Do Process-based Systems Support Emergent, Collaborative and Flexible Processes? Comparative Analysis of Current Systems

Hanane Ariouat*, Eric Andonoff, Chihab Hanachi

IRIT, Université Toulouse 1-Capitole, 2 rue du Doyen Gabriel Marty, 31042 Toulouse Cedex, France

Abstract

Advanced process-based applications such as crisis and risk management require Emergent, Collaborative and Flexible (ECF) processes. These three features correspond to specific real-world requirements. Firstly, to face unpredictable situations, emergent processes are needed, *i.e.* processes whose model/schema is (partially) unknown at design-time and gradually defined/refined at run-time. Secondly, the complexity of the application domain being addressed requires collaborative processes, *i.e.* processes whose definition and enactment require interactions between groups of actors using their skills and experiences to make the process convergence towards their common goal. Thirdly, to adapt to the frequent changes that occur in their operating environment, flexible processes are needed, *i.e.* processes whose model/schema can be modified in real time. The objective of this paper is to present a literature review to study the state of the art of process-based systems with a focus on the three above mentioned properties. Our approach is to specify real world requirements, then to discuss the adequacy of current systems to these requirements and finally to provide advices for their improvement. Examined systems fall within the following areas: Business Process Management (BPM), Adaptive Case Management (ACM), Computer Supported Collaborative Work (CSCW) and Knowledge Management (KM). The paper concludes on the insufficiency of existing systems and the need for designing and implementing a specific process-based system that integrates these three properties in a coherent framework.

Peer-review under responsibility of KES International

Keywords: Emergent, Collaborative and Flexible Processes; BPM; ACM; KM; CSCW; Context-aware Processes; Knowledge-aware Process;

1. Introduction

Advanced process-based applications falling within crisis and risk management, virtual organizations, or factory of the future, manage processes with specific features. In particular, such applications help companies to coordinate the different actors involved in their processes by automating repetitive tasks and facilitating the distribution of

* Corresponding author. Tel.: +33-561633563
E-mail address: Hanane.Ariouat@ut-capitole.fr

information and control in distributed settings. Firstly these processes are *emergent*, as their model/schema is (partially) unknown at the beginning and gradually defined/refined during execution: process definition and execution intertwine. Secondly, these processes are *collaborative* as they require skills and experiences from several groups of actors both for their modelling and execution: each group of actors is responsible for modelling parts of the process according to its skill area and the whole group is responsible for the synchronization of each group contributions. To sum up, both process modelling and execution are actor-driven, these actors focusing on a common goal and the modelled and executed process being the result of a consensus between them. Thirdly, we can mention that such processes operate in dynamic environments and are subject to frequent changes. As a consequence, they should be *flexible i.e.* able to respond to changes occurring in their operating environment.

For example, let us consider a crisis management application [1]. In this application, several participating actors (experts) have to act simultaneously and urgently to reduce the crisis and its impacts on the real world. To achieve this common goal efficiently, these actors must collaborate, or at least act in a coordinated way in order to make their activities as efficient as possible. Designing this coordination is a difficult task. It requires the use of business skills of the involved groups of actors, who are responsible for defining actions to be undertaken on the ground and their coordination according to their areas of expertise, and the process driving crisis resolution merges the defined actions from all relevant actors. Thus a crisis resolution process schema is the result of a collective and consensual modelling work between actors involved in crisis resolution. On the other hand, such processes are subject to frequent changes to meet the evolving requirements of the crisis. In addition, these changes may be performed while there are ongoing process actions on the ground. This means that design and execution of such processes are strongly intertwined: actors are allowed to design the schema, *i.e.* the actions and their coordination, of the considered process while executing it.

Advanced process-based applications should benefit from existing process-based systems relevant from the four following complementary areas: Business Process Management (BPM), Adaptive Case Management (ACM), Knowledge Management (KM) and CSCW (Computer-supported Collaborative Work). However process-based systems from these areas lack in meeting the following requirements, derived from the three above mentioned properties:

- need for process flexibility as well as a strong intertwining of process design and process execution,
- need for human and collective dimension integration in order to support collaborative design and execution of the process driving these applications (*e.g.*, the crisis resolution process),
- need for context and knowledge support to both describe the current situation (*e.g.*, the impacted ground and its evolution in a crisis application) and the already existing or created knowledge to deal with such a situation (*e.g.*, reusing part of past experiences in a crisis application) in order to provide actors involved in this collaborative design and execution with the relevant information and data.

