

Business process modeling pluralized

Citation for published version (APA):

Turetken, O., & Demirors, O. (2013). Business process modeling pluralized. In H. Fischer, & J. Schneeberger (Eds.), *5th International Conference on S-BPM ONE 2013, Deggendorf, Germany, March 11-12, 2013*. (pp. 34-51). (Communications in Computer and Information Science; Vol. 360). Springer. https://doi.org/10.1007/978-3-642-36754-0_3

DOI:

[10.1007/978-3-642-36754-0_3](https://doi.org/10.1007/978-3-642-36754-0_3)

Document status and date:

Published: 01/01/2013

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

This is the author's version of a work that was submitted/accepted for publication in the following source:
Turetken, O., Demirors, O. (2013). Business Process Modeling Pluralized, Subject-oriented Business Process Management (S-BPM) ONE 2013, CCIS 360, pp. 34–51, Springer-Verlag Berlin Heidelberg.

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source: <http://link.springer.com/>

Business Process Modeling *Pluralized*

Oktay Turetken¹ and Onur Demirors²,

¹ Eindhoven University of Technology, Information Systems Group,
School of Industrial Engr., De Lismortel 2, Pav.D.08, 5600 MB Eindhoven, Netherlands
o.turetken@tue.nl

² Middle East Technical University, Informatics Institute, 06531, Ankara, Turkey
demirors@metu.edu.tr

Abstract. Traditional centralized business process management approaches pose difficulties in coping with rapid changes and evolving process models. We developed the Plural method to allow for decentralized modeling of processes. The Plural method enables process participants, rather than a centralized group of process engineers or managers, to model and maintain their processes. In previous works, we introduced the Plural method and discussed its applications in case studies. This paper elaborates more into the notation component of the Plural method. We describe the diagram types used for capturing process information. We also present a case study performed in a small web application development company. We discuss the results together with a synthesis of the findings and lessons learned from our previous case studies.

Keywords: Decentralized business process modeling, subject-oriented business process modeling, role-oriented process modeling, Plural

1 Introduction

In the society, where knowledge is the primary resource for individuals and for the economy, many researchers on business management agree that the traditional structures of organizations are not appropriate for creating products and services that require knowledge work and its integration [1], [2]. Quality improvement is possible if continual learning becomes a way of organizational life. Senge [1] argues that this can be achievable if traditional authoritarian, command-and-control hierarchy -where the top thinks and the local acts- is broken. Merging thinking and acting at all levels is necessary. Such organizations, -which are sometimes called 'lattice' or 'agile'-, involve direct transactions, self-commitment, and natural leadership, where tasks and functions are organized through commitments [3].

In many aspects of an organization's life process models are core assets. They are used to understand and analyze organizational knowledge, automate the processing, and act as the primary means for communication [4]. The value of this asset increases if they are embraced by the performers, if they accurately reflect the executed processes, and if they can easily evolve to adapt to ever-changing business requirements.

Traditional business process modeling typically employs a centralized and top-down approach, which assumes a central group of experts (process engineers,

designers, or managers) working with groups of individuals (process participants or performers) to elicit process information and depict processes [5]. The involvement of the process participants is key to accurately capture the executed processes and to help the designed processes be embraced by the participants [6]. We believe that the full advantage of this involvement is obtained when the individuals themselves model their own processes.

With a top down, centralized approach, where processes are modeled consecutively by a small group, it usually takes considerable amount of time. More importantly, once process definitions are considered stable it is usually difficult and not desired to change them frequently [7], [8]. However, to respond to the changing business environment, organizations should be able to change their way of working rapidly. In that respect, their process management infrastructure should be flexible to allow for rapid changes also by process participants [9]. We believe that this goal could be achieved by a decentralized modeling approach aligned with the principles of subject-oriented business process management.

We developed the *Plural* method as a disciplined guideline for organizations to perform *process modeling* in a decentralized way, allowing process participants to take responsibility for describing and improving their own processes, and collectively building and maintaining the organization's process-base. Our earlier work [10] presents the Plural method in detail and describes the results of its application in a case study performed in a small-size organization. The case study revealed several advantages as well as limitations of the method with promising avenues for future research.

This paper elaborates more into the *Plural notation* as a key enabler of the method. It describes a set of diagrams that accompany the implementation of the method. It also discusses the results of a case study to validate and confirm the benefits identified in our previous case studies.

The remainder of the paper is organized as follows: Section 2 presents a brief overview of the Plural method. Section 3 introduces the Plural notation. In Section 4, we describe the application of the method in a case study and discuss the findings. Section 5 presents the conclusions.

2 The Plural Method

The Plural method is grounded on the idea of allowing process participants to *model* their processes and *maintain* these definitions. Process participants define the *operations* they perform (serve) with respect to the roles they act for within a specific process. In addition, they define their *interface* to their operations in terms of the messages they exchange with other participants, stakeholders, and entities in the business environment. In cases of inconsistencies between the definitions of different process participants, they communicate to solve the issue. The definitions (for operations) can be integrated where necessary to visualize process information in various ways, and give insight into the way the organization works. The models that can be generated include end-to-end process diagrams, process dependency and role-dependency diagrams depicting dependencies based on the messages exchanged.

Plural is an *iterative* approach for process definition with three main phases as depicted in Fig. 1. We summarize the phases below and refer the reader to [10] for an elaborate description of the phases and the roles that are involved.

