AFRL-IF-RS-TR-2006-146
Final Technical Report
April 2006

AWARENESS-ENABLED COORDINATION

Telcordia Technologies, Inc.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

APPROVED: /s/

JAMES M. NAGY
Project Engineer

FOR THE DIRECTOR: /s/

JOSEPH CAMERA
Chief, Information and Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** APRIL 2006	**3. REPORT TYPE AND DATES COVERED** Final Nov 2002 – Apr 2006

4. TITLE AND SUBTITLE AWARENESS-ENABLED COORDINATION	**5. FUNDING NUMBERS** C - F30602-03-C-0006 PE - 31011G PR - GENO TA - A0 WU - 03
6. AUTHOR(S) Mariane Nodine, Dimitrios Georgakopoulos	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Telcordia Technologies, Inc. 445 South Street Morristown New Jersey 07960-6438	**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFED 525 Brooks Road Rome New York 13441-4505	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER** AFRL-IF-RS-TR-2006-146

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: James M. Nagy/IFED/(315)330-3173/James.Nagy@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(Maximum 200 Words)*

Many of the technologies and software products that have been developed to support virtual team collaboration have problems with scaling upwards. While some existing technologies support many users, they provide only limited help for users and virtual teams to deal with the complexity of the environment they operate in. The Awareness-Enabled Coordination (AEC) project is to improve support for (large scale) collaboration between intelligence gathering organizations and CT analysts. This is accomplished by providing tools and services that increase the efficiency of collaborative work, especially cross-agency collaborative work, while minimizing the overhead and learning curve necessary to use these tools and services

14. SUBJECT TERMS Awareness Enabled Collaboration, virtual team collaboration, multi-organizational team collaboration, distributed, context model, context management, dynamic contextualization, policy, team coordination.			**15. NUMBER OF PAGES** 46 **16. PRICE CODE**
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	**18. SECURITY CLASSIFICATION OF THIS PAGE** UNCLASSIFIED	**19. SECURITY CLASSIFICATION OF ABSTRACT** UNCLASSIFIED	**20. LIMITATION OF ABSTRACT** UL

TABLE OF CONTENTS

LIST OF FIGURES

1. AEC Objectives

The overall goal for the Awareness-Enabled Coordination (AEC) project is to improve support for (large scale) collaboration between intelligence gathering organizations and CT analysts. This is accomplished by providing tools and services that increase the efficiency of collaborative work, especially cross-agency collaborative work, while minimizing the overhead and learning curve necessary to use these tools and services. AEC combines a variety of approaches to support this work, including the following:

Support the analyst and the manager of analysts:

- The AEC coordination framework helps to organize the user's activities and help the user coordinate with others with whom he is working.
- The AEC model is easy to understand, and uses a small number of simple concepts that map onto real-world entities and artifacts.
- AEC provides easy-to-understand interfaces to the tools that manage coordination. It must be usable in the day-to-day activities of the user.
- AEC provides references to the organizational, cultural and jurisdictional information, policies and procedures, resources, and best practices relevant to his situation. This helps train new analysts, and helps trained analysts remember information that is not used very often.
- AEC adapts to changes in the situations and settings in which the analyst is working.
- AEC provides information and awareness, both with respect to changes in the situation and with respect to the internal progress of the activities in which he is involved or for which he is responsible. This information is contextualized to the situation in which the user is working, and at a level the user can process easily. This enables him to determine the impact of these events on his work.
- AEC supports the process of structuring activities from conception through their execution. The user can structure, evolve and manage his work in a natural manner. He can begin planning or coordinating activities when he only has a basic idea of what needs to be done.
- AEC allows the user to access useful external tools from within his activities. This includes support for automatically setting up and invoking tools and automatically configuring them and populating them according to the current tasks for which they are being used.
- AEC is able to automatically execute routine and bureaucratic processes if required by the system or user.
- The environment in which activities are performed may determine the exact method for doing them, the resources they use, and the policies to which they are subject to. AEC helps the analyst to follow correct procedure for each environment for each task that he is involved with, so he can avoid mistakes or extra work. AEC informs users when they attempt to perform an activity that violates a policy, and recommend changes to activities that will maintain and recover policy compliance.

- Using AEC, analysts can record preferred, attempted and failed processes so that they can learn from their previous experience, the experience of others, or from agency procedures.