The objective of this paper is to present a literature review to study the state of the art of process-based systems with a focus on the three above mentioned properties. Our approach is to specify real world requirements, then to discuss the adequacy of current systems to these requirements and finally to provide advices for their improvement. More precisely, the contribution of the paper is a critical and comparative analysis of the main process-based systems in BPM, ACM, CSCW and KM areas, considering the above mentioned requirements. We have carried out this analysis in three steps answering the three main following questions: (i) do examined process-based systems support the required flexible dimension, *i.e.*, process flexibility, as well as process design and execution intertwining, (ii) do examined process-based systems support the required collaborative dimension and finally (iii) do examined process-based systems support the modelling of the required context and knowledge dimension, so that decision-making is make easier.

Accordingly, the paper is organized as follows. The next three sections give an overview of the examined process-based systems addressing respectively the flexible, the collaborative, and the context and knowledge dimensions support in processes. Section 5 presents a comparative analysis highlighting the benefits of examined process-based systems to support collaborative and emergent processes with respect to the previous requirements but also highlighting their lacks within a comparison grid. This section also positions collaborative and emergent processes with respect to examined process-based systems areas, thus underlying that none of the examined systems support collaborative and emergent processes. Finally, Section 6 concludes the paper and gives some directions for future works.

2. Flexible Dimension Support in Process-based Systems

The section discusses flexible dimension for processes according to process flexibility and process design and execution intertwining.

2.1. Flexibility of Processes

The flexibility of a process is related to its ability to respond to changes occurring in its operating environment. According to the taxonomy presented in [2], we distinguish between *flexibility at design-time*, which refers to foreseeable changes which can be taken into account in modelled process schemas, and *flexibility at run time*, which refers to unforeseeable changes occurring during process execution. In addition, this taxonomy identifies four needs of flexibility:

- *variability*, for representing a process differently, depending on the context of its execution,
- *adaptation*, for handling occasional situations or exceptions that have not been necessarily foreseen in process schemas,
- *evolution*, for handling changes in processes, which require occasional or permanent modifications in their schemas,
- *looseness*, for handling processes whose schemas are not necessarily known at design-time and which have to be defined at run-time. Such processes require loose specifications.

Process flexibility has been the focus of many contributions in BPM area. The main BPM systems addressing this issue are ADEPT, YAWL†, DECLARE‡ and FLOWER[§]. ADEPT [3] advocates a procedural approach for process modelling: activities of the process and the way these activities are synchronized are explicitly modelled within a process model. Regarding process flexibility, ADEPT supports ad-hoc modifications of running process instances on-the-fly (*e.g.*, to dynamically add, delete or move process fragments), and process model change with instance migration using versions. Thus ADEPT supports flexibility by adaptation and evolution. Regarding process model change, we can also mention [4], which recommends a set of 14 adaptation patterns to structurally change process models. These patterns correspond to high-level primitives making the update if process model easier to perform.

YAWL [5], which also advocates a procedural approach for process modelling, addresses flexibility by variability, adaptation and looseness. YAWL provides support for flexibility by looseness through the notion of *worklet*, an extensible repository of self-contained sub-processes and associated selection rules, grounded in a formal set of work practice principle derived from Activity theory. Thus process designers can model activities which will be substituted at run-time with a dynamically selected worklet, according to selection rules and contextual data values [6]. Regarding flexibility by looseness, we can also mention [4], which recommends a set of 4 change support feature patterns to allow process actors to add information regarding unspecified parts of process models at run-time. YAWL also provides support for flexibility by adaptation as process designers can define exception handling processes (called *exlets*) to be invoked when certain events occur in processes [7]. These exception handling processes can be defined at design-time but they can also be added at run-time in order to really support exception handling. Finally YAWL also addressed flexibility by variability. Indeed, YAWL supports variants modelling as adjustment of configurable process models.