Context Definition. The method initiates with the identification of the *scope*, which consists mainly a high-level process network, participating roles and agents, and their structural relationships. Process participants and other stakeholders (sponsors, etc.) (i) determine the purpose of the modeling initiative, (ii) identify the processes to be covered and the roles that take part in each process, and (iii) select the coordinator(s) that will facilitate the modeling throughout the iterations. They finally (iv) assign participants to roles and plan the first iteration for the modeling.

Coordinator role is key to ensuring that Plural method is appropriately applied. A coordinator guides participants in modeling and maintaining the process network, remove the roadblocks, and makes sure that Plural principles are properly followed. However, he has no authority over participants. He envisages the top view of processes as a whole, verify individual operation definition models, identify problems and capture high-level improvements.

Description and Conflict Resolution. Having been assigned to a set of roles, process participants first identify the *operations* they perform with respect to the processes they participate. Each operation is a cohesive set of activities performed by a specific role. Next, participants define the behavior for each role-operation. This consists the *activities* they perform, the *information items* they require as inputs and those that they produce as outputs. In addition (and as a key concept in Plural), participants provide the sources of the inputs and destinations of the outputs, if any. The sources might be other roles or entities, such as project repositories, folders, software tools, or other operations of the same role. Participants also represent the activities their roles perform with other roles. This representation of the interactions forms the *expectations* of that role from other roles or business entities.

A role's expectations are satisfied (and thus the models are consistent) if, in the models of the other roles, the expectations are acknowledged and shown at the expected interface. For example: suppose in a simplified *loan processing* scenario, the *loan manager* defines that he needs *loan information* from the *loan processing clerk* as input to his 'approve loan' operation. This expectation is considered 'satisfied' if the clerk, in any of her operation model, declares that she provides this information

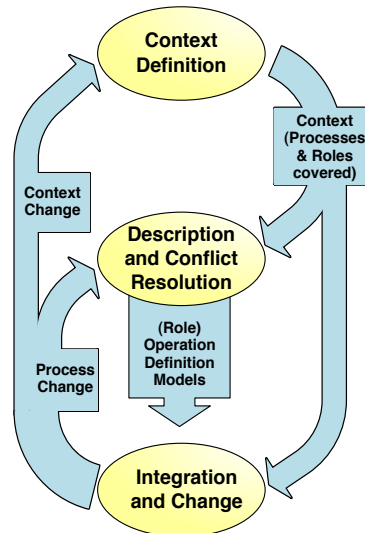


Fig. 1. The Plural phases.

item to the manager. Otherwise, we consider that there is an *inconsistency* between the expectations of these two roles.

Inconsistency resolution is participants' responsibility. Inconsistencies with respect to unsatisfied expectations may originate due to a range of reasons; from a simple typo or –more seriously– a misunderstanding or a concealed assumption regarding how the process executes (or will execute). In the later case, the resolution typically incurs an interaction between participants to share a common understanding.

The inconsistencies between operation definition models can be automatically identified and presented to the involved participants. In [10], we present an add-on developed on top of a commercial BP modeling and analysis tool, which allows participants to analyze the expectations and possible inconsistencies at anytime during process definition.

Integration and Change. Once the role-operation models are correct, complete and consistent, i.e., all expectations of roles are satisfied within and all individual models are verified and validated, the organization has a set of models that implicitly or explicitly convey a great amount of information regarding how the organization operates.

Based mainly on the operation definition diagrams, a variety of models can be generated, each presenting the process information from different perspectives and in different abstraction levels. Each model is a query to the process-base that visualizes a portion of the processes from a specific perspective. A generated model is valid until a change is performed to the models that form the base for its generation.

The changes regarding the behavior depicted in operation definitions are made by process participants. With respect to the principle of encapsulation, if a change does not affect the interface of the role, it is an alteration in role's context and does not affect the interaction between the roles and the way they perform their tasks. If an update modifies the role's interface (and thus its expectations), the change should either be incorporated in all related models or it should be revoked after negotiation between parties. Such cases manifest themselves as inconsistencies between expectations and resolved in the relevant models.

As an output, the Plural method generates a set of models that depict process relevant information in different forms. The Plural is developed to facilitate the modeling and visualization of processes as well as help validating and maintaining them. Currently, however, it does not incorporate mechanisms to support the enactment of the defined processes based on executable definitions generated from these models.

3 The Plural Notation

Common business process modeling notations such as Business Process Model and Notation (BPMN) or Event-Driven Process Chains (EPC) do not explicitly address the issue of multiple modelers and are not readily applicable for subject-oriented approaches without significant refinements and extensions. In order the Plural method to be effectively applied for process modeling, it should be coupled with a notation

that can reflect the unique characteristics of the underlying principles that root back to the notion of subject-orientation. Moreover, the notation should be able to capture not only the behavior aspects of the processes but also the static relationships between subjects (organizational) and data/information objects (informational). It should be simple to facilitate modeling by the process participants, who are very rarely experts on process modeling.

We strived to use a minimum set of process elements to help models to be simple and readily understandable by all involved parties. Yet the notation incorporates a range of diagrams that capture and present process information from different viewpoints. Only a subset of these diagrams is key to the appropriate implementation of Plural and many of the optional diagrams can be automatically generated based on available models.

The diagrams are principally based on Unified Modeling Language (UML) and refined EPC notation. We have chosen UML use case diagrams for representing the high-level functional aspect of processes mainly due to its unique ability to allow for capturing inherent generalization and composition relationships between process functions. We use UML class diagrams for capturing the static relationships (including inheritance, aggregation, and association) within information items and within organizational roles. The class diagrams are one of the most commonly used representation tools for capturing such interrelationships between concepts and are the main building block of object-oriented modeling . items informational, and organizational aspects of processes. ,for the behavioral aspect. Table 1 lists the diagrams, the phase where they are applicable in, the Plural roles that are responsible for their creation and maintenance, and their significance for a successful implementation of the method.