Support organizations, jurisdictions and cultures:

- Different organizations, jurisdictions and cultures have different ways of accomplishing their goals. AEC provides the ability to remember doctrine, policies and procedures, best practices, preferred resource usage, and other aspects that may be specific to an organization, jurisdiction, or culture within specialized contexts.
- AEC keeps track of how different organizations, jurisdictions and cultures are related to one another.
- AEC validates that analysts and other users within the organizations and jurisdictions are following the doctrine, policies and procedures, best practices, etc. that they have put in place.
- The management of organizational and jurisdictional information can be under the aegis of the organization or jurisdiction itself.
- AEC supports change at this level, including reorganization and merging/splitting of organizations.

Support cross-organizational groups working on a common task:

- AEC provides a comprehensive coordination framework to support teaming, policy compliance, tracking of progress, and accountability.
- AEC provides the capability to set up cross-organizational teams, involve individuals as needed, and provide support for the roles that the different individuals may take within the work of the group.
- Group membership in AEC is dynamic – able to change as the work progresses. When group members discover new tasks that need to be done or new information requirements, they need to be able to locate new and/or different individuals who can do the tasks or provide the information that cannot otherwise be provided by existing group members.
- The work of a group is affected by the different agencies, jurisdictions, and cultures of the team members. AEC facilitates collaboration and information sharing among group members, even when the membership spans agencies or organizations. This includes the validation of the activity with respect to the policies and procedures, resource preferences and best practices placed on the activity via the membership of the group.
- AEC provides key managers, facilitators and members the ability to monitor relevant aspects of the group's progress and to locate problem areas as they develop.
- AEC keeps records, for accountability purposes, of the work that the group has accomplished, the individuals responsible for each aspect of the work, and products produced by the group.

2. Approach

2.1. Technical Overview

Many of the technologies and software products that have been developed to support virtual team collaboration [Groove, Vignette, FIL06, Ge04] have problems with scaling upwards. While some existing technologies support many users, they provide only limited help for users and virtual teams to deal with the complexity of the environment they operate in. Additionally, they do not provide coordination artifacts geared towards significantly increasing efficiency in achieving team objectives.

Awareness-Enabled Coordination (AEC) is a platform designed to support effective collaboration of large multi-organizational teams, possibly operating in dynamically changing situations (e.g., disaster response). Collaborating team members may be distributed across many states or nations, be employed by or be serving multiple agencies or organizations, and may have different policies and procedures for doing similar activities; these factors conspire to make the collaboration environment very complex. Unlike other existing technologies for supporting collaboration, AEC supports the ability of its users to deal with the complexity of their environments; specifically, the policies, processes, resources and events that often arise from various contexts that reflect different organizations, jurisdictions, teams, and activities. AEC accomplishes this by providing a context model, as well as mechanisms and tools for *context management* and *contextualization*.

Context management provides models, tools and repositories that organizations, jurisdictions, teams and persons can use to model and store for reference the policies, processes, resources, and types of event that are of interest within their own scope, as well as their relationships with other contexts. Ideally, this modeling is accomplished by experts within the organization, jurisdiction or team, or by the person (for his or her own context). Initial effort required for context modeling is worth while, since it enables AEC to automate the dynamic contextualization of its activities.

Activities are performed under the aegis of one or more contexts, with which they are related either directly or transitively. When a user declares his intent to perform an activity, *dynamic contextualization* utilizes this related context information to determine automatically possible methods (e.g., processes) for doing the activity, the resources it may use, the policies to which it is subject, and the events that may apply to it. Contextualization effectively reduces the context space the user has to consider by helping users focus only on the policies, processes, resources, and events that relate to each specific activity, and filtering out those contexts and their contents that do not apply.

In addition to reducing the complexity of the collaboration environment, AEC aims to increase efficiency in achieving team and personal objectives by providing capabilities that support for the following:

- *Team coordination and process automation*: AEC supports the specification/modeling and automation of organizational, team, and information sharing processes. AEC-supported processes are flexible, since they accommodate team and individual styles of work ranging from highly structured business processes to dynamically self-organized work. This increases efficiency by streamlining efforts to coordinate and communicate among team members.

- *Ongoing policy enforcement*: AEC permits capturing policies for coordination and information sharing aspects of real-world administrative policies, doctrine and laws, and provides corresponding mechanisms for continuous and immediate policy enforcement. AEC automatically evaluates policy compliance throughout the lifetime of each activity, informs the users when they attempt to perform an action that violates a policy, and recommends actions that can maintain policy compliance. This facilitates efficiency by preventing having to redo work because current policies were not followed, proscribed resources were not used, or because the team member was not aware of current policy.
- *Situational and team awareness*: AEC provides awareness both with respect to the progress of the team towards the completion of the activity, and any situational changes that the team believes may impact their work. It automatically monitors specified information in all contexts and other external sources, analyzes it to detect specified events and event patterns, and delivers such awareness to the targeted users. This facilitates efficiency by notifying appropriate team members of potential problems in a timely manner.