Unlike ADEPT and YAWL, DECLARE [8] advocates a declarative approach for process modelling: activities of the process and constraints existing between these activities are defined when modelling the process and any

[†] http://www.yawlfoundation.org/
[‡] http://www.win.tue.nl/declare/
[§] http:// www.pallas-athena.com/

execution of activities that does not violate constraints is possible. Such an approach for process modelling, with a non-explicit representation of process model, makes process flexibility easier to achieve. Indeed DELARE achieves flexibility by looseness, including late binding and late modelling, in a dynamic way when users involved in process execution decide whether to invoke an activity or a sub-process. DECLARE also provides support for flexibility by evolution by allowing change operation of running processes: constraints and activities can be added or removed. Finally DECLARE supports flexibility by adaptation distinguishing between mandatory and optional constraints: mandatory constraints must obviously be fulfilled during process execution while optional ones can be violated. The main drawback of this approach is, from user point of view, the usability of the modelling approach.

FLOWER is an Adaptive Case-Management system that allows deviations from process models. More precisely FLOWER is strongly based on data, is case-based (*i.e.*, considers process instances) running according to a process schema and provides users with assistance rather than guidance, focusing on what can be done instead of what must be done, and allowing deviation within specific activities: *skip* to skip activities that should be performed, *redo* to re-execute activities already performed and *open* to execute activities that are not supposed to be performed yet [9].

2.2. Process Design and Execution Intertwining

However, the flexible dimension of a process can also be related to the ability of the system executing it in supporting a strong intertwining of process design and process execution. Indeed, when considering crisis emergent processes, groups of actors define activities to perform on the ground and adapt this definition according to crisis evolution adding existing or created activities, deleting, moving or interrupting existing ones, or updating activities synchronization, while executing the process itself.

In the following, we mention two contributions allowing a step-by-step definition of processes. CARAMBA [10] is a process-aware CSCW system supporting ad-hoc and semi structured processes, *i.e.*, processes that are not based on predefined process schemas but rather created on-the-fly. Such processes are defined according to project needs, adding new activities or existing activities modelled as process templates implementing good practices. CARAMBA acts as a recommendation system helping user in choosing the next action to be executed, which can either be an action of a template or an already modelled action, independent from any template. However [10] does not clearly discuss about process design and execution intertwining. More precisely, CARAMBA supports step by step execution of process instances, but does not intertwine design and execution as added activities/actions that are previously modelled. We also mention [11], a contribution implemented on top of BONITA** BPM system, which gives up the notion of process schema to consider only (instances of) processes to support coordination of cooperative activities. These process instances are designed by users who have the ability to define them by derivation or composition of existing ones. In addition, at run-time the process owner can modify them by adding new activities or activities referring process fragments. Thus [11] provides a mean to intertwine design and execution of process instances, but this intertwining is driven by process owners. Finally, CAMUNDA†† supports a step-by-step process execution in which the addition of existing activities is allowed.

3. Collaborative Dimension Support in Process –based Systems

We use the well-known Clover taxonomy [12] for analyzing collaborative dimension support in Process Management Systems (PMS). This model distinguishes between three forms of collaboration: *production*, which refers to objects produced or shared by multiple users, coordination, which refers to dependencies between activities of produced objects, and *communication*, which refers to protocols that support information exchange between activities of produced objects, and *communication*, which refers to protocols that support information exchange between multiple users. In our work, the produced object is a process as we considered new advanced process-based applications such as crisis and risk management applications, virtual organisations applications. In addition, we also

** http://www.bonitasoft.com/
†† http://www.camunda.com

measure when this collaborative dimension is taken into account by process-based systems, thus distinguishing collaborative design and collaborative execution of processes.

3.1. Collaborative Design of Processes

Considering BPM systems, to the best of our knowledge, the BONITA model presented in [11] is the only one supporting collaborative design, which corresponds to the analysis of the collaborative dimension at design-time. More precisely, BONITA supports collaborative production of partial schema for instance to be executed. However, BONITA does not discuss about communication between users involved in instance design and coordination between activities is specified by the instance (process) owner only. We can also mention ProCollab [13], which is a BPM system supporting collaborative communication for template selection: several process designers select in a collaborative way the template to be instantiated to deal with a specific common goal and message exchange eases their decision-making. Considering CSCW systems, CARAMBA supports collaborative production and communication [10]. Several teams involved in a virtual organization project contribute according to team skills and project needs, and each team communicates with other to make this production easier in a chained-execution setting.