Table 1. The Plural diagrams.

Plural Phase	Diagram	Description	Plural Role	Criticality
Context Definition	Scope Diagram	Processes and participating roles	All / Coordinator	Key
	Role Diagram	Roles and their static relationships		Key
Description and Conflict Resolution	Operation Definition Diagram	Behaviour description of a single (role) operation	Process Participants	Key
	Information Item Diagram	Static structure of information items/business objects		Optional
Integration and Change	Process Model (Operation-, Activity- or Context-level)	Integrated process models on the operation, activity or process levels.	All / Coordinator	Partly Optional
	Role Dependency Diagram	Role dependencies in terms of message (information item) exchange		Optional
	Process Dependency Diagram	Process dependencies in terms of message (information item) exchange		Optional

3.1 Scope Diagram

A scope diagram (together with the *role diagram*) defines *what* the organization does (will do) in high-level based on its vision and goals and how it is (will be) organized to do it. It represents the voice of the top to guide and provide direction, coordination and a sense of discipline for bottom-up modeling phase that follows. These processes to be covered are decided by all process participants and relevant process stakeholders. Together with the role diagram, a scope diagram offers a high-level picture, helps to coordinate involved parties, and provides a skeleton to be advanced from bottom and filled through individual modeling in the description and conflict resolution phase.

A scope diagram portrays the processes to be covered, their relationship and the roles (subjects) that participate in these processes. Underlying principles of scope diagrams are grounded on UML use case diagrams [11]. Fig. 2 gives an excerpt from a scope diagram, and depicts the diagram elements and their relationships.

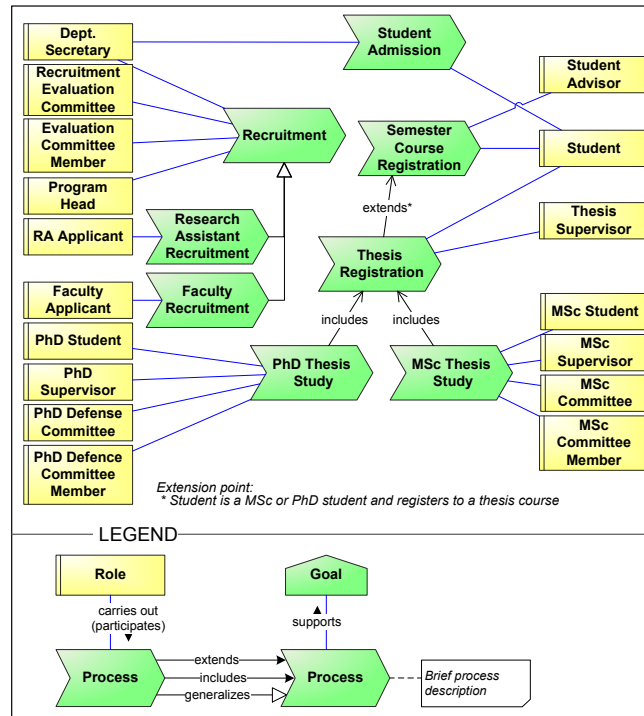


Fig. 2. A Scope Diagram.

A connection between a role and a process implies that the role is participating in the process by offering one or more operations to serve the process goal. The *include* relationship between a process represents a reusable process that is unconditionally incorporated into another one. The *extend* relationship, on the other hand, is the incorporation of the reusable processes, depending on certain conditions. *Generalize*

relationship is ‘a type of’ relationship in which case the basic functionality of the *general* process can be the same, but the *specific* process may differ in some aspect.

3.2 Role Diagram

There are inherent structural relationships between roles (subjects) and these relationships are depicted in a role diagram. Capturing these relationships is important particularly in identifying the inconsistencies in terms of the messages they exchange between each other. Similar to the relationships in UML class diagrams, relationships between roles can be of association, aggregation (composition) and generalization type. Process participants that act for the roles can also be depicted in role diagrams.

An excerpt from an example role diagram is depicted in Fig. 3-a. Role diagrams may also represent the operations offered by each role (Fig. 3-b).

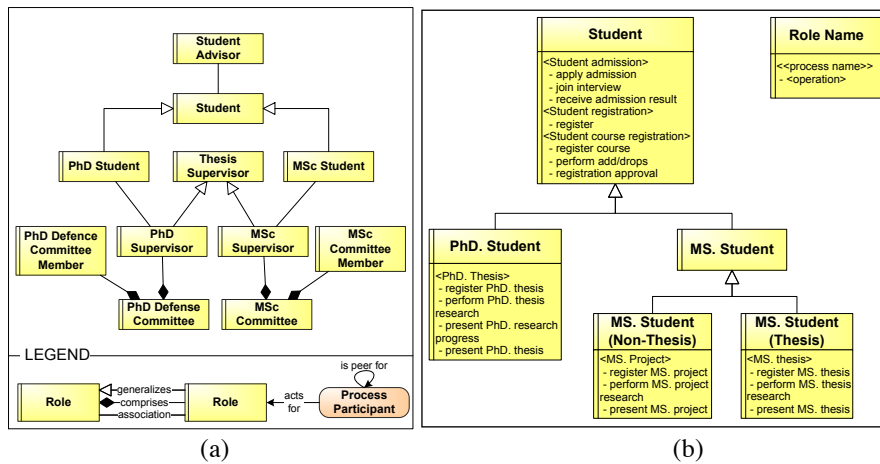


Fig. 3. (a) A role diagram. (b) An extended role diagram.

