Department of Computer Science
San Marcos, TX 78666

Report Number TXSTATE-CS-TR-2008-15

Metric-based dynamic process adaptation for crisis mitigation

Rodion Podorozhny
Dimitrios Georgakopoulos
Anne H.H. Ngu

2008-05-16

Metric-based dynamic process adaptation for crisis mitigation

Rodion Podorozhny[1], Dimitrios Georgakopoulos[2], Anne H.H. Ngu[1]

[1] Computer Science Dept., Texas State University,
San Marcos, TX 78666, USA
{rp31, hn12}@txstate.edu
[2] Telcordia Technologies, 106 E. Sixth Street, Suite 415,
Austin, TX 78701, USA
dimitris@research.telcordia.com

Abstract. Business processes emphasize the capture of best practices and improvement of business activities. A dynamic process aims to deal with a dynamic environment where processes have to change and adapt to respond to unexpected events and situations that have not been anticipated. Dynamic process change currently focuses on mechanisms and engines for dynamic process specification and execution. However, existing approaches do not ensure that dynamically adapted processes achieve desirable performance and optimality goals that are set by the target application. In this paper we suggest an approach that assembles a process customized for a particular crisis and dynamically modifies it, if necessary, to maintain an optimal execution.

Keywords: business process, process adaptation, crisis mitigation

1 Introduction

The challenges posed by the many real world problems call for solutions that involve dynamic process adaptation. For example, processes for dealing with emergency mitigation must be dynamically adapted to coordinate the activities of multiple organizations and deal with unanticipated incidents, and do this efficiently. However, existing approaches do not ensure that dynamically adapted processes achieve desirable performance and optimality goals that are set by the target application.

In this paper we propose a solution for dynamic process adaptation that includes methods for metric-based process adaptations at run-time. The proposed solution involves process evaluation via combined metrics. Whenever such metrics violate a threshold, the process is adapted to achieve a better combined metric. The paper is organized as follows: In Section 1, we introduce a motivating application. Section 2 describes the dynamic adaptation approach itself, while Section 3 provides a description of the infrastructure process management system that supports such dynamic adaptation. Discussion of related work is in Section 5. Finally, in Section 6 we draw some conclusions.

2 Motivating application

As a motivating application we take a crisis initially arising due to a fire. As more information is gathered about the severity of a fire or its possible causes, a crisis can be escalated to an emergency and additional resources may be committed to deal with it. Thus a created and dynamically adapted process should be able to coordinate the activities of various fire departments, police departments and other kinds of organizations that might be involved in resolving a crisis of this kind.

The environmental events of an example scenario unfold as follows: a fire is detected on a particular floor and a room (room1) in a building that is located close to a border of responsibilities of two adjacent fire departments (time t1), at a later time (time t2; t2>t1 by about 15 min) a second fire is detected in the same building and floor, but in a different room (room2). The second fire is a result of what appears to be an explosion. The room in which a second fire is created is not adjacent to the room with the fire detected earlier. A third fire closely follows a second one. The third fire is also due to what appears as an explosion. It is in a room (room3) adjacent to the room with the second fire. This fire is detected at time t3 (t3 > t2 by about 1 minute).

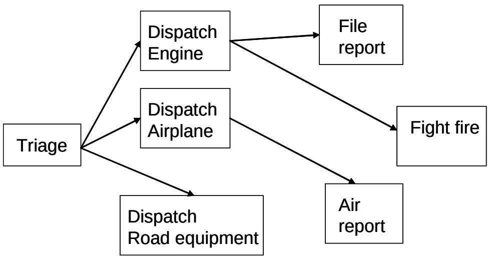

Figure 1. Fire department process for fire fighting (FFP)

We assume that there are two fire departments (FD1 and FD2) and a single police department (PD1) that might be potentially involved in resolving this crisis (due to the location of the building). Initially only one fire department is alerted by the first fire. Another assumption is that there is no combined command and control (C2) team at time t1. This team is created in response to a third fire after time t3. The sequence of second and third fire detections in close succession is regarded as an emergency. We also assume that there is an organization that receives the fire reports (fire report monitor, FRM). One of the duties of this organization is to detect conditions for escalating a crisis to an emergency.

