time, the Monitoring Interface passes the Attributes in Table 3 to the Tool's Monitoring service. The service presents the Monitoring GUI of the Tool to the user to allow him/her to monitor or control various aspects of the Tool. For example, a teacher is able to monitor the progress of students (including intervening in discussion as the teacher, or editing/hiding inappropriate student comments, etc).

Table 3. Monitoring time Attributes passed to Tools.

Attribute	Description
toolSessionName	Descriptive name for the Tool instance.
toolSessionID	ID of this Activity, includes links to userID, GroupID, etc.

At Tool administration time, various properties of Tools may be passed from the system to Tools via Admin service of the Tools. The service is also used for Tool deployment. Since this does not concern BPM or modelling issues, this topic is omitted from this paper.

3 Human collaborative workflow and BPSC

In the following section, we analyse key terms from relevant Business Process and Services Computing specifications (especially BPMN and BPEL4People) in relation to our experience with human collaborative workflow.

3.1 Human flow and BPMN

ProcessType: BPMN has a provision to map BPMN Processes to lower level collaborative language by specifying ProcessType = Collaborative. The ebXML [Oa02] and Web Service Choreography Description Language (WS-CDL) [W305] are named as candidates. To model a human flow, this may be the option to be used. However, the word "Collaborative" is used to mean Choreography clearly, which is a type of collaboration between different business processes. It is different from the usage of the word in human flow, which means the concerted and simultaneous interactions between human users (either synchronously or asynchronously). This may include, for example, discussion board, chat, mailing list, a consensus builder, collaborative authoring tool, idea pooling & exchange and whiteboard. This BPMN feature, however, is not implemented in the current BPMN definition and it is not possible to evaluate it for use for collaborative human workflow at this time.

TaskType: BPMN [Om06] defines a Task as an atomic activity that is operated by either end users or applications, which cannot be broken down into further details. It defines the following TaskType:

Service | Receive | Send | User | Script | Manual | Reference | None

The Human flow may consist of User Tasks, with following Attributes:

Table 4. Additional Attributes for User TaskType in BPMN [Om06].

Attributes	**Description**
TaskType	User
Performers	A specific individual, a group, or an organization.
InMessage	An outgoing message to be sent to the task implementation at the start of this Task, after any InputSets are made available.
OutMessage	An incoming message received from the task implementation at the completion of the task, which may cause the production of an OutputSet.
Implementation	Web Service	Other	Unspecified

The Messages are defined when a designer imports WSDL/XSD files to the BPMN modeller. Additional variables may be also created by using the XSD. The difference between User Task and the Service Task is that the latter does not have the Performers Attribute.

User Task is expected to schedule the task with a certain task scheduler which will manage "Tasks" and send notices to the human user(s). If we used a scheduler in LAMS, this would cause the task scheduler to post every human activity to the users. To complete a workflow, the users would need to go to the task list to retrieve and act on them. In other words, this would not be suitable for use in human flow because this will break the smooth transition between activities.

The use of the scheduler, however, is well justified for a system which uses automated processes and in which the human activity requires notification. As BPEL4People noted [IS05], what needs to be considered is a smooth "transition" between an automated activity and a human activity. In such a scheme, a scheduler may play a role when an automated activity transits to a human activity. If the subsequent activity is also a human activity, however, the system may not use the task scheduler and may present a smooth transition effect and subsequently the next activity.

The User Task has an access to the Participant Attribute of the Pool, which is either Entity name or Role name used in the Pool. It may be more appropriate if the Task can pass the current user reference, who is assuming the Role and running the workflow, but accessing such a runtime value is not defined in BPMN. This is because the Participant or Role means the Role of the Process, not a human aspect normally in a Process since it may be agnostic to who is the actual human user.

The InMessage typically is expected to be some data for a Form. The human interface of the implementation is expected to present the Form to the addressed human user when the Task is accessed.

Figure 5 represents the typical example of modelling the human activity in BPMN. The "User" layer is expected to schedule tasks and to post notices to human users. It also acts as a human interface to present the Messages received from the Process layer in appropriate interface. For example, it may present a simple form-like entry field to ask for a human input.