3.3 Operation Definition Diagram

An operation definition diagram portrays in detail how a particular role-operation executes, what inputs it gathers and outputs it produces from/to its environment. In particular, a process participant describes the activities performed by its role in that operation, the information items it requires while performing these activities and the outputs it produces. In addition, participants represent the sources of the inputs and the destination of the outputs. The sources might represent other roles or items, such as project repositories, folders, software tools or other operations of the same role. Participants also represent the activities that their roles perform together with other roles. As discussed above, these representations of interactions form the expectations of that role from other roles.

Operate definition diagrams are based on extended Event Driven Process Chain (eEPC) diagrams. We use a columnar (swimlane) view in which the process participant depicts the activities performed by his/her role in the primary swimlane. Activities in other swimlanes are not allowed in operation definition diagrams but used to represent the interacting parties.

Fig. 4 gives two examples of operation description diagrams for the *project proposal preparation* process (modeled in one of our case studies). Fig. 4 (a) shows the ‘*organize first meeting*’ operation of the ‘*interviewer*’ role, while Fig. 4 (b) depicts ‘*perform first meeting*’ operation of the same role.

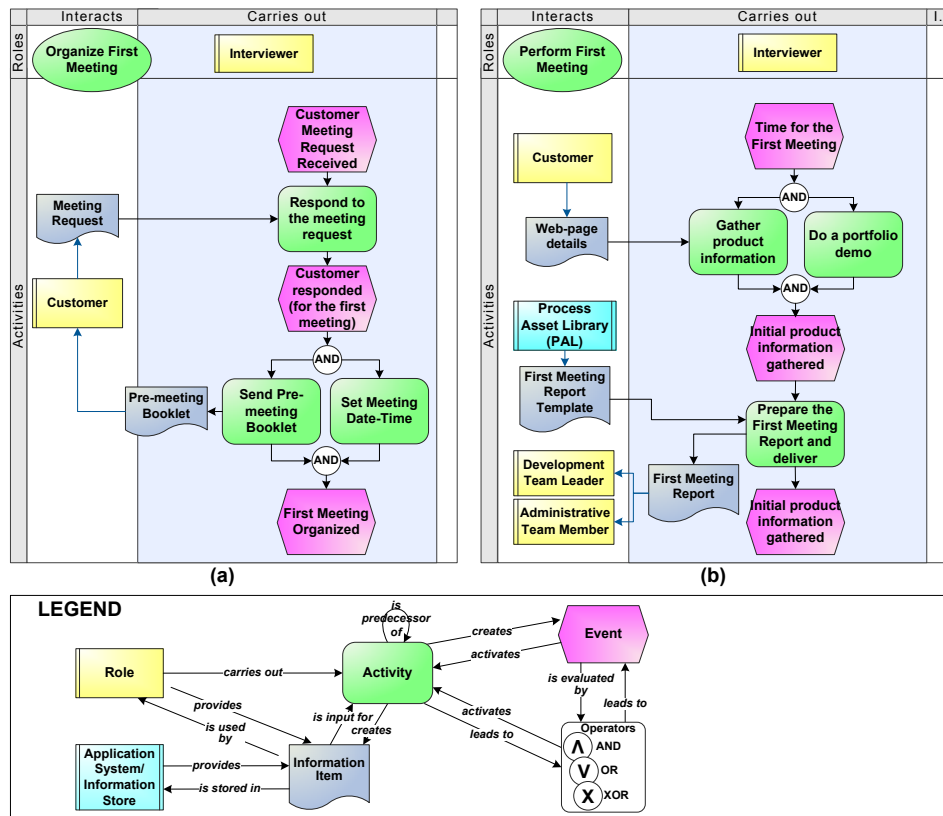


Fig. 4. Examples of two operation definition diagrams

3.4 Information Item Diagram

Information item represents a means to store and transmit information. The information typically relates to a business objects that is being manipulated or business events. An item can be in the form of a document, an email, a fax, a CD, or a verbal message that is input to or output from an activity, or a resource retrieved from an information store. An information item can be:

- an *input* to an activity and consumed during its processing; or
- an *output* as a final object produced or resulted out of an activity; or
- a *resource* that is fed into a process and used as a part of the transformation process. Unlike inputs, information resources are not consumed. Templates or standards are examples of this kind.

The information item diagram represents the static relationships between information items. Similar to the role diagrams, the relationships between information items can be in association, generalization (is type of), and aggregation (composition) type. Fig. 5 presents examples of information item diagrams.

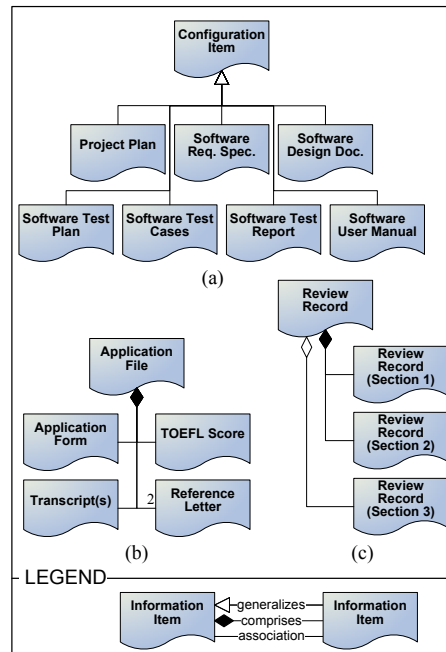


Fig. 5. An information item diagram.