On the other hand, integration of social aspects in BPM is an important research direction in order to foster flexibility, speed-up decisions and inter-participant collaboration in business process support. Thus resulting social BPM systems or contributions provide a better integration of end-users involved in process execution to the designers teams, to benefit from their business skills. As a consequence, process design is collaborative production between members of the designers teams. We mention two main contributions: [14], which recommends a wiki-based approach for process schema design, and [15], which introduces a social network to support end-users decision-making when designing processes. More precisely, in [14], end-users participate in process schema design as members of a Wiki workflow system, each contributing to process schema defining according to his business skills. Thus [14] supports a collaborative production of process schema. The Wiki also supports communication between multiple end-users involved. However, coordination between modelled activities is unclearly defined. On the other hand, [15] introduces BPMN4People, a BPMN extension allowing end-users involved in process execution to participate in process design for specific parts of the process. A social network serves as a basis for communication between these end-users who contribute voting, adding comments, and the results are exploited by the process owner to produce the final process schema. To sum up, [15] supports partial collaborative production, collaborative communication but do not support collaborative coordination as the coordination of activities is defined by the process owner only.

3.2. Collaborative Execution of Processes

Of course process-based systems support collaborative execution of processes through the notion of role, but we are more particularly interested in execution of collaborative activities which help in decision-making during process execution. Such activities correspond to decision-points influencing the way processes are executed (*e.g.*, the voting result can guide process execution to a given branch). Accordingly, we mention three main contributions. Firstly CARAMBA supports collaborative execution of activities composed of synchronized actions as different end-users are involved in action execution [10]. More precisely CARAMBA supports collaborative production and coordination. However CARAMBA does not address decision-point notion as the execution of these collaborative activities does not influence whole process execution. Secondly, [16] supports execution of collaborative activities influencing whole process execution in a process-based crisis application. More precisely these collaborative activities implement useful protocols such as vote, negotiation, delegation, and sub-contracting, helping crisis cells in decision-making. [16] addresses collaborative coordination. Thirdly ProCollab supports collaborative production and coordination: several activities, represented as task-tree coordination templates, are composed of actions executed by different end-users [13]. In addition, ProCollab supports e-mail exchange between end-users (knowledge workers) during activity execution.

4. Context and Knowledge Dimension Support in Process-based Systems

Collaborative and emergent processes of new BPM advanced applications require context and knowledge description. This section surveys corresponding contributions in the BPM area.

Context in BPM is defined in [17] as the minimum of variables containing all relevant information that impact the design and execution of a process. Context has mainly been introduced to ease process flexibility and more particularly process variability, or to enhance the adequacy of activity assignment during process execution and the identification of proper end-users for the defined roles [18]. Three main approaches have been proposed for context formal representation: the value-based approach, which models contextual information as couples (attribute,value), the model-based approach which extends the previous one considering relationships between contextual information in addition to their values (*e.g.*, [18]), and finally the ontology-based approach which adds reasoning to the modelled contextual information and corresponding relationships (*e.g.*, [19]). Only few BPM and CSCW systems support this notion of context. In CARAMBA [10], according to project needs, involved teams decide which team has to take over in the next activities of the project and the contextual information of the project make this decision-making easier to perform. In ProCollab[13], the contextual information eases the definition of actors involved in activity execution. However, in both systems, the context is not modelled formally. In [14] the context is defined as the set of social links existing between process end-users along with their reputation, and annotation of objects managed by these end-users. Finally FLOWER recommends a data-centric approach for case execution, and data values help end-users in guiding/routing case execution: more precisely these data values make the decision to skip, open or redo an activity easier for end-users.

Regarding knowledge description, we examined systems supporting Knowledge-Intensive Processes (KIPs), which correspond to processes that strongly involve knowledge socialization and exchange among multiple users. KIPs are supported by both KM process-based systems (*e.g.*, PROMOTE [20]) and BPM systems handling knowledge intensive processes (*e.g.*, ProCollab [13], [21]). In addition, CARAMBA [10] and Cognoscenti [22] support knowledge exchange between users. More precisely, in PROMOTE, users describe knowledge shared and related to activities as instances of a specific model. More precisely users can annotate activities, supply or consult discussion forums, frequently asked questions, databases, or blackboard. CARAMBA [10] recommends a hybrid approach supporting e-mail exchange on top of a specific shared space in which useful information is given to the project teams. Common to [13] and [21] is that they share knowledge via e-mail exchange. In addition, [21] recommends a text-mining approach extracting the relevant knowledge from these e-mails, thus making implicit knowledge explicit. Finally, Cognoscenti [22] recommends a shared space as an xml document supporting knowledge exchange between process users. However, except PROMOTE, these contributions do not explicitly and formally model knowledge.