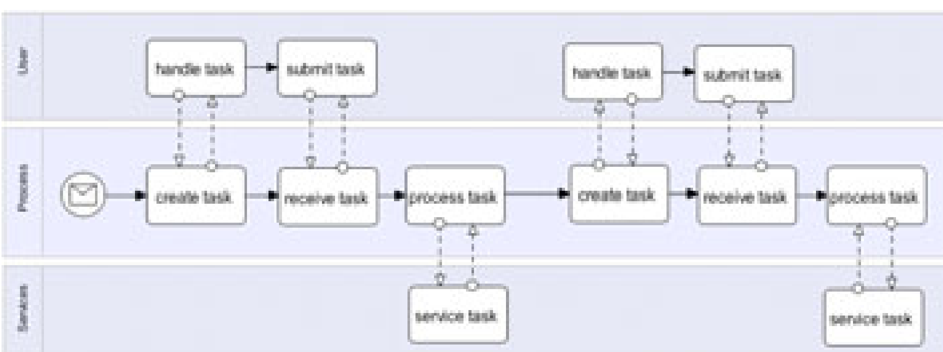

Fig. 5. A typical modelling of human activities in BPMN.

In human flow, on the other hand, what we really want to model is the human interface in which human users can see transitions happening as well as see when rich human interactions can take place. In other words, the Process layer is more integrated with human workflow functionalities (Figure 6). We analyse this possibility in the following section.

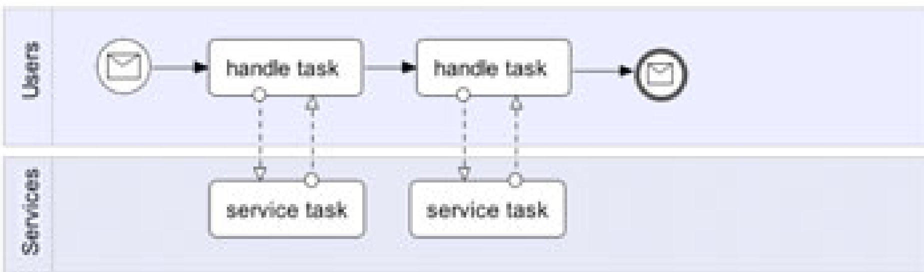

Figure 6. Integrated human flow and the relationship with services. Human GUI layer is omitted.

Multi-instance sub-flow: In the collaborative human flow, multiple users are flowing on a workflow design. One way to achieve this multiplicity with BPMN may be to put all Collaborative activities inside a Multi-instance Sub-process (or sub-flow) activity. The figure 7 indicates an attempt to model the human flow in a BPMN fashion. In BPMN, a Multi-instance activity is categorized as a Parallel Loop type and this example may be defined with parameters:

LoopType = MultiInstance
MI_Ordering = Parallel
MI_FlowCondition = None (completion is not synchronized)
MI_Condition = <expression which resolves to an integer>

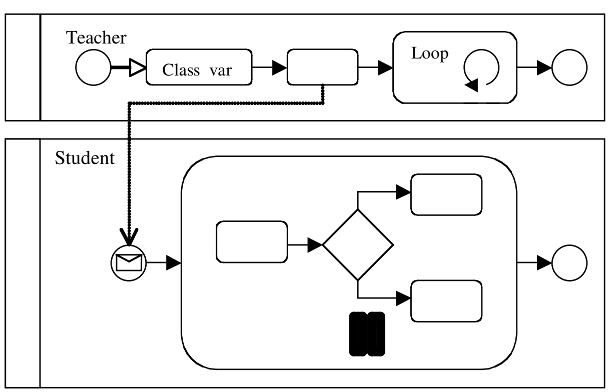

Fig 7. Collaborative human flow as a Multi-instance sub-process activity.