As we discussed above, expectations are represented in terms of the information exchanged between roles. For example, there is an inconsistency in the definition if an information item that is expected by a role is not provided by the expectant role. In that respect, it is sometimes necessary to capture the relationship between information items to, for example, accommodate for the differences in the abstraction level used by different process participants in modeling. This also allows them to continue using their own vocabulary, instead of obliging them to adapt others' just to preserve consistency between models. For example, in an employee recruitment process, the *HR manager* may process the inf. item 'application folder' as a whole in its operations, while the *HR specialist* may need to process and refer each of its section (e.g. *application form*, *reference letter*, etc.) separately. Accordingly, either HR manager or HR specialist may define an inf. item diagram to represent this relationship (as shown in Fig. 5-b).

Similar to operation definition diagrams, inf. item diagrams are modeled by process participants during the description and conflict resolution phase.

3.5 Process Model (Operation-, Activity-, or Context-level)

Correct, complete and consistent operation definition models (together with scope, role and information item models) that are produced in prior Plural phases carry valuable information about what processes are carried out, which roles participate in these processes, what information a role needs, when it needs it, as well as how it acquires it. Based on these definitions, diverse types of models with different abstraction levels can be generated in the integration and change phase based on the *operation definition models*.

A generated model is a snapshot of the process-base at any point in time and valid until a change is performed to the (operation definition) models that form the base for its generation. When a change is performed, the model should be re-generated in order to reflect the change. The model generation can be fully or partially automated using appropriate tools. The details regarding the rules and techniques for generating process and dependency models are presented in [12].

Operation-level process diagrams depict the role operations and the information exchange between them that take place within a particular process. It shows the behavior of a process in such a level that provides sufficient detail to understand the general flow and message exchanges between process roles and external entities. Fig. 6 depicts an operation-level process diagram *project proposal preparation* process.

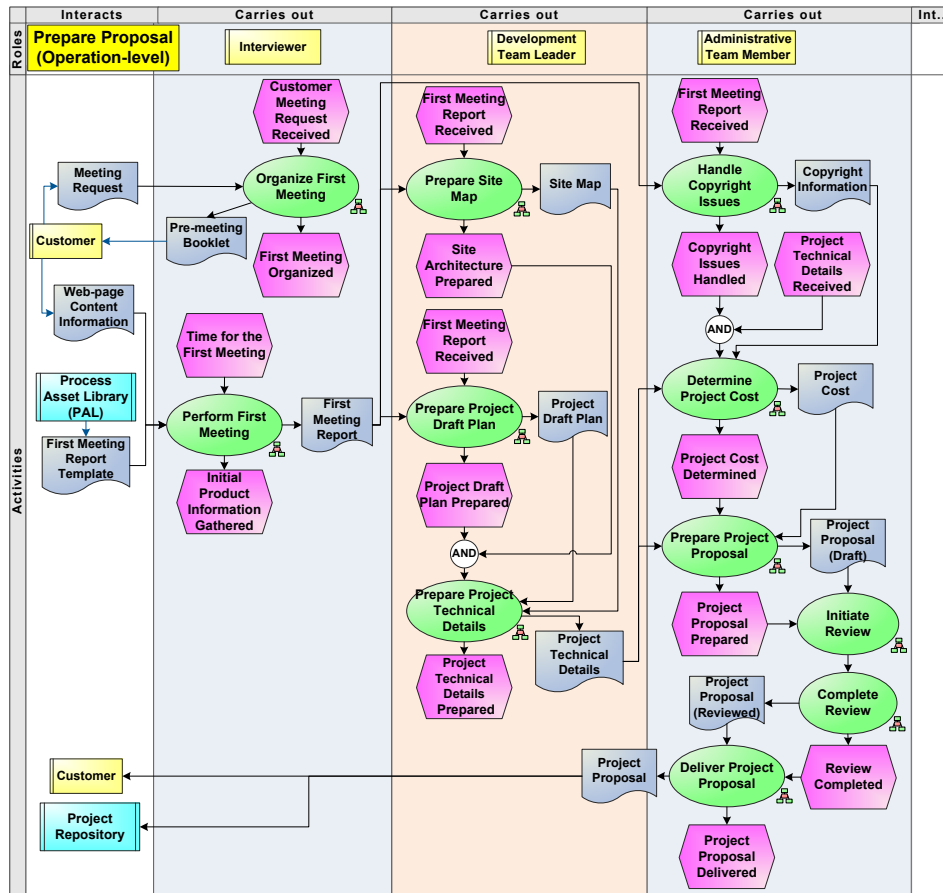


Fig. 6. An operation-level process diagram (for *project proposal preparation* process)

It is also possible to integrate operation definition models into a diagram that depicts all the activities performed in that process at the lowest level of detail. We call these models *activity-level process diagrams*. The generation of these diagrams requires each operation definition models of that process being placed side-by-side and joined with the message exchanges among roles and other entities.

The *context-level process diagrams* depict the data exchange between a specific process and entities external to the process (e.g.: other roles, application systems, information stores, etc.) at the highest level of abstraction. An example model is presented in Fig. 7.