Each of these four organizations has a set of predefined processes for dealing with various crises and emergencies. Some of these predefined processes can be relevant for dealing with the escalating crisis. We assume that there is no predefined process and command control organization that will coordinate the escalating crisis that started with the first fire.

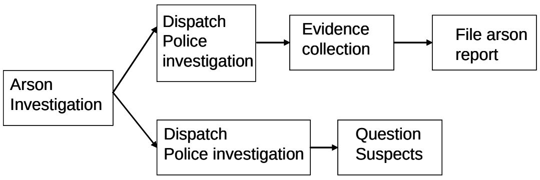

Figure 2. Police department process for arson investigation (AIP)

Next we will briefly describe the predefined processes of these organizations that are relevant to the escalating crisis. For description of the processes we will use a simplified version of a process language introduced by Telcordia’s Awareness-Enabled Coordination (AEC) platform [GNBC06]. In this language a visual representation of a process uses rectangles to denote process activities and arcs to denote dataflow dependencies. Thus, a process definition with three steps (S1, S2, S3) and two directed arcs {(S1,S2),(S1,S3)} prescribes that steps S2 and S3 may execute in parallel after completion of step S1. So that to define a conditional execution, mutually exclusive conditions must be associated with the arcs originating from the same step.

The predefined processes for the organizations in the example scenario are described in the Figures 1-3.

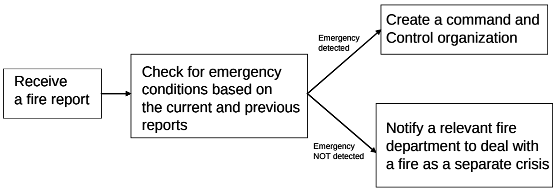

Figure 3. Fire report monitor process for escalating a crisis to an emergency (FMP)

A predefined process for a fire department (FD) to deal with a fire is described in Figure 1 (Fire Fighting Process or FFP).

A predefined process for a police department to investigate an arson is described in Figure 2 (Arson Investigation Process or AIP).

A predefined process for a fire report monitor to decide if to escalate a crisis to an emergency is described in Figure 3 (Fire Monitoring Process or FMP).

3 Dynamic process adaptation approach

In this section we describe a semi-automated solution that deals with the problem of dynamic process adaptation at runtime while maintaining the optimality of the dynamically adapted process. This is achieved by guiding the process adaptation by a set of performance metrics that measure the degree of process success in meeting its operational goals. The proposed solution modifies the process dynamically (at process run-time) whenever process performance metrics indicate that the (measurable) process objectives are not being met.

It should be noted, that the process resulting from such dynamic adaptation is typically non-reusable, since it is highly optimized for a specific incident and no two incidents are identical (if they were identical there will be no need for emergency response).

The main aspects of the proposed solution include methods for:

- **Escalation** of a process (e.g. adding resources if crisis information updates indicate that the crises is of greater magnitude and/or severity).
- Creation of an initial process for a **normal operation.**
- **Modification** of the process to comply with the metrics and to deal with the **lack of resources** required by activities that are dynamically introduced or refined.

In the Table bellow, we outline the effect of the proposed solution to the application scenario described in Section 2.

Time	Activities
t1	Before the first fire report arrives, an instance of process FMP is executed by the FRM organization. The Fire Departments FD1 and FD2, and the Police Department PD1 do not execute any processes associated with a fire.
t2	The first fire report at time t1 forces process FMP to notify fire department FD1 to start an instance of process FFP.
t3	FD1 initiates a process FFP to deal with fire 1.
t4	The second fire report at time t2 forces process FMP to notify fire department FD2 to start an instance of process FFP.
t5	FD2 initiates a process FFP to deal with fire 2.
t6	The third fire report at time t3 forces process FMP to escalate a crisis to an emergency. This entails that a Command and Control (C2) organization is created to deal with emergency 1.
t7	C2 organization creates an initial process that combines and coordinates the activities of organizations FD1, FD2 and PD1.
t8	The coordination of activities of FD1, FD2, and PD1 are achieved by combining (partially executed) processes FFP by FD1, FFP by FD2 and AIP by PD1.