5. Comparative Analysis and Discussion

This section reports a comparative analysis of examined systems with respect to the previous requirements. It also highlights the benefits and drawbacks of examined systems in meeting these requirements.

5.1. Comparative Analysis

We analyzed the main process-based systems in BPM, ACM, CSCW and KM areas with respect to the required dimensions for supporting advanced process-based applications. These dimensions correspond to the need for process flexibility as well as a strong intertwining of process design and process execution, the need for supporting collaborative design and execution of the process driving these applications, and the need for context and knowledge-support to both describe the current situation and the already existing knowledge needed to deal with such a situation.

In the three previous sections, we analyzed in depth these dimensions, identifying specific criteria for each of them. Firstly we examined the required flexible dimension considering ability of systems in supporting both process flexibility according the taxonomy presented in [2] that distinguishes between flexibility at design and flexibility at run-time along with four flexibility needs (variability, adaptation, evolution and looseness). We also have examined the required flexible dimension considering ability of systems in supporting process design and execution intertwining. This second criterion is very important for Emergent, Collaborative and Flexible (ECF) processes. For example, if we consider crisis-based applications, the crisis resolution process defining actions to be executed on the ground and their coordination is subject to frequent changes to meet the evolving requirements of the crisis. These changes can be performed while there are ongoing process actions on the ground. This means that design and execution of such processes are strongly intertwined: actors are allowed to design the actions and their coordination of the considered process while executing it.

Secondly, we examined the required collaborative dimension in the light of the Clover taxonomy [12], which distinguishes between three forms of collaboration: production, coordination and communication. In addition we took into account when collaboration is supported: design-time or run-time. This second criterion is related to the needed intertwining of process design and execution. This collaborative dimension is of paramount importance for Emergent, Collaborative and Flexible processes such as crisis management processes. In such a context, several participating actors (experts) have to act simultaneously and urgently to reduce the crisis and its impacts on the real world. To achieve this common goal efficiently, these actors must collaborate, or at least act in a coordinated way in order to make their activities as efficient as possible. Designing this coordination requires the use of business skills and experiences of the involved groups of actors, who are responsible for defining actions to be undertaken on the ground and their coordination according to their areas of expertise, and the process driving crisis resolution merges the defined actions from all relevant actors. Thus a crisis resolution process model/schema is the result of a collective and consensual modelling work between actors involved in crisis resolution.

Thirdly, when examining the required context and knowledge dimensions to support collaborative and emergent processes, we highlighted on the one hand why and how the context is modelled and used, and on the other hand how the knowledge is modeled, produced, shared and reused. Context and knowledge dimensions are very important when considering ECF processes. For instance, in a crisis-based application, actors involved in crisis resolution must have a clear representation of the crisis (its context); they also must have access to explicit knowledge existing about past and similar crises in order to reuse it possibly. Finally, at the end of the crisis, the adopted solution, represented as a crisis resolution process, can be eventually (and partially) integrated as an explicit knowledge, for the purpose of easing future and similar crisis resolution.

Fig. 1 and Fig. 2 give an overview of the evaluation results of the examined systems. More precisely, these figures indicate whether a system provides full support (+), partial support (+/-), weak support (-) or no support for each specific criterion of the considered dimension. We have drawn these evaluation results from analysis provided in the three previous sections.

dimensions and criteria / systems	FLEXIBLE DIMENSION							COLLABORATIVE DIMENSION					
	design-time	run-time	intertwining	variability	adaptation	evolution	looseness	production design-time	production run-time	coordination design-time	coordination run-time	communication design-time	communication run-time
ADEPT [3]	+	+			+	+							
YAWL [5]	+	+		+	+		+						
DECLARE [8]		+			+	+	+						
FLOWER [9]		+			+								
CARAMBA [10]			-		+/-			+	+		+	+	
BONITA [11]			-			+/-		+					
ProCollab [13]		+				+/-			+		+	+	
[14]								+				+	
BPMN4People [15]								+/-				+	
[16]	+	+			+/-		+/-				+		
PROMOTE [20]													
[21]													
Cognocenti [22]								+					
CAMUNDA			-		+/-								