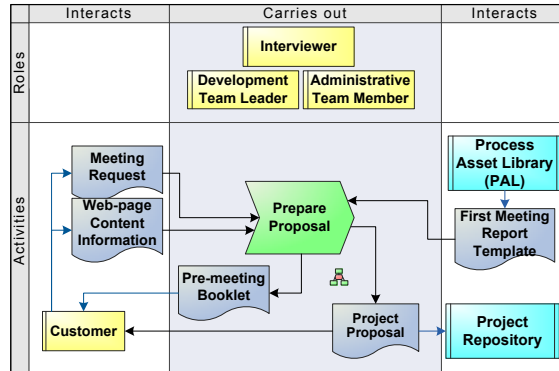


Fig. 7. An context-level process diagram (for *project proposal preparation* process)

3.6 Role and Process Dependency Diagrams

Interactions between different roles, processes, and other external entities in terms of the message exchanges can be captured using dependency diagrams. This type of diagrams helps organization to understand the interactions between different entities in the organizational and the implications of changing these interdependencies.

Role dependency diagrams depict the interaction between a set of roles while process dependency diagrams show the message exchanges between a selected set of processes. Similar to process models, these diagrams can also be generated based on the operation definition models. Fig. 8 depicts an example of a role dependency model within a particular process. The scope of this dependency model can be extended to cover multiple processes.

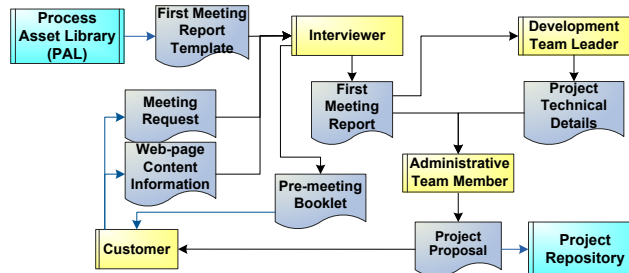


Fig. 8. An context-level process diagram (for *project proposal preparation* process)

4 The Multiple Case Study

In this section, we present a multiple case study with *three* cases that we conducted to understand the applicability of the Plural method and its components. We aimed to uncover the benefits that an organization can gain from applying the Plural method

together with its notation and the tool. We projected benefits in several directions. We expected the Plural method to help:

- i. Capture a wider extent of process information and yield more complete process models,
- ii. Discover the interaction points, expectations, and conflicts between process participants,
- iii. Incorporate change easily, and in turn assist evolving processes,
- iv. Decrease the total duration needed to model processes.

We were not able to design our case studies to capture quantitative or qualitative evidences to empirically validate the materialization of all benefits listed above. However, in this section we will summarize the evidences we collected towards validating the benefits ‘i’ and ‘ii’. For benefits ‘iii’ and ‘iv’, we discuss the underlying rationale for our claims based on related works in the literature.

The *first* case study was conducted in a graduate school and included mostly its administration processes, such as staff recruitments, student admissions and enrollments, etc. The *second* case study was performed in a small software company that develops software and provides consultancy and training to other software developing companies. The *third* case was performed in a small-scale web design and development company, involving processes such as project management, design, prototyping, and deployment.

We presented our findings in the first two case studies in our previous works [13] and [10]. In this paper, we discuss the results in our third case study and provide a synthesis of the findings for our multiple case study. Table 2 presents the properties regarding the extent and type of processes covered in the cases,

We acted also as observers during the case study conducts to gather information regarding the difficulties the participants and coordinators faced in following the method, their reactions in certain situations (such as in inconsistencies and conflicts), and the method they followed in resolving such issues. The case study participants were asked to record a set of measures including the time spent for specific activities. Finally, participants were interviewed to investigate further on the benefits and difficulties they faced. During the interview, they were given a questionnaire to augment the discussions and to elicit further feedback on the method.

The tool used in these studies was the ARIS Collaborative Suite¹, and its web based plugin (Web Designer) that enabled modeling through web browsers. The models resided in a central repository, which could be accessed by ARIS client applications and web designer through Internet. The tool was extended with an add-on that analyzes process repository to detect and present inconsistencies between operation definition models.

Table 3 presents the scope of the case studies, the effort utilized and the total duration.

Table 2. Multiple Case Study Properties.

Case1	Case 2	Case 3
-------	--------	--------

¹ ARIS Platform latest version: http://www.softwareag.com/corporate/products/aris_platform/

	Case1	Case 2	Case 3
Case Location	Graduate School	Software Dev. & Consultancy Company (small-scale)	Web-based Application Dev. Company (small-scale)
Process / Process Areas	Student admissions, enrollments, staff recruitments, etc. (12 process areas)	Project management, review, training, configuration-change management (5 process areas)	Project development, RFP preparation, project mng., requirements gathering, deployment (5 process areas)
Team	- 6 process participants - 2 Coordinators (4 familiar with process modeling)	- 4 process participants - 1 Coordinator (all familiar with process modeling)	- 4 process participants - 1 Coordinator (limited knowledge on process modeling)
Tools	- ARIS Collaborative Suite – Web Designer (Ver.7.01) - The Plural add-on on ARIS tool to: <ul style="list-style-type: none"> • Present inconsistencies between operation definition models • Help model integration and generation 		

Table 3. Multiple Case Study - Quantified Results.