Another critical requirement for supporting large scale collaboration is permitting dynamic change. Advanced capabilities for providing efficiency should not hinder the ability of organizations, jurisdictions and teams to change policies, preferred methods or available resources, or for teams or individuals to change their course of action. AEC permits *dynamic adaptation* to deal with dynamically changing situations and their impact on activities. In particular AEC's models and mechanisms for contexts, processes, policies, and events enable activities to dynamically change their process at any point (e.g., to achieve policy compliance, respond to external events, deal with changes in available resources, or utilize new/different methods for performing a task).

2.1.1 Context Management

A *context* is a mosaic of information, knowledge, resources and programs that are in the same scope or are gathered together for a particular purpose. Each context has a particular focus or *scope*, which encapsulates and bounds the information, knowledge, resources, constraints and effects pertaining to its purpose. Examples of scopes and purposes include organizations, jurisdictions, teams working towards specific purposes or on specific activities, and people. Administrators with appropriate authority and training use AEC context management tools to populate and maintain their contexts, storing their current information and knowledge in AEC repositories, and making it accessible within their own context and to related contexts.

An AEC context consists of a *scope* and a set of *context elements*. Currently, AEC context elements include directly accessible *policies*, *resources*, and *methods* that have meaning relative to the context. Context elements may include references to or copies of elements in other contexts. The context scope provides *referential relationships* to other contexts containing relevant policies, resources, methods, etc. The intent of the scope is to provide boundaries on the visibility and accessibility of elements in other contexts. A set of contexts that are interconnected with context references forms a *context netowrk*.

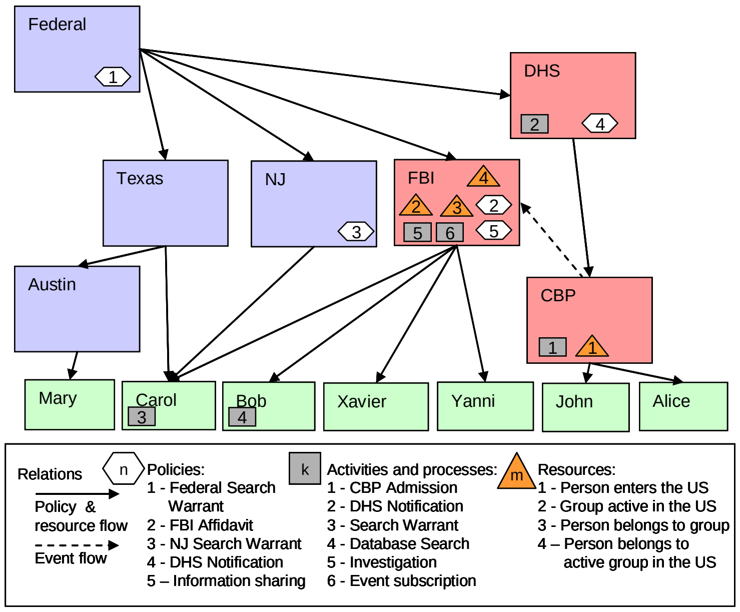


Figure 1: Context network example

Figure 1 depicts an example of a AEC context network from the intelligence gathering domain, depicted as a set of contexts and a set of referential rlationships. Two types of referential relationships are depiected in Figure 1: *policy and resource flow* and *event flow*. Referencetial relationships are defined in the opposite direction of the arrows in Figure 1. For example, the policy and resource flow relationship between DHS and CBP is defined by including DHS in CBP's scope. Event flow is determied by publish/subscribe relationships; in Figure 1.

Each organization (e.g., government agancy) has its own *organizational context* that typically includes its resources, policies and processes. Large organizations may have a hierarchy of organizational contexts related via (possibly typed) referential relationships that mirror its organizational structure. For example, in Figure 1 the Customs and Border protection (CBP) organization is a part of the Deprtment of Homeland Security (DHS). All DHS policies and processes apply to CBP, and DHS resources may be usable by the CBP (unless they are access restricted). This is reflected in the relationship between the two contexts.

Jurisdictional contexts (e.g., the Federal, Texas, NJ, and Austin contexts in Figure 1) maintain the laws, policies, processes, and resources (e.g., the database of people entering the US, and the roles of judges for issuing a search warrant), and other information for each jurisdiction. The Federal context is typically at the top of the goverenment's jurisdictional context network. Changes in both jurisdictional and organizational contexts are relatively infrequent.