One problem of this approach is that the Multi-instance activity must receive the fixed MI_Condition when it is invoked. This number is used to create the required number of child Sub-processes. In case of Human flow, unknown number of the users in this *Role* may use workflow and spawn Sub-processes as they "join" in the workflow and there is no need to create the instances all together at one instance. For this reason, MI_Condition do not have to be known a priori. BPMN does not have a similar mechanism. As indicated in the above figure, the students spawn the Sub-process in the lower Pool. In this figure, it is desired that they "join" to the lower student Pool when they "start". (The starting mechanism has been discussed by Wilson *et al.* [Wi07].)

Activities in Sub-processes are collaborative activities and must interact together. One way to indicate this at design time is to provide a unique "Class" variable, which indicates that all Sub-processes belong to the same instance of the teacher process. This must be passed to the Student Pool and the activities must be able to access this variable. Apparently this can be done once BPMN is converted to the BPEL level for execution. The BPEL elements being involved are known as *correlation* and *perpertyAlias*. Otherwise, BPMN does not assume any knowledge about the system on which it is run. Therefore, it does not know any user management mechanism to obtain the actual group information, such as a "Class". Therefore, the Class variable is a design level variable that must generate a unique value to mark the "Class" when workflow design is executed.

Creating students' Sub-processes for the "Class" and *correlating* them is actually what happens in LAMS implicitly without a need to specify actual details in a Process or workflow diagram. This is because LAMS processes interpret the main control flow diagram as Multi-instance flows and attache the Class ID automatically for all the Sub-processes in order to be able to handle them together.

Synchronisation: Another point to consider is a Synchronisation Gate activity in human multi-instance flow. In collaborative human flow, Synchronisation means synchronising different users in the multiple Sub-processes, i.e. individual users wait at a stop point until all users arrive at this point, and only then is the stop point removed, so that all users can then progress to the next task. This is different in normal BPM where synchronisation means an AND-join. In BPM, synchronisation happens either in (1) AND-join Gate of branched folks of a single Process or (2) implicit merge of multiple instances at the completion of (2a) Multi-instance activity, (2b) Multi-instance Sub-process activity or (2c) a Loop activity, if MI_FlowCondition = all. These types of AND-join merge and synchronise the parallel activities of a single user (unless different users are assigned to different split instances later), whereas Synchronisation Gate of human flow synchronises different users who are in different flows without merging their flows.

Since the synchronisation in human flow is different from that of conventional BPM, a different Synchronisation Gate activity symbol needs to be introduced (Figure 8). Similar to the Class variable above, a Group variable must be passed to the Synchronisation Gate, except Group = Class where a sub-grouping is not applied.

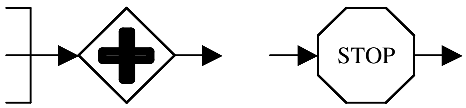

Fig. 8. Normal And-join and Human flow Synchronisation Gate. The Synchronisation Gate always has only one arrow coming in.

3.2 Collaborative activities and BPMN

In the above example, the created instances by the Multi-instance sub-flow activity do not know that they are supposed to interact without any additional information. Again, this is technically solvable by indicating what parent the children belong to. There, however, is no symbol in the BPMN standard to instruct that such relationship and interactions should happen. Furthermore, there are two relationships. The first is that all the sub-flows belong to the parent (Figure 9). The second is that activities in the sub-flows belong to the corresponding parent activities that are in the sub-flow design (Figure 10).

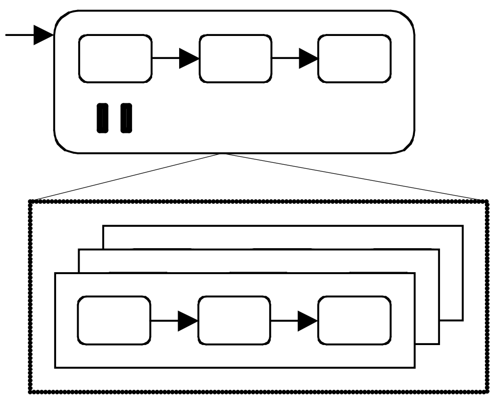

Fig. 9. Sub-processes spawned from the Multi-instance Sub-process Activity.