	Case1 Graduate School	Case 2 Software Dev. & Co. Company	Case 3 Web-based App. Dev. Company
Process (areas)	12	5	5
Participating Team	6 particpt. / 2 Coord.	4 particpt / 1 Coord.	4 particpt. / 1 Coord.
Roles identified	30 (13 in-scope)	18 (15 in-scope)	9 (8 in-scope)
Role operations	78	48	13
Total Effort (person-hour)	128	40	42
Context Definition:	18	10	9
Definition and Conflict Res.:	90	25	29
Process Participant 1:	9.0	9.0	14
PP 2:	5.0	5.0	10
PP 3:	20.5	2.5	2
PP 4:	13.0	2.5	1
PP 5:	11.5	-	-
PP 6:	17.0	-	-
Coordinator 1	12.0	6	2
Coordinator 2	2.0	-	-
Integration:	20	5	4
Total Duration (hour)	40.5	18	19
Context Definition:	6.0	4	3
Definition and Conflict Res.:	20.5	9	14
Integration:	14.0	5	2

4.1 Findings

Above, we discussed the benefits we expected in applying the Plural method. *First*, we projected the Plural method to help capture a wider extent of process information and yield more complete process models. The Plural method enables organizations to capture the perspectives of multiple process participants, which are partial. The approaches considering process information captured from multiple perspectives (or viewpoints) help model completeness [14]. Hence, capturing and representing process information from multiple viewpoints enables organization to capture wider and accurate process information. Explicit modeling of the information flow between organizational entities contributes further to the accuracy of the process models.

For many of the processes covered in the case studies, the organizations already had process descriptions mostly in text descriptions. During the case study conducts, we observed several cases where participants identified problems stemming from ambiguous and incomplete process definitions as well as implicit assumptions, which are uncovered when participants started modeling the information flow as expectations. Despite the deficiencies of current process definitions, the processes were performed in one way or another, as the participants had tacit knowledge to handle the ambiguities and fill the gaps.

On the other hand, while we observed improvements in process models in several directions, it is difficult to argue that these improvements originate merely from the implementation of the Plural method and may not surface if another redesign or improvement approach was applied.

As a *second* point, we expected to discover the interaction points, expectations, and conflicts between process participants. The interaction points between roles are one of the most fragile points in processes performed by knowledge workers and are potential locations in identifying implicit assumptions of process participants [15]. The Plural ensures that a role's expectations (in the form of information flow) are explicitly defined and are visible to the organization. It is shown that business process models that are rich in presenting communication flows and interactions between organizational entities can help process redesign practitioners identify key problems, particularly in information-intensive processes [16]. This helps discovering communication points and conflicts, and provides an explicit representation of unfulfilled expectations where key dependencies, and individual and process-level goals are not being achieved satisfactorily.

To verify the ability of Plural to discover expectations and conflicts, we sought diverse sources of evidence. These sources included observations from the execution of the case studies to locate traces of hidden assumptions and conflicts revealed through the use of Plural method; examining the differences between existing prior process definitions and the new definitions; and observing elements that had been missed and captured.

Thirdly, we claimed that it would be easier to maintain and incorporate change to the process definitions built using the Plural method. In the case studies, the process definitions were structured into loosely coupled role operations encapsulating certain behavior of a role. By encapsulation, the information regarding the internal workings of role's operations was hidden to outside world, yet its interface – i.e., mainly its inputs and outputs- was visible as expectations. Any change regarding the internal

behavior of a role was an alteration on how the operation is offered and did not influence the inner workings of other operations. Whereas, a change on role's interface generated an inconsistency in the process-base, which was made visible instantly and solved by the participants. Some of these changes involved further alterations propagating over other descriptions.

The claim to increase maintainability is closely related to our underlying hypothesis regarding the benefits of following a subject-oriented approach - such as the Plural. Shifting from traditional procedural to subject-oriented modeling paradigm promises quality increases in the resulting process models. Defining processes as a set of interacting subjects (roles) through the operations they provide to the organization offers improvements on critical aspects of the process quality, such as maintainability, complexity, and understandability. Here, it is important to stress again the ability of organizations to adapt rapidly to changing business requirements. Traditional process modeling follows a behavioral approach and is grounded mainly on procedural paradigms. However, it is proven that applications developed using object-oriented approaches have higher maintainability than those developed using procedural approaches [20]. This is mainly due to the characteristics of O-O approaches, such as the information hiding and encapsulation. In Plural, size and complexity of process models also decrease as a process model is inherently structured into a set of operations (modules), which tend to be cohesive and reduced in size and complexity. As the size of a process model decreases, the likelihood of the model to have errors decreases [21]. The modular structure also increases the understandability of a process model [22]. In that respect, we believe that process models developed and structured using subject-oriented approaches (which are based on object oriented approaches), such as Plural, is likely to produce process models that are relatively easier to understand and maintain.

Finally, we expected to observe decreases in the total duration needed for modeling processes. However, it is reasonably difficult (if possible) to set up a laboratory environment to experiment and compare the Plural method against other conventional methods in terms of the values for the total effort and efficiency. Nevertheless, since the effort for modeling is shared among the individuals, as the number of the agents participating in concurrent process modeling increases, the probability of having a reduction in total duration increases (the upper limit for the concurrent development agents being the number of roles covered within the scope). Accordingly, the total time for the description and conflict resolution phase becomes the time committed by the participant that utilized the highest amount of effort for that phase.

Table 3 shows the effort and duration values for the case studies. The cases covered a significant portion of the processes carried out in the organization and yet the total duration for the cases were 40.5, 18, and 19 hours, respectively. In other words, we were able to decrease the process improvement cycles in the order of days.

The case studies also revealed some limitations of the approach. We observed that the efficiency and effectiveness of the modeling increases if the process participants are accompanied by their peers or coordinators and do – what we may call- *pair modeling*. It is also observed that participants' process modeling skill influences the success, but their willingness to collaborate influences more.

We also observed that the maturity of the organizations, in terms of its process culture, may influence the success in following the Plural approach. The more mature the organization is, the more benefit it is likely to get by following the Plural method.