The contexts at the bottom of Figure 1 are personal contexts for the individuals that belong in different organizations. These individuals may be subject to multiple jurisdictions. For example, Carol's context in Figure 1 is subject to the policies in FBI, Texas, and New Jersey. This is indicated by the "policy and resource flow" relationship between Carol's context and the FBI, Texas, and NJ contexts.

Increasingly complex context interrelationships apply in situations where multi-organizational teams (such as a task forces and emergency preparedness teams) include members that operate under multiple juristictions and organizational policies (This case is not depicted in Figure 1). For example, suppose that a virtual team of FBI and CBP agents is formed to investigate a suspected terrrorist that enters the US. This will involve the addition of a new team context in the context network in Figure 1. The scope of this context will be set to include the FBI and CBP contexts (as well as DHS via CBP). The vitual team members, say John and Carol, will be subject to the policies specified in the new team context, their organizational contexts (i.e., FBI and CBP), as well as the jurisdictional contexts of the states they operate. Processes, resources, and events in these contexts that are relevant to the team-related activities performed by John and Carol in will be detemined via dynamic contextualization.

2.1.2 Dynamic Contextualization

Every activity in AEC is done within the direct scope of one or more contexts. These contexts constrain how the activity may be performed by providing policies, resources and suggested methods related to the activity, and provide capabilities for awareness of progress and events related to the activity. The determination of the set of policies, resources, methods and events that may apply to the activity at the current time is called *dynamic contextualization*. The accuracy and completeness of the contextualization is dependent on the quality of the context management.

When a team or user states his intent to perform an activity, AEC's contextualization mechanism performs the following steps:

1. *Dynamically determine the environment of the activity*: Consider C to be the context of an activity A. AEC explores specified referential relationships from C to other contexts in C's scope. The elements of C together with the elements of the contexts referenced (either transitively or directly) by C together form the environment of A (which is also the environment of any other activity that runs in C). Therefore, the environment of any activity running in context C is the context network with C as its root.
2. *Reduce the environment of the activity to the set of elements that are relevant to it*: For a given activity of a given type, this set includes all elements that are compatible with the type of the activity, i.e., methods having a type compatible with the activity type, resources of a type that can be utilized by the activity type, and policies that refer to the activity type. Where alternatives exist, AEC either selects preferred alternatives (where they are well-defined) or allows users or team members to make the choice explicitly.
3. *Map relevant elements from the activity environment to the activity context*: This is accomplished by mapping (adding references to or copies of) the elements determined in step (2) to the activity context. For resources, when the activity context expects the same type as the resource located in its environment, this mapping is automatic and straightforward. If the activity context expects a different (but semantically equivalent) type of resource than the one selected in step (2), then AEC may be able to copy the resource and map it into the expected type, possibly using ontology- or semantic- based approaches similar to those proposed for data integration in federated databases and agent-based systems, e.g., in [NFK00]. These approaches are outside the scope of this paper.

In the presence of dynamic change, contextualization involves having each activity constantly track the contexts and context relationships, adapting the activity to relevant changes, establishing access to new elements in other contexts, and dropping references to elements that became inaccessible.

2.1.3 Process-based Coordination and Automation

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 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 their 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].