The combination at time t8 is accomplished by C2 performing the following steps:

1. Instances of processes FFP by FD1, FFP by FD2, and AIP are modified to include communication activities (to be described in greater detail below).
2. The instances of these processes are combined in one process to deal with emergency 1 (Combined Emergency Process or CEP).
3. Combined process CEP is modified to include continuous evaluation activities (to be described in greater detail below). These activities are performed by the C2 organization. The result of these activities can modify the CEP process to optimize it according to the problem domain metrics.
4. Combined process CEP is evaluated by a method, e.g. as in [Zeng04]. The method takes a process definition and a set of criteria as input and produces an optimal path through the process which is represented as a directed acyclic graph to allow for possibility of parallel execution threads by more than one organization. Thus it is possible to get a rough metric estimate of the static definition of the CEP process so that to enable humans in the organization C2 evaluate several alternative definitions of the combined process CEP.
5. Process CEP starts executing. Various activities of CEP are being executed by organizations C2, FD1, FD2, and PD1. The activities that were finished while processes FFP and AIP were separate are not repeated.
6. As process CEP is executed, the continuous evaluation activities are performed by the C2 organization, both periodically and according to the activities woven into the CEP process. If the metrics show that the emergency situation is not resolved quickly enough, the C2 changes the CEP process dynamically. Thus a metric-based adaptation is achieved.
7. In addition, the C2 organization can change the CEP process to assign certain activities to organizations that were not initially intended for them.

For instance, a police department can receive activities directly related to fire fighting even though the initial AIP process by PD1 does not contain any fire fighting activities. We call this modification of the CEP process “adaptation for customized mitigation”.

3.1 Adaptation for communication

Processes for coordinating activities of involved organizations are modified by way of insertion of communication activities so that the organizations resolving the same crisis could communicate and coordinate.

Some reasonable places to insert the communication points are:

- After the first (initiation) activity of a process
- Before the activity of a process that greatly contributes to the process result (for instance the Fight Fire activity in the FFP process)
- In between the activities that greatly contribute to the process result so that to coordinate and adjust the main activities. The frequency of communications strikes a trade-off between the overhead due to the cost and duration of communication and adaptability of the process. In the example of process FFP the Fight Fire activity is broken up into several portions and communication activities are inserted between them (Figure 5). Similarly, in the AIP process the Evidence Collection activity is treated accordingly. There are some techniques for determining communication policies (i.e. whether to communicate at certain points) and evaluating the optimality of such policies. For example, a technique for deriving a practical communication policy that takes into account communication costs and uncertainties of the real world environment is described in the work by Milind Tambe [Tam97]. A method for evaluation of optimality of a communication policy is suggested in the work by Milind Tambe and David Pynadath [PynT02]. Informally, the mentioned communication policy suggests that an organization should communicate to other organizations involved in the combined process that a certain (possibly intermediate) goal G has been achieved if the cost of miscoordination outweighs the cost of communication that the said goal G has been achieved. The cost of miscoordination is the difference between an estimated metric of a combined process without the said communication and that process enhanced with the said communication. The communication adaptation mechanism can use such a technique for deciding if and when the involved organizations should communicate about partial results of the

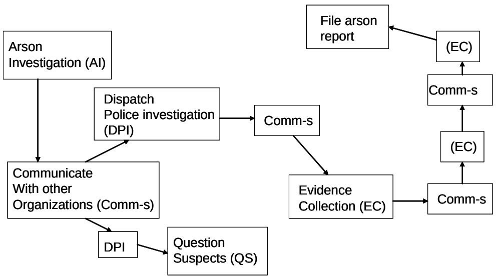

Figure 4. AIP after communication adaptation

combined process. As a matter of fact this communication policy can be used to decide whether to enhance the combined process with communication activities *dynamically*.

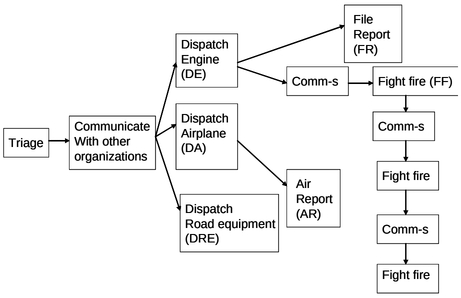

Figure 5. FFP after communication adaptation

We show an example of communication adaptation in Figures 4-6.