The tool support for the method is critical to streamline its application. Without a tool that addresses the unique requirements of the approach, it is not possible to get the claimed benefits as expected.

5 Conclusions

We developed the Plural method to foster an approach that emphasizes the empowerment of process participant to define their processes, and to provide them necessary mechanisms and tools to maintain these definitions as loosely coupled process components (role operations). Our findings are based on the analysis of the case study results provide evidence that organizations can benefit from following the Plural approach.

Plural puts the process participant at the hearth of process management and facilitates empowerment of these individuals. Giving the chance (and responsibility for) process participants to think about and define their processes facilitates also the *process change* to be owned and performed by them. This helps to increase employee involvement, which in turn facilitates a culture that fosters problem solving and process improvements. Participation and commitment allows employees to make decisions themselves and in turn enhances the ability to reorganize rapidly to adapt to changes in the environment [17]. It creates an environment of ‘ownership’, allowing rapid exchanges of innovative ideas [18]. Aligned with the shift from ‘command-and-control’ structures to ‘coordinate-and-cultivate’ management, such an environment fosters the establishment of decentralized structures of loose hierarchies and democracies centered around enduring human values [19]. Process modelling for these people should not be seen as a burden on daily work activities but a chance (as well as a duty) to reflect on the way they perform their work and improve it.

We see a great future in the BPM approaches that facilitate the empowerment of individuals and communication between them, thereby allowing ‘decentralization with co-ordinated control’ [23]. We believe that research contributing towards these directions will have significant impact in practice.

References

1. Drucker, P.F.: The New Society of Organizations. Harvard Business Review. Sept-Oct, 95–104 (1992).
2. Senge, P.: It’s the Learning: The Real Lesson of the Quality Movement. Journal for Quality & Participation. 22, (1999).
3. Lucas, B.: The Imperative of having an Agile Organization Structure, <http://www.compaid.com/caiinternet/ezone/AgileOrg-Lucas.pdf>.

4. Ould, M.A.: *Business Processes - Modelling and Analysis for Re-engineering and Improvement*. John Wiley & Sons, Chichester, UK (1995).
5. Fleischmann, A.: *What Is S-BPM? S-BPM ONE—Setting the Stage for Subject-Oriented BPM*. pp. 85–106. Springer Berlin Heidelberg (2010).
6. Hwang, M.I., Thorn, R.G.: The effect of user engagement on system success: A meta-analytical integration of research findings. *Information & Management*. 35, 229–236 (1999).
7. Kasse, T.: *Practical Insight into CMMI*. Artech House (2004).
8. Singer, R., Zinser, E.: *Business Process Management — S-BPM a New Paradigm for Competitive Advantage?* In: Buchwald, H., Fleischmann, A., Seese, D., and Stary, C. (eds.) *S-BPM ONE – Setting the Stage for Subject-Oriented Business Process Management*. Springer, Berlin, Heidelberg (2010).
9. Kurz, M., Fleischmann, A.: *BPM 2.0: Business Process Management Meets Empowerment*. *S-BPM ONE 2010, CCIS 138*. pp. 54–83 (2011).
10. Turetken, O., Demirors, O.: Plural: A Decentralized Business Process Modeling Method. *Information & Management*. 48, 247–235 (2011).
11. Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G.: *Object-Oriented Software Engineering - A Use Case Driven Approach*. Addison-Wesley (1992).
12. Turetken, O.: *A Method for Decentralized Business Process Modeling*, PhD. Thesis, <http://etd.lib.metu.edu.tr/upload/12608496/index.pdf>, (2007).
13. Turetken, O., Demirors, O.: Process modeling by process owners: A decentralized approach. *Software Process: Improvement and Practice*. 13, 75–87 (2008).
14. Darke, P., Shanks, G.: User viewpoint modelling: understanding and representing user viewpoints during requirements definition. *Information Systems Journal*. 7, 213–219 (2008).
15. Brady, S., DeMarco, T.: *Management-aided Software Engineering*. *IEEE Software*. 11, 6–25 (1994).
16. Kock, N.F.: *Systems Analysis & Design Fundamentals: A Business Process Redesign Approach*. Sage Publications, Inc (2006).
17. Chen, C.-J., Huang, J.-W.: How organizational climate and structure affect knowledge management—The social interaction perspective. *International Journal of Information Management*. 27, 104–118 (2007).
18. Chang, L., Powell, P.: Towards a Framework for Business Process Re-engineering in Small and Medium-sized Enterprises. *Information Systems Journal*. 8, 199–215 (2002).
19. Malone, T.W.: *The Future of Work: How the New Order of Business Will Shape Your Organization, Your Management Style, and Your Life*. Harvard Business School Press (2004).
20. Henry, S., Humphrey, M.: Comparison of an Object-Oriented Programming Language to a Procedural Programming Language for Effectiveness in Program Maintenance, Technical Report TR 88-49. Computer Science, Virginia Polytechnic Institute and State University (1988).
21. Mendling, J.: Metrics for Business Process Models. In: van der Aalst, W., Mylopoulos, J., Sadeh, N.M., Shaw, M.J., and Szyperski, C. (eds.) *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*. pp. 103–133. Springer-Verlag, Berlin, Heidelberg (2008).
22. Reijers, H.A., Mendling, J.: A Study Into the Factors That Influence the Understandability of Business Process Models. *IEEE Transactions on Systems, Man, and Cybernetics*. 41, 449–462 (2011).
23. Sloan Jr., A.P.: *My Years with General Motors*. Doubleday (1963).