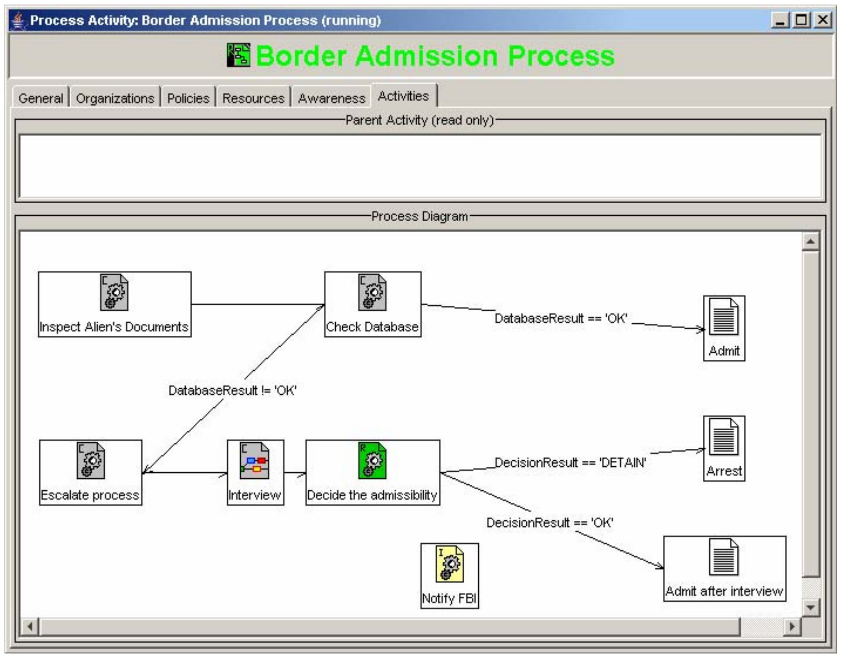


Figure 2: A flexible process in AEC

As an example of a partially structured coordination process, consider the graphical process specification in Figure 2. This specification shows the activities of a notional border admission process, such as might be used by Customs and Border Protection (CBP). Boxes in the diagram represent activities, with the intent of the activity indicated as text within the box. Control flow dependencies are shown as arrows in the figure. Text on the arrows indicates conditions on the control flow based on some data resource in the activity's context. The "Notify FBI" in Figure 2 activity has no dependencies and can be executed at any time. Both child activities and dependencies may be added to the process as they become apparent. Note that the addition of control flow dependencies to order the execution of the "Notify FBI" activity would convert the partially structured process in Figure 2 into a fully structured process (e.g., a traditional business process), while relaxation of control flow dependencies would result in an unstructured process (as it would have been if these activities were performed in an ad hoc manner).

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 specified according to the domain-specific activity ontology). As the team obtains more information concerning the details of what needs to be done, the members may refine it incrementally 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. Ongoing policy enforcement is discussed further in Section 2.1.4.

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.

2.1.4 Coordination Policies and Ongoing Policy Enforcement

Policies in AEC are constraints in the execution of activities. Policies currently supported by AEC include:

- Coordination policies ensure that the child activities of a process are coordinated appropriately with other child activities. For example, a coordination policy might state that if a Border Control officer admits an alien after conducting an interview at the border, that the FBI must be notified.
- Information sharing policies coordinate how information is shared among teams and organizations. For example, an information sharing policy might state that all inquiries from a Border Control officer to the FBI must go through the appropriate liaison.
- Role resolution policies ensure that the people who are performing certain activities fill certain roles in the company. For instance, a role resolution policy might constrain the interviewer in a border crossing situation to be a Border Control officer of a certain level or above.
- Resource utilization policies constrain how or which resources can be used at a given point. For instance, a resource utilization policy might constrain a report concerning a border interview to be completed according to a particular form supplied by the FBI.

In AEC, a policy is defined by a *policy locus* and a *policy effect*. The locus of a policy P is a precondition that determines in what context the P should be evaluated, while the effect of P is a post condition that determines whether a policy P is been satisfied or not. Both policy locus and effect are defined by constraints that are specified using AEC's policy specification language.

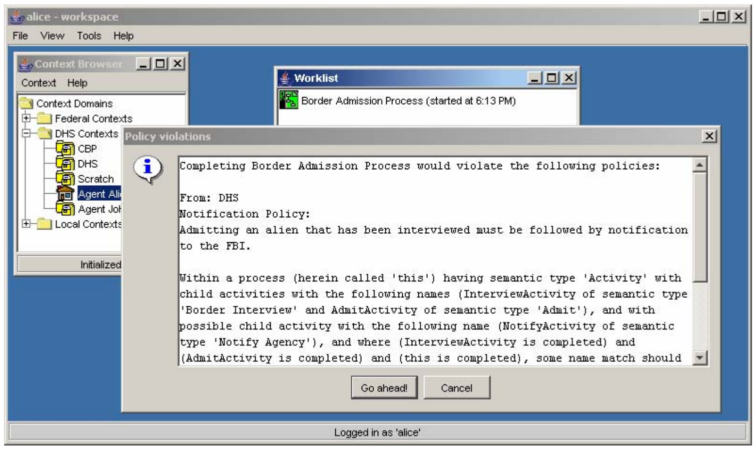
Coordination policies specify relationships between activity intents (i.e., the semantic activity types defined in AEC's activity ontology). In particular, the specification of the locus and effect for coordination policies involves variables of activity intents, activity states, Boolean operators, and temporal operators (e.g., before, after). As an example, consider again the coordination policy that requires the notification of the FBI whenever a Border Control officer admits an alien after conducting an interview at the border. This policy is relevant to the border admission process in Figure 2. In particular, this policy relates the instances of activities with intent *Admit after interview* (these are represented by an activity variable A), and the instances of activities with intent *Notify FBI* (these are represented by an activity variable B). To define that this policy should be evaluated whenever a user attempts to start an activity instance, the policy locus might specify "A of semantic type *Admit after interview* and A is *Ready*". To avoid a violation of the policy, the policy effect may specify "B of semantic type *Notify FBI* is *Completed*". If the policy locus is satisfied, then the process must also contain an activity with intent *Notify FBI*. If the process does not meet the policy effect, then the policy is violated, and the activity A is not allowed to start (e.g., the responsible party for A may terminate it).