Fig. 1. Evaluating Systems against Flexible and Collaborative Dimensions.

dimensions and criteria / systems	CONTEXT DIMENSION					KNOWLEDGE DIMENSION		
	role	variability	value-based	model-based	ontology-based	e-mail exchange	shared space	model
ADEPT [3]								
YAWL [5]								
DECLARE [8]								
FLOWER [9]		+	+					
CARAMBA [10]	+		+			+	+	
BONITA [11]								
ProCollab [13]	+		+			+		
[14]	+		+					
BPMN4People [15]								
[16]								
PROMOTE [20]								+
[21]						+		
Cognocenti [22]							+	
CAMUNDA								

Fig. 2. Evaluating Systems against Context and Knowledge Dimensions.

5.2. Discussion

Fig. 3a positions these examined process-based systems with respect to the considered areas, namely BPM, ACM, CSCW and KM areas. Of course, the intersection of these areas is not empty, and examined process-based systems belong to both of them. According to this figure, ADEPT [3] is only relevant to the BPM area as it is a process-based system considering both process instance and process model. ADEPT provides a full-support for process flexibility but does not provide any support for the collaborative dimension and the context and knowledge dimensions. The same conclusion can be drawn for YAWL [5], DECLARE [6], and CAMUNDA. FLOWER [9] is an ACM system. It only considers process instances and provides support for process flexibility and more particularly process adaptation. In addition, it provides support for context modelling as it advocates a data-centric approach in case execution. Cognoscenti [22] is an ACM system but it also provides a weak support for knowledge modelling, producing and sharing. Thus [22] is at the border of ACM and KM. Both CARAMBA [10] and BONITA [11] are relevant to BPM and CSCW. More precisely [10] is a CSCW system providing support for process modelling and execution while [11] is a BPM system providing support for collaborative activities. In both systems, the flexible dimension is weakly addressed. [14] and [15] are BPM systems, and more precisely social BPM systems. As BPM system, they provide support for process modelling and execution, and their social dimension make them collaborative. Thus they are relevant to both BPM and CSCW. [16] is a BPM system providing support for collaborative activities. Consequently, it provides a weak support of the required collaborative dimension and it is at the border of CSCW and BPM. Finally, [13], [20] and [21] are relevant to both BPM and KM.

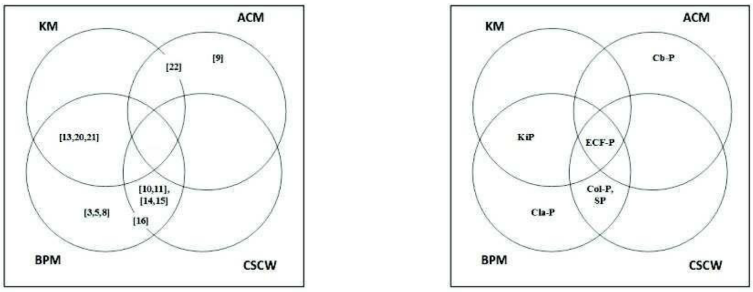

Fig. 3. Positioning (a) examined process-based systems and (b) class of processes, with respect to system areas.

Classes of processes that examined process-based systems support are ranging from Classical Processes (Cla-P) to Case-based Processes (Cb-P), Collaborative Processes (Col-P), Social Processes (SP), and Knowledge-intensive Processes (KiP). Fig. 3b positions these classes of processes with respect to the considered areas. In this paper, we have introduced a new class of processes, named Emergent, Collaborative and Flexible (ECF) processes, which are processes of advanced process-based applications such as crisis and risk management, virtual organizations, or factory of the future. These processes have specific features. Firstly, they are *emergent* as they require the intertwining of their design and execution. For instance, in a crisis application, crisis resolution processes are gradually defined during their execution according to the evolving requirements of the crisis. More precisely, the actors involved in crisis resolution process use their skills, expertise and experience to add and synchronize existing or new created set of activities according crisis evolution. Secondly, these processes are *collaborative* as they require an efficient collaboration of the groups of actors involved in their design and execution. More precisely, addition and synchronization of existing or new created set of activities is the result of a consensus between the involved groups of actors. Moreover, these latter benefit from their skills, expertise and experience to ensure ECF process design and execution and it is of utmost importance to provide them with means to model, produce, share and reuse their *knowledge*. Thirdly, these processes operate in a dynamic environment and they are subject to numerous unforeseen changes: thus they require *flexibility capabilities* but also an explicit representation of the *context* in which these changes occur. Indeed contextual information is of utmost importance for group of actors involved in ECF process design and execution as it eases their decision-making.