In Figure 4 the FFP process is modified by insertion of Communication activities after the initial activity and before and in-between the broken up major activity – Fire Fight. In Figure 5 the AIP process is modified in a similar fashion. Finally, Figure 6 combines two instances of the FFP process (executed by FD1 and FD2 organizations) and a single instance of the AIP process. The communication activities serve as synchronization points.

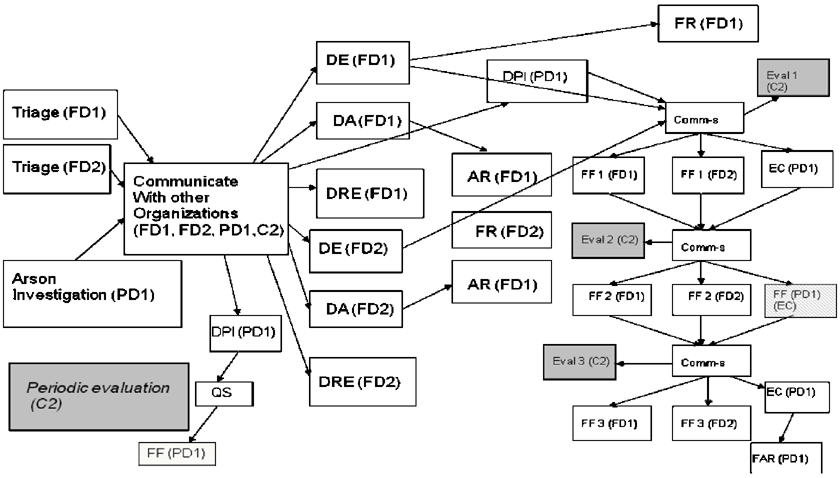

Figure 6. Combined Emergency Process (CEP) after evaluation-driven adaptation

3.2 Adaptation and metrics for continuous evaluation

So that to achieve a metric based dynamic adaptation of the process the evaluation activities performed by the Command and Control organization (C2) have to be introduced in the Combined Emergency Process (CEP). The result of this adaptation is illustrated in Figure 6. The rectangles denoting the evaluation activities have a dark gray background. One of the evaluation activities is performed periodically (Periodic evaluation). It is shown as a rectangle without any incoming or outgoing arcs. According to the semantics of the AEC process language this indicates that this activity runs in parallel to the rest of the activities on that diagram because it does not have any dependencies on them. In addition to the periodic evaluation activity, the evaluation adaptation introduces activities in-between the major process activities. In this case they are synchronized with the communications activities.

An evaluation activity is performed by the problem domain experts comprising the C2 organization. The many use tools for fire impact analysis that take into account current temperatures, winds, humidity and the extent of the fire. However, the tools are not integrated with the CEP process. This kind of activity calculates a combined metric that shows the "goodness" of the current situation. This combined metric depends on a (possibly) hierarchy of lower level metrics specific to the problem domain.

If the value of a combined metric is not satisfactory then the C2 can modify the process dynamically in hopes that the modified process will achieve a better value of the combined metric. An example of such a modification is mentioned in section 3.3.

While the particulars of low level and combined metrics are not the focus of this paper we suggest a possible set of metrics for a fire fighting problem domain.

Estimated time for the assets to assume certain locations necessary for fire fighting: estTimeToLoc = max (1=<i<=n, pathLength(currentLocation(asset$_i$), assignedLocation(asset$_i$))/speed(asset$_i$)), where n is the number of assets moving to new locations, function *path* determines the length of a path a certain asset must travel to move from its current location to its new location (assuming constraints such as walls). Informally, this function chooses the maximum time it would take all involved assets to assume assigned locations from their current locations.