AEC provides a comprehensive policy specification language, but its details are outside the scope of this paper. Specification of resources utilization and role resolution policies involve the definition of utilization constraints between semantic activity and resource types (roles are resources in AEC). Information sharing policies involve coordination, resource utilization, and access control constraints. To assist in policy definition, AEC provides a policy wizard that facilitates the definition of many of the types of policies that AEC can enforce.

Policies are typically defined within the jurisdictional, organizational, and team contexts. For example, the FBI would define FBI-engendered policies within the FBI context. As described in Section 2.1.1, AEC contextualization determines which contexts (and therefore which policies) are in the environment of each activity. When an activity is executing, AEC dynamically determines the set of policies that apply to it using its contextualization mechanism. Each time an activity changes execution state, the AEC policy validation mechanism performs the following:

1. Determine which policies in activity's environment apply to the activity because the constraint in the policy locus is satisfied.
2. For each such policy, determine whether the policy effect is present (i.e., the constraint in the policy effect is satisfied) in the activity.
3. If the effect of the policy is not present, notify the responsible party for the activity of the violation.

Figure 3 shows a policy violation notification related to the Border Admission Process in Figure 2. Assume that the Border Admission Process does not have the "Notify FBI" activity. In this situation, the policy violation in Figure 3 would be sent to Alice, the responsible party for this instance of the Border Admission Process. If Alice were to rectify the situation by adding a "Notify FBI" activity to the process, the resulting process would be identical to that in Figure 2.



```
Completing Border Admission Process would violate the following policies:

From: DHS
Notification Policy:
Admitting an alien that has been interviewed must be followed by notification
to the FBI.

Within a process (herein called 'this') having semantic type 'Activity' with
child activities with the following names (InterviewActivity of semantic type
'Border Interview' and AdmitActivity of semantic type 'Admit'), and with
possible child activity with the following name (NotifyActivity of semantic
type 'Notify Agency'), and where (InterviewActivity is completed) and
(AdmitActivity is completed) and (this is completed), some name match should
```

Figure 3: Policy violation notification in AEC

Note that, since the organizational, jurisdictional, and other contexts in the activity's environment are maintained independently, there is no guarantee that different contexts may specify conflicting policies with respect to a given activity. In this situation, the responsible party might receive conflicting notifications, each specifying the policy that was violated. In this situation, AEC does not attempt to correct the situation directly, but rather places the onus on the responsible party to determine either whether the process can be made conformant to both policies or which of the conflicting policies can be violated, with the resulting consequences.

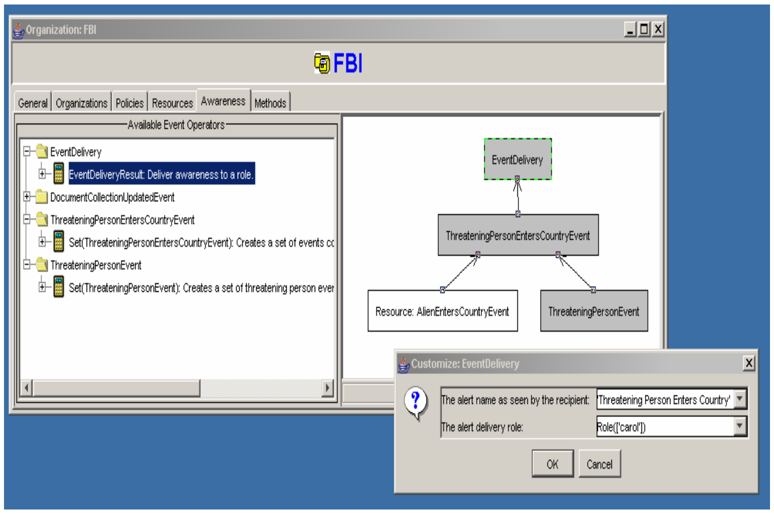
2.1.5 Awareness

We define awareness as the stream of events that carry highly relevant information to a specific user role and situation. Because a human's attention is a finite resource that must be optimized, awareness events must be digested and delivered to exactly the users who need them via alerts. The information in the awareness event must be related to the context that gave rise to it. Note that, if awareness events provide less information or they are targeted improperly, users will act inappropriately or be less effective. Users receiving too many or uninteresting events must deal with an information overload that adds to their work and masks important information.