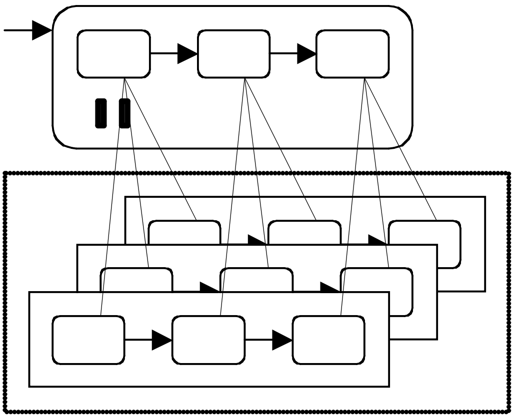

Fig. 10. Child Collaborative Activities have relationship via their parents.

As discussed in 3.1, this situation may be handled by passing the Class variable to link the multiple sub-flows together and the Group variable (Group = Class if there is no sub-grouping is used) to link the Collaborative activities together by the use of BPEL *correlation*. We, however, are not implementing the capability in the BPMN itself to express multiple instance nature of human flow or graphical cue for Collaborative nature of activities.

3.3 Sub-grouped activity in BPMN

Although BPMN can produce Multi-instance Sub-processes, normally these are not considered to form a group or sub-groups. For this reason, there are no graphical symbols to indicate Grouping, Sub-grouped activity or Group-based Branching in BPM.

If BPM were to be modified, (1) the Grouping and Grouped Activities and Group-based Branching could be added, (2) and the Group variable may be passed to one or more Grouped activities (or to the branches), and (3) the collaboration may be handled only between members within the same Group (Figure 11).

At the moment, this may be modelled using a variable or DataObject Artifact to pass the Grouping in a flow, if the Grouping is done per user basis. If the Grouping must be done as a whole Class (e.g. if the entire Class list or on-line user list is required), they may be coordinated with *correlation* or by a service obtaining the information.

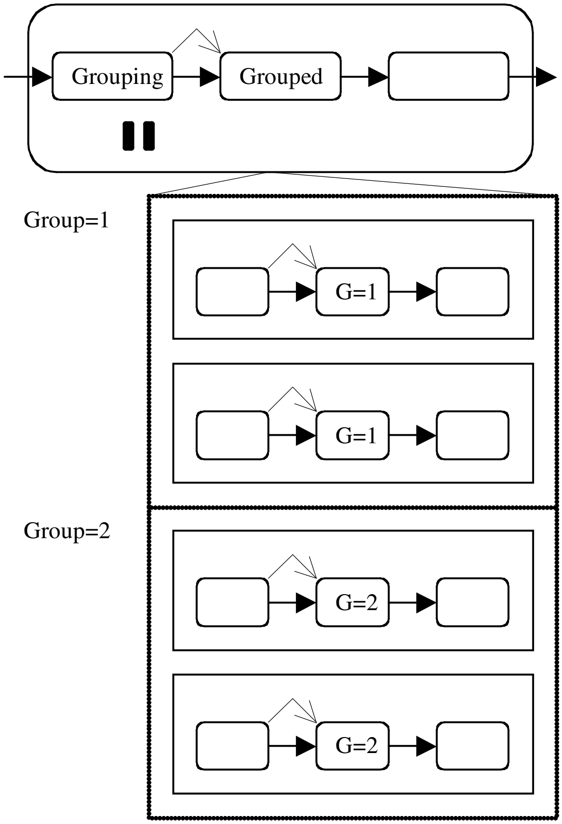

Fig. 11. Grouped Activities using a Group variable in Multi-instance Sub-flow Activity (this example represents two groups of two students each).

3.4 Tool Interface and BPMN Service Interface

The LAMS/RAMS Tool Interface may be analogous to the Service or User Tasks interface used by BPMN. Although BPMN is language agnostic and simply refers the interface with e.g. “Message” Attributes, the modellers actually rely on WSDL/XSD as BPEL to be the suggested execution language. Nonetheless, at the notation level, it may not need these files.