Speed of extinguishing a fire (a function that takes into account combined fire fighting speeds of agents involved, the speed of fire propagation). Assuming that speeds of fire fighting and fire propagation are measured in (area unit)/(time unit), this function can be expressed as: speedToExtinguish(fire$_j$) = sum(1=<i<=n, fireFightSpeed(asset^j_i)) – firePropSpeed(fire$_j$). Informally, it is the sum of fire fighting speeds of all assets assigned to fight a certain fire with index j minus the speed of propogation of the said fire with index j. Index j corresponds to a certain number assigned to a fire (e.g. according to the time of fire's detection: 1<=j<=numberOfFires).

Estimated time to extinguish a fire (fire area divided by the speed of extinguishing the said fire): estTimeToExtinguish(fire$_j$) = area(fire$_j$)/ speedToExtinguish(fire$_j$).

3.3 Adaptation for customized mitigation

If an evaluation activity delivered a value of a combined metric that is below a certain threshold (e.g. the estimated time to extinguish is later than expected) then the C2 organization can dynamically modify the combined process. The AEC process enactment environment enables such a dynamic modification [GNBC06]. In Figure 6 a possible customized mitigation in response to a negative evaluation is shown.

Let us assume that the combined fire fighting speed of undamaged assets belonging to FD1 is 0.5 m^2/min, that of assets belonging to FD2 is 0.7 m^2/min. The speed of propagation of fire 2 to which the assets of FD2 were assigned is 0.2 m^2/min. Thus the speed of extinguishing fire 2 is 0.7 – 0.2 = 0.5 m^2/min. Further, let us make the area of fire 2 to be 15 m^2. This means that the estimated time to extinguish fire 2 with undamaged assets of FD2 is 30 min which is under an acceptable time limit of not greater than 30 min. During the execution of activity *FF 1 (FD2)* (Figure 7) the combined fire fighting speed of FD2 assets drops to 0.4 m^2/min due to a certain contingency. Let us say that this drop occurred 3 min after start of *FF1 (FD2)* and that an evaluation activity finished in another 2 min (the details of execution of communication activities and the associated simple arithmetic calculations are omitted). Thus, 0.7*3 + 0.4*2 = 2.9 m^2 of fire 2 have been extinguished. An evaluation shows that the remaining time for extinguishing fire 2 is (15-2.9)/0.4 = 30.25 with the total time being 30.25 + 5 = 35.25 min. This estimate is above the acceptable time limit so the C2 organization modifies the process by assigning fire fighting activities to the police department PD1. This example is kept simple for illustration purposes – a more complicated model with integration over time and taking into account uncertainty can be suggested.

Thus the C2 organization charged the police department (PD1) with the Fire Fight activities. The modified and added activities have a light gray background in Figure 6. One Fire Fight activity was added after the Question Suspects (QS) step. Also, one of the Evidence Collection (EC) activities has been replaced with a Fire Fight activity. Addition of the fire fighting activities is likely to reduce the estimated time to extinguish because the speed of extinguishing a fire is increased. At the same time the speed of evidence collection is reduced but it is deemed a reasonable trade-off by the C2 organization.

4 Infrastructure for metric-based dynamic process adaptation

Awareness-Enabled Coordination (AEC) is a platform designed to address the problem of coordination of large multi-organizational teams. AEC provides a contextualization mechanism that helps its users deal with complex, real world environments where teams involve humans, tools, and services that come from different organizations, are subject to multiple jurisdictions, and provide diverse expertise. To provide efficiency in achieving team objectives, AEC provides process-based coordination and automation, ongoing policy enforcement, as well as situation and project-related awareness. To allow individuals, teams, and organizations to deal with dynamically changing situations, AEC permits dynamic adaptation of user activities, process, resources, organizations, and teams at any time.

Coordination and automation of team activities enhances efficiency. AEC provides a *flexible processes model* and corresponding context-based process management mechanisms to enhance the efficiency of coordinating and information sharing among members of multi-organizational teams in a dynamic setting. When processes or parts of processes are well-structured and well-defined, AEC provides the option to automate them, reducing the work load on the team members. In the following paragraphs we describe AEC’s flexible process model, and describe in more detail its novel capabilities for process-based coordination and automation.