Awareness in AEC is computed according to user-defined *awareness specifications* that permit any AEC user to specify the following:

- what are the events of interest,
- how AEC should detect them, i.e., when, where (from which source context), and which method (e.g., AEC-provided event stream filter, join, or aggregation operator) to use, and
- who should be alerted

Awareness specification in AEC involves building graph diagrams consisting of customized, interconnected event operators. Figure 4 shows the AEC awareness editor being used to specify a *Threatening Person Enters Country* event in the FBI context. The graph diagram for this specification is depicted in the right side of Figure 4.



EventDelivery
EventDeliveryResult: Deliver awareness to a role.
DocumentCollectionUpdatedEvent
ThreateningPersonEntersCountryEvent
Set(ThreateningPersonEntersCountryEvent): Creates a set of events cc
ThreateningPersonEvent
Set(ThreateningPersonEvent): Creates a set of threatening person ever

The alert name as seen by the recipient: 'Threatening Person Enters Country'
The alert delivery role: Role(['carol'])

Figure 4: An awareness specification in AEC

AEC supports generic relational algebra (e.g., filter, join, aggregation) and temporal algebra operators (e.g., before, after), as well as alert (e.g., event delivery) operators. AEC users typically create a customized, domain-specific palette of operators. New customized operators can be added to an AEC palette at any time. Operator customization is necessary for the following reasons:

- Events in AEC carry information in the form of event parameters concerning the situation described in their source context. Users need to customize operators to pull the most relevant information out of the constituent/input event(s) or summarize such information in the parameters of the computed composite/output event. Information provided on alerts is available at alert delivery time for user inspection. This computation of event parameters by event operators enables AEC to automatically generate the information on an alert that the user is most likely to need.
- Users need to associate each operator to a specific context. In addition, alert operators need to be assigned to a specific role in the operator context.
- To improve ease of use by non-experts, it is desirable to provide operators relevant to domain-specific functions that they can understand easily. For example, the ThreateningPersonEntersCountryEvent operator in Figure 4 has been specifically created for the FBI context. Its advantage is that it has a clear meaning in the FBI context and it requires little of no customization when it is used in this context.

Operator customization in AEC may be performed via dialog boxes or via a programming language. To define awareness specifications, users drag operator instances into the specification from a palette of operator types, as shown on the left side of Figure 4. To customize an event operator, users invoke an operator-specific dialog box.

The edges between the operators represent the flow of events from producer to consumer; as shown on the right side of Figure 4. The leaf operators of an awareness specification derive events from AEC resources, which we refer as *basic* events. Interior operators combine one or more such basic events and generate *composite* events that describe a situation that is more specific than the situations giving rise to the input events. An *EventDelivery* operator at the root of an awareness specification graph acts as delivery instructions for the input events. As shown in the dialog box in Figure 4, the *EventDelivery* operator can be customized to change the title of the alerts and a role of people to whom to direct the alerts. Awareness operators computed output events related to a specific context. For instance, all operators in the awareness specification in Figure 4 are context-specific, i.e., they compute awareness from resources of the FBI context and deliver it to a role defined in this context.

Awareness is computed incrementally and continuously. Incremental computation enables awareness to be delivered to targeted users in a timely fashion. Alerts are not just generated once, but can evolve over time as relevant information is updated. Awareness specifications can also be edited “on the fly” with the resulting alerts immediately recomputed.

2.1.6 AEC's Run-time Architecture

The AEC platform is comprised of a collection of contexts distributed over a set of nodes that support these contexts. The AEC architecture is depicted in Figure 5.

Contexts visible to the AEC users are depicted in the upper plane in Figure 5. As we discussed in earlier sections, examples of AEC contexts include those for organizations, jurisdictions, activities, and individuals. The arrows between contexts represent the referential relationships specified between them. The contexts themselves reside on nodes distributed across the network, hosted in different locations and by different organizations or agencies. Each node hosts one or more contexts and provides a set of system services to them, such as messaging, naming, event, etc. Nodes are illustrated in the lower plane in Figure 5.