LAMS uses its own “message” formats defined in a similar way, i.e. the system defines common elements in the message and each Tool can freely define its own elements. The information is provided to the system and used wherever the system needed to delegate the handling of the Tool Contents to Tools.

Apart from the workflow conceptual aspects, it is advantageous that the Tool Interface has four interfaces to handle different sub-systems in human flow. For example, considering monitoring of services at runtime; it makes sense to ask the same services to provide runtime status information in human flow system. In BPM this may not be a concern. In BPM, the system may be monitoring processes but not each service. For services in BPM, getting normal or error responses may be enough. In human activity processes, however, the service may be executing for a considerably longer time. It may be more important to be able to monitor the status of activities (or services) while the human users are using the services.

At this moment, in LAMS there is no data flow between the Activities and the contained Tools or between LAMS Activities. BPMN has DataObject Artifact as either input or output Association to Activities. This concept may be useful in LAMS, in order to provide exchange of data between Activities and Services and between Activities.

3.5 Human flow and BPEL4PPeople, and the impact on BPMN

BPEL4People recognises the necessity of human involvement in the BPM. It also recognises that the human activity may be more complex than automated processes and services, such as 4-eye principle.

For the current discussion, the BPEL4People white paper may be summarised as:

(1) Implement BPEL4People as an extension to BPEL.
(2) Set up roles: *process initiator*, *process stake holder*, *potential process owners*, *actual process owner* (who claimed processes), *business administrator*.
(3) Add *people link* element.
(4) Add people resolution which resolves the people link to actual human users.
(5) Add new activity type, *people activity* to BPEL, which create a *Task* instance
(6) Implement Tasks for *people activity*: *Query Task, Claim Task, Revoke Claim, Complete Task, Fail Task.*
(7) Implement task status: *ready, claimed, completed, failed.*
(8) Add context: execution *context* to let BPEL know
(9) Implement rich and varied user interface for *people activity*

This proposal is analogous to our findings in this paper:

a) In order to include human activities in BPSC, it seems that it is essential to use a new element in the workflow model to handle the activity differently from the process activities. This is named as *people activity* in the white paper. LAMS already handles the human activity differently from the automated process.
b) Human roles must be included in the workflow, but this has to be distinguished from the conventional BPMN or BPEL Roles, which are used to mean “roles” of automated processes (Pools). Human roles are different and this is represented as *people link* in the proposal.

c) Since the workflow at design time may not know the actual human, therefore, the *people links* needs to be resolved to actual human users who exist in the system at runtime.
d) *People activity* requires a system to provide a special user interface (e.g. GUI).

As to the Task model, LAMS does not use the concept, therefore, it may not be essential for any implementation. Probably this should be left to each implementation. Also, in the LAMS experience to date, we did not find the need for the proposed human roles.

The impact of these findings on BPMN is that BPMN also needs to implement *people activity, people link,* people resolution and the interface. As we saw in 3.1, modelling human activity in BPMN only results in a diagram with too many details. If it also desires to be able to model human collaborative workflow, it should be able to handle multi-instance flow or a human flow as a flow option.

4 Discussions

The normal case of Multi-instance activity assumes a single user invoking it to create multiple child instances. In human flow, however, multiple users with the same role share a flow and create the Multi-instance activity (which is multi-instance in runtime). This was satisfactory in LAMS where the Student role was implicit. In RAMS, however, there may be explicit roles and they may even be dynamic. Users with same and different roles may share flows and, therefore, activities. For example, researchers (standard user), project manager (moderator and supervisor), project owner (monitor) may be using the "same" flow and the collaborative activity, which may be used by each user for different purposes and with different authorities. Also, human users may switch roles or there may be different sets of roles for different activities. This was not the case with BPMN, where fewer Roles were expected.