AEC's flexible process model permits interleaved definition, refinement, and execution of activities and processes. A process activity in AEC is a collection of child activities, possibly constrained by dependencies on their execution. A child activity may be intended to be done by humans, be a program or service that is accessed directly (either with the help of a human or in an automated fashion), or in turn be a nested processes. Semantic activity types, called *activity intents*, are defined in AEC's *activity ontology*, whose purpose is to provide a common semantic type system that allows AEC users to indicate their intent when they start a new activity (e.g., by selecting the appropriate semantic activity type in the ontology). Activity intents do not define how an activity is to be done. Thus, the activity's *method* must be defined before activity becomes concrete (i.e., executable by AEC).

Child activities of an AEC process may be constrained in terms of when they can be executed. Like in many traditional process models [WfMC,BPEL,FIL06,COS03], AEC's process model supports control flow dependencies that order the execution of activities, forcing one activity to precede the other. Resource selection dependencies define the resource types required by each activity in AEC. Resource flow dependencies in AEC are constraints in the flow of resources to or from the context of each activity. If a process has control flow dependencies between all its child activities, we call it a (fully) structured process. Partially structured processes only have control flow dependencies between some child activities, while the child activities of unstructured processes have no control dependencies (i.e., there basically set of activities and of these activity can be performed at any time). We use the term *predefined* to refer to activities or processes that have all their control flow, resource selection, and resource flow dependencies defined before there are executed. We refer to activities and processes being defined (e.g., by adding a method, changing control and resource flow or adding a resource selection) after the execution of their parent process starts as *dynamically refined*.

In addition to dynamically refining a process activity from just the intent, AEC supplies additional functionality for selecting a method to fill the intent of an activity. One approach is for the user to access a *method catalogue*. Each context has an indexed catalogue of suggested (or required) specifications for methods related to specific intents. The user can access the method catalogue from the activity's context based on the intent of the activity, and choose the appropriate entry in the catalogue to use as a basis for his own activity. The user may use the selected method as a starting point for refinement, or may be required to follow a method strictly (for instance, if it is a traditional business process).

AEC's *flexible process model* supports a wide variety of process-based coordination styles ranging from fully *structured* to *unstructured* processes. Furthermore, the AEC flexible process model permits *dynamic refinement and change* of any process during its execution. For example, just as in many other process management systems (e.g., workflow systems and EAI integration platforms [COS03,BEA06,FIL06]) AEC supports the specification and automation of *business processes* for organizations, jurisdictions, and teams. Using the terms we defined above, business processes are predefined and structured AEC processes that apply to organizational and jurisdictional AEC contexts. Specified business processes can be analyzed and measured to assess and improve their efficiency. Process automation can drastically reduce overhead and cost for assigning tasks to people, coordinating activities, tracking progress towards achieving process goals, and maintaining accountability information. AEC provides all these benefits of business process management; however, since we follow well-understood methodologies, we will not discuss these capabilities further in this paper.

The main novel aspect of AEC's flexible process model is that it supports (partially) unstructured processes and/or dynamically-refinable processes. This lack of a requirement for fully-structured process definition permits AEC to accommodate individual and team work styles ranging from highly-structured business processes to dynamically self-organized work. This capability gives AEC a distinct efficiency advantage over the ad hoc coordination advocated by many groupware tools such as [Groove].

Dynamic refinement and change during process execution permits AEC process execution to start even if a process is only partially defined. The process may be further refined as progress is made towards accomplishing its intent. For example, refinement may occur when decisions concerning the method are made during execution, when a resource is assigned to a child activity immediately before it is executed, or when external events require abandoning planned activities and initiating new unplanned activities in response. For processes, dynamic process refinement modifies the specification of the child activities and the control flow dependencies between them to make the activity more concrete. To accommodate such refinement, an AEC process can include activities that are only specified at the level of what the activity intends to accomplish. (as we discussed earlier activity intents are defined according to a domain-specific activity ontology in AEC). As AEC users obtain more information concerning the details of what needs to be done, the activity may be refined until it is fully-specified.