Moreover, these processes have two additional features. They are *case-based* and this is due to their emergent nature. Thus we only consider cases, *i.e.* instances of processes, and the case schema is defined by the involved groups of actors while executing it. Finally, they are *loosely-coupled* [23] as each involved groups of actors take care of a specified part of the process design and execution while interweaving their respective contribution. Consequently, as indicated in Fig. 3b, ECF processes fall within BPM, ACM, CSCW and KM areas.

Superposition of Fig. 3a and Fig. 3b highlights that none of the examined systems support ECF processes. The main lack of these systems is their inability to intertwine collaborative design and execution of processes. More precisely, as illustrated in Fig. 1, [10] supports adding already modelled activities at run-time (*i.e.*, added activities are not collaboratively modelled at run-time) while [11] supports the individual creation and addition of activities at run-time (*i.e.*, only the process owner is involved in this creation and addition). This also holds for CAMUNDA. So [10], [11] and CAMUNDA weakly support the intertwining of process design and execution. On the other hand, these systems do not support loosely-coupled coordination of groups of actors involved in ECF process design and execution. However, some of the examined systems partially fulfil ECF process requirements. From the flexible dimension point of view, ADEPT [3] is an interesting contribution with respect its process adaptation mechanism (change patterns [4]), and FLOWER [9] provides interesting deviation primitives (*e.g.*, skip, redo) to improve ADEPT process adaptation mechanism. From the collaborative dimension point of view, CARAMBA [10] and ProCollab [13] could be extended to achieve collaborative production, coordination and communication. Finally, from the context and knowledge dimensions point of view, PROMOTE [20] could be extended to integrate the context dimension as defined in [19] in addition to the knowledge dimension.

6. Conclusion

This paper has introduced a new class of processes, namely Emergent, Collaborative and Flexible (ECF) processes, encountered in advanced process-based applications such as crisis and risk management, or virtual organizations. The contributions of the paper are the following: identification of a set of requirements to support ECF processes, and comparative analysis of the main process-based systems falling within the BPM, ACM, CSCW and KM areas and discussion of their ability in supporting ECF processes. The conclusion is that none of the examined systems support ECF processes. Future works will address the process-based system supporting ECF process design and implementation issue, dealing with the following questions, in a crisis-based application setting:

- Which architecture must be specified for such a system?
- Which collaborative environment must be defined in such a system? Which kind of interaction must be allowed in this collaborative environment?
- How to support process design and execution intertwining?
- How to support deviation primitives as in ACM systems and change patterns as in BPM systems to face flexibility needs?
- How such a system must take into account contextual information and knowledge useful for decision-making by the involved groups of actors?

Acknowledgment

This work is supported by the Genepi project, funded by the French research national agency (ANR).

References

1.Tahir O., Andonoff E., Hanachi C., Sibertin-Blanc C., Bénaben F., Chapurlat V., Lambolais T.: A Collaborative Information System Architecture for Process-Based Crisis Management. Int. Conference on Knowledge-Based Intelligent Information and Engineering System, Zagreb, Croatia, September 2008, pp. 630–641.

2. Reichert M., Weber B.: Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Springer, 2012.

3. Dadam P., Reichert M.: The ADEPT Project: a Decade of Research and Development for Robust and Flexible Process Support. Computer Science - Research and Development, vol. 23, n°2, 2009, pp. 81–97.

4. Weber B., Reichert M., Rinderle S.: Change Patterns and Change Support Features – Enhancing Flexibility in Process-Aware Information Systems. Data and Knowledge Engineering, vol. 66, n°3, 2008, pp 438–466.