This may not be modelled easily with the Multi-instance sub-process and simply setting up Pools (processes) representing fixed human roles in a modelling like BPMN. In addition to roles, the instantiation of flows for different roles must be handled and the *correlation* of the instances must be set up properly. This can be all set up explicitly, but the challenge is to find a model which can hide all details and provide a simpler model. One solution may be a role-overlay (e.g. properties), in which roles are not expressed as Pool or Lanes, therefore, the roles can have more complex structure and the topology does not have to be expressed as a graph on the main diagram.

5 Conclusions

It was useful to attempt to model LAMS/RAMS workflow with BPMN in order to see how the human flow modelling fits with the BPM modelling.

Since the problem areas are different and models are different, direct fit of models was not expected. Rather, the current study was aiming to examine whether one experience may learn from the other.

The study makes it clear that, to handle human activities, a process needs to implement a human activity type, which needs to be handled differently from the services. BPMN already has the User Task. Therefore, the special handling of it can be left to the execution layer, which could be BPEL plus BPEL4People.

The concept of *people link*, however, will also have to be implemented in BPMN and used in BPEL4People when the model is translated to the execution code.

Greater dependency on BPEL and BPEL4People means that a BPMN modelling system or server probably has to be shipped with BPMN, BPEL and BPEL4People together.

In order to model collaborative human flows, it seems that BPMN needs to indicate multi-flow and collaborative activities on the flow diagram, maybe as an option. When multi-flow is employed, the multiplicity of collaborative activities is that of the flow, therefore, there is no need to mark the activities multi-instance. Alternatively, it has to use Multi-instance Sub-flow activity and *correlation* technique, which will not look simple to the people who attempt it.

Acknowledgements

This work was funded by the Australian Government Department of Education, Science and Training.

References

[Aa03] van der Aalst, W.M.P; ter Hofstede, A.H.M.; Kiepuszewski, B.; and Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases, 14(3), pp 5-51, 2003.

[Da03] Dalziel, J.: Implementing Learning Design: The Learning Activity Management System (LAMS). In G.Crisp, D.Thiele, I.Scholten, S.Barker and J.Baron (Eds), Interact, Integrate, Impact: Proceedings of the 20th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education. Adelaide, 7-10 December. 2003. Available from http://www.ascilite.org.au/conferences/adelaide03/docs/pdf/593.pdf

[Da06] Dalziel, J.: Research Activityflow and Middleware Priorities project proposal. 2006 Available from http://www.ramp.org.au/RAMP.MainBodyPublic.doc

[Fo03] von Foerster, H.: Understanding Understanding: Essays on Cybernetics and Cognition, New York: Springer-Verlag. 2003.

[Ha05] Harrison-Broninski, K.: Human Interactions: The Heart and Soul of Business Process Management. Tampa: FL, Meghan-Kiffer Press, 2005.

[Im03] IMS Learning Design specification V1.0. 2003. Available from http://www.imsglobal.org/learningdesign/index.cfm

[IS05] IBM, SAP: WS-BPEL Extension for People – BPEL4People. A Joint White Paper by IBM and SAP. 2005.Available from http://www-128.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/

[Ko01] Koper, R.: From change to renewal: Educational technology foundations of electronic environments. 2001. Available from EML website: http://www.eml.ou.nl/

[Oa02] OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee: Collaboration-Protocol Profile and Agreement. Specification Version 2.0. 2002. Available from http://www.ebxml.org/

[Oa07] Web Services Business Process Execution Language Version 2.0. Committee Specification. OASIS Web Services Business Process Execution Language (WSBPEL) TC. 2007. Available from http://docs.oasis-open.org/wsbpel/2.0/

[Om06] OMG (Object Management Group): Business Process Modeling Notation (BPMN) Specification. Final Adopted Specification. 2006. Available from http://www.bpmn.org/

[W305] W3C: Web Services Choreography Description Language Version 1.0. 2005. Available from http://www.w3.org/TR/ws-cdl-10/

[Wi07] Wilson, S., Takayama, Y., Ghiglione, E., Dalziel, J.: Designing systems for managing dynamic collaborative research processes. International Journal of Learning Technology. 2007 (In Press)