AEC automates processes to provide further coordination efficiency. Flexible process execution in AEC is performed by process engine functionality that is distributed in all contexts maintained by AEC. When a flexible process starts in a context C, the AEC process engine in C enables the execution of each of its child activities once the following conditions are met on that activity:

- The child activity is defined well enough that execution can begin. Otherwise, the user is asked to refine the activity until it reaches the point where the execution can begin. This may involve defining a process or selecting a preexisting method (e.g., a one that is available in an organizational context).
- The child activity has access to all of the resources on which it depends. Otherwise, the user whose role is specified in the activity is asked to select and bind resources in its context to resources in its environment. The selection of available resources is determined via dynamic contextualization.
- All of the activities' control flow dependencies are satisfied.
- Starting the activity would not violate any coordination policies.

Once a child activity is ready for execution, it can either run automatically (if it is flagged as *automatic*), or be started by the intervention of some responsible team member. The process may be monitored by any team member, but a specific responsible party (this is a specified activity *role*) is given the task of dealing with any issues during process execution. AEC provides tools for defining, refining, and monitoring flexible processes.

Related work

With respect to process-based coordination and automation, many existing workflow systems (e.g., COSA [COS03], FileNet [FIL06]), Enterprise Integration platforms (e.g., WebLogic Integrator [BEA06], and NetWeaver [SAP06]) as well as standards for process workflow management [WfMC] and process-based web service integration [BPEL] are all geared towards modeling and automating processes that are predefined and fully structured. Therefore, these technologies and standards lack the flexibility of AEC's flexible process model that is necessary to support teamwork in changing environment. The Collaboration Management Infrastructure (CMI) system [GSC00] and it commercial derivative ATLAS [ATL06], as well as others [BK95, HHJ+99], have explored relaxing control flow constraints to support some partially structured processes. Other researchers have designed systems such as Caramba [Du04] that permit either structured or ad hoc activities, or have proposed formal frameworks for dealing with dynamic process change [Wes01, RRD04]. None of these technologies supports contextualization, policy enforcement, or dynamic change [Ge04].

With respect to policies, AEC coordination policies operate in a similar spirit to that described in [DDLS01], which defines a policy as a "rule that defines a choice in the behavior of a system". We consider the subject of our coordination policies to be orthogonal to the policies that are represented in languages such as KAoS [Us+03] and Ponder [DDLS01], because our policies are focused on activities. AEC policies regulate sets of activities, the relationships among those activities, the roles of those who can do those activities, and the resources that are used by the activities.

Existing groupware tools such as Groove [Groove] typically rely on informal human coordination. When such tools are used for large scale collaboration, the overhead involved in exchanging coordination messages hinders collaboration efficiency. Another problem is that their basic capabilities for contextualizing messages and information do not scale up as well. AEC addresses this issue by scoping activities, policies and resources into appropriate contextual settings. Document sharing systems such as Vignette [Vignette] support the sharing of documents and other resources, but provide only token support for coordination. Additionally, their policy support is typically restricted to access control policies for documents.

Finally, although various workflow systems support process monitoring and groupware tools provide limited awareness on the status of shared resources, (e.g., [GGR96]), none of these technologies provides models and mechanisms for customizing situation and work related awareness to serve the needs of each user. ATLAS [ATL06] provides such capabilities but lacks contextualization and unrestricted dynamic change.

6 Conclusion

In this paper we introduced an approach for dynamic process adaptation that is driven by efforts to ensure process efficiency and to meet (possibly changing) goals of the dynamically adapted process. We also described the design of the Awareness Enabled Coordination system that supports this approach. These provide a solution for dynamic (run time) process adaptation that is driven by continuous evaluation at run time. We described the benefits of the proposed solution in the area of fire emergency mitigation, where we illustrated the effectiveness of the proposed adaptation approach to manage the activities and communication of multiple organizations that deal with a fire fighting emergency. The proposed process adaptation can be applied in several other applications domains, including physical security, military operations, time-based competition in real time enterprises, and medical safety, and can provide similar benefits, which we intent to illustrate in future work.

References

[ATL03] Telcordia: ATLAS, http://www.argreenhouse.com/ATLAS/, 2003.

[BEA06] BEA: WebLogic Integrator, http://www.bea.com/, 2006.

[BGSC02] D. Baker, D. Georgakopoulos, H. Schuster, and A. Cichocki, Customized Process and Situation Awareness, *International Journal of Cooperative Information Systems*, March 2002.