5. ter Hofstede A., van der Aalst W., Adams M., Russel N.: Modern Business Process Automation – YAWL and its Support Environment. Springer, 2010.

6. Adams M., ter Hofstede A., Edmond D., van der Aalst W.: Worklets: a Service-Oriented Implementation of Dynamic Flexibility in Workflows. Int. Conference on Cooperative Information Systems, Montpellier, France, October 2006, pp. 291–308.

7. Adams M., ter Hofstede A., Edmond D., van der Aalst W.: Dynamic and Extensible Exception Handling for Worklows: A Service-Oriented Implementation. Int. Conference on Cooperative Information Systems, Vilamoura, Portugal, November 2007, pp. 95–112.

8. van der Aalst W., Pesic M., Schonenberg H.: Declarative Workflows: Balancing between Flexibility and Support. Computer Science – Research and Development, vol. 23, 2009, pp. 99–113.

9. van der Aalst W., Weske M., Grünbauer D.: Case Handling: a New Paradigm for Business Process Support. Data and Knowledge Engineering, vol. 53, n°2, 2005, pp. 129 van der Aalst W., 162.

10. Dustdar S.: Caramba – A Process-aware Collaboration System supporting Ad hoc and Collaborative Process in Virtual Teams. Distributed and Parallel Databases, vol. 15, 2004, pp. 45–66.

11. Charoy F., Guabtni A., Faura M.: A Dynamic Workflow Management System for Coordination of Cooperative Activities. Business Process Management Workshops, Vienna, Austria, September 2006, pp. 205–216.

12. Ellis C. Wainer J.: A Conceptual Model of Groupware. Int. Conference on Computer-Supported Cooperative Work, Chapel Hill, USA, October 1994, pp. 79–88.

13. Mundbrod N., Beuter F., Reichert M.: Supporting Knowledge-intensive Processes through Integrated Task Lifecycle Support. Int. Enterprise Distributed Object Computing Conference Adelaide, September 2015, pp. 19–28.

14. Erol S., et al.: Combining BPM and Social Software: Contradiction or Chance? Int. Journal on Software Maintenance and Evolution: Research and Practice, vol. 22, 2010, pp. 449–476.

15. Brambilla M., Fraternali P., Vaca, C.: A Model-driven Approach to Social BPM Applications: Chapter in Social BPM Handbook, BPM and Workflow Handbook series, Future Strategies, 2011, pp. 95–112.

16. Andonoff E., Hanachi C., Le Nguyen Tuan T., Sibertin-Blanc C.: Interaction Protocols for Human-Driven Crisis Resolution Processes. IFIP Working Conference on Virtual Enterprises, Albi, France, October 2015, pp. 63–76.

17. Rosemann M., Jan R., Christian F.: Contextualization of Business Processes. Int. Journal Business Process Integration and Management, vol. 3, n°1, 2008, pp. 47–60.

18. Saidani O., Nurcan S.: Context-Awareness for Adequate Business Process Modelling. Int. Conference on Research Challenges in Information Science, Fes, Morocco, April 2009, pp. 177–186.

19. Saidani O., Rolland C., Nurcan S.: Towards a Generic Context Model for BPM. Int. Conference on System Sciences, Kauai, Hawai, USA, January 2015, pp. 4120–4129.

20. Woitsch R., Karagiannis D.: Process-Oriented Knowledge Management Systems based on KM-services: the PROMOTE Approach. Int. Journal on System in Accounting, Finance and Management, vol. 11, n°4, 2002, pp. 253–267.

21. Soares D., Santoro F. Baião F.: Discovering Collaborative Knowledge-intensive Processes through e-mail Mining. Journal of Network and Computer Applications, vol. 36, n°6, 2013, pp.1451-1465.

22. Swenson K.: Cognoscenti Open Source Software for Experimentation on Adaptive Case Management Approaches. Int. Conference on Enterprise Distributed Object Computing, Demonstrations Session, Ulm, Germany, September 2014, pp. 402–405.

23. van der Aalst W.: Loosely Coupled Inter organizational Workflows – Modelling and Analyzing Workflows crossing Organizational Boundaries. Information and Management, vol. 37, n°2, 2000, pp. 67–75.