[BK95] D. Bogia and S. Kaplan, Flexibility and Control for Dynamic Workflows in the wOrlds Environment, Proceedings of *ACM Conference on Organizational Computing Systems*, 1995.

[BPEL] Updated: Business Process Execution Language for Web Services Version 1.1". http://www-106.ibm.com/developerworks/webservices/library/ws-bpel

[COS03] COSA Solutions, COSA product suite, http://www.cosa.de/, 2003.

[DDLS01] N. Damianou, N. Dulay, E. Lupu and M. Sloman, The Ponder Policy Specification Language, Proceedings of *Workshop on Policies for Distributed Systems and Networks*, Springer LNCS, 2001.

[Di02] F. Dignum, Abstract Norms and Electronic Institutions, Proceedings of *Int'l Workshop on Regulated Agent-Based Systems: Theories and Applications*, 2002.

[Du04] S. Dustdar, Caramba: A process-Aware Collaboration System Supporting Ad hoc and Collaborative processes in Virtual Teams, *Distributed and Parallel Databases*, D. Georgakopoulos (ed.), Vol. 15, No. 1, Jan.2004.

[FIL06] FileNet: Business Process Manager, http://www.filenet.com/, 2006.

[GGR96] Gutwin, C.; Greenberg, S.; Roseman, M.: Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation, People and Computers, Proceedings of *HCI'96*, 1996.

[GNBC06] Dimitrios Georgakopoulos, Marian Nodine, Donald Baker, Andrzej Cichocki, Awareness-based Collaboration Driving Process-based Coordination, *Proceedings of the Second International Conference on Collaborative Computing (CollaborateCom'06)*, 2006

[GSC00] D. Georgakopoulos, H. Schuster, D. Baker, and A. Cichocki, Managing Escalation of Collaboration Processes in Crisis Mitigation Situations, *Proceedings 16th International Conference on Data Engineering (ICDE'00)*, San Diego, 2000.

[Ge04] D. Georgakopoulos, Teamware: An Evaluation of Key technologies and Open Problems, *Distributed and Parallel Databases*, D. Georgakopoulos (ed.), Vol. 15, No. 1, Jan. 2004.

[Groove] "Groove Virtual Office", http://www.groove.net.

[HHJ+99] P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, and M. Teschke, A Comprehensive Approach to Flexibility in Workflow Management Systems, Proceedings of the *Int. Joint Conf.on Work Activities Coordination and Collaboration (WACC'99)*, 1999.

[NFK00] M. Nodine, J. Fowler, T. Ksiezyk, B. Perry, M. Taylor, A. Unruh, Active Information Gathering in InfoSleuth, *International Journal of Cooperative Information Systems*, Vol. 9, No. 1, 2000

[PynT02] David Pynadath and Milind Tambe: Multiagent Teamwork: Analyzing the Optimality and Complexity of Key Theories and Models, AAMAS, 2002

[RRD04] S. Rinderle, M. Reichert, P. Dadam, Flexible Support of Team Processes by Adaptive Workflow Systems, *Distributed and Parallel Databases*, Vol. 16 No. 1, July 2004

[RVO04] A. Ricci, M. Viroli and A. Omicini, Role-based Access Control in MAS using Agent Coordination Contexts, Proceedings of *Workshop on Agent Orgainzations: Theory and Practice*, AAAI Press, 2004.

[SAP06] NetWeaver, SAP, http://www.sap.com.

[Tam97] Milind. Tambe: Towards flexible teamwork, JAIR, 7:83-124, 1997

[Us+03] A. Uszok et.al. KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction and Enforcement, Proceedings of *IEEE International Workshop on Policies for Distributed Systems and Networks*, IEEE press., June 2003.

[Vignette] Content Management and Portal, Vignette Solutions. http://www.vignette.com.

[WfMC] The Workflow Management Coalition, www.wfmc.org.

[Wes01] M. Weske, Flexible Modeling and Execution of Workflow Activities, Proceedings *31st Hawaii Int. Conf. on System Sciences*, Software Technology Track , 1998.

[Zeng04] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, Marlon Dumas, Jayant Kalagnanam, Henry Chang: QoS-Aware Middleware for Web Services Composition.