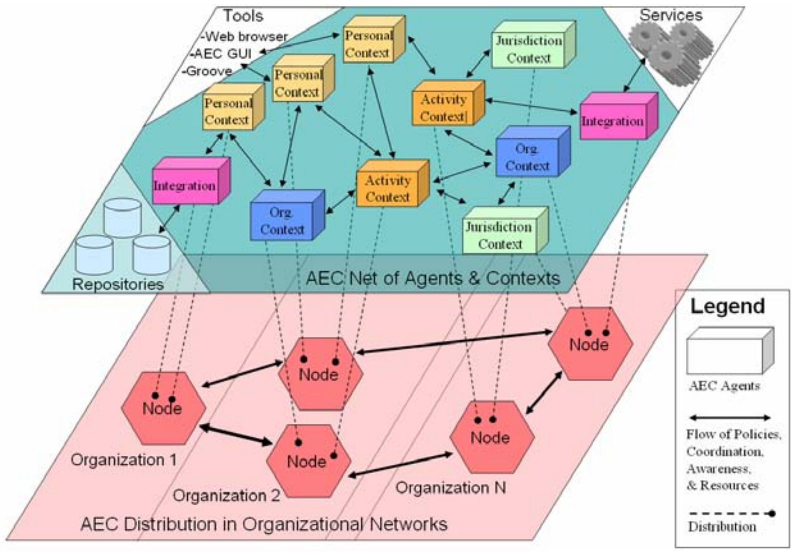


Figure 5: AEC architecture

2.1.7 Dynamic Change and Refinement

AEC is designed and built to permit dynamic change and refinement. Change can come in the form of organizational restructuring, changes in policies and goals, or changes in available resources such as human, physical, or financial resources. Human decision making can be another source of change to the extent that each person's decisions can impact the environment of others. In order to be effective, decision making must be informed by an up-to-date understanding of his or her situation, including information relevant to each person's work. In addition to supporting change, dynamic adaptation also involves the capture of the response so that improvements can be remembered and shared with others. These improvements enable the evolutionary improvement of best practices of an organization.

AEC supports dynamic adaptation through its mechanisms for process, policy, and awareness. Processes in AEC support continuous refinement, a mechanism that allows the specification of a process to overlap with its execution. Such a process can be refined, as needed by the situation by end users who, in many cases, must make decisions based on the details of their situation. Refinement can include the addition of activities, the selection of appropriate activity methods, adding or modifying resources, or inter-activity dependencies.

2.1.8 Summary

An organization can use AEC coordination policies to constrain the specification and behavior of processes being refined by end users. Managers of an organization can create coordination policies to ensure compliance to organizational goals and rules. When combined with an organization's method catalogue, managers of an organization can provide the appropriate amount of flexibility for end users to effectively do their work. This flexibility spans a range of work styles from highly structured to dynamically self-organized teams. Coordination policies are continuously enforced with end users receiving guidance as to how to perform their work in a policy compliant way.

Awareness facilitates human understanding of dynamic change by harnessing changes to resources and other time-based information into information that is highly relevant to the work and roles of each user. Properly informed users can make better decisions and they are empowered to do so through authoring policies, selecting activities to execute, refining the processes in which they play a part, or changing the structure of resources or even organizations. The various aspects of dynamic change in AEC, process, policy, and awareness, are in turn supported by AEC's mechanisms for contextualization and resource flow. The combination of these capabilities helps users and virtual teams deal with the complexity of their environment, and provide efficiency in achieving team objectives. These capabilities make AEC suitable for supporting large scale collaboration.

2.2. Methodology

2.1.1 Science Involved & Concepts Employed

In looking at how to build the Awareness-Enabled Coordination system, Telcordia examined in detail several related fields, in which we have expertise. These included *process-based systems*, particularly those that did not require rigid structuring of processes (such as workflow systems), *agent-based systems*, and the representation of concepts through *ontologies*, work on *policy representation*, execution domains and *contexts*, *publish/subscribe paradigms* and *stream-based query processing* suitable for awareness.

Process-based Systems: Coordination within AEC is not intended to be ad-hoc, but rather AEC tries to provide features that allow for the work to be structured as a set of coordinated activities. To that end, we examined flexible process-based technologies, including our own ATLAS system [ATL03] and others as described in the related work above. The concepts we took out of the process-based system world include the structuring of activities via *control and data dependencies*, and the concept of *roles* within a process.

Existing process-based systems including 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 [ATL03], as well as others [BK95, HHJ+99], have explored relaxing control flow constraints to support some partially structured processes. Caramba [Du04] permits either structured or ad hoc activities. Both [Wes01] and [RRD04] have proposed formal frameworks for dealing with dynamic process change. 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].

Within process-based systems, activities represent units of work. Each of the activities in a process may either be done automatically, or may be assigned to a person or a role. A role is a description of one or more people in an organization that have a particular responsibility. The person or persons filling the role may change over time. For example, the activity of reporting expenditures to the government regarding a particular contract may be assigned to the role of financial contact for that contract.

Processes structure their activities via control and data dependencies. Control dependencies imply that a particular activity must start after some preceding activity has changed its state – e.g., all expenditures with respect to a contract should be accounted for before the financial report and invoice are sent to the contracting agency. Control dependencies may be conditional as well; for example, the above dependency could be conditioned on whether or not the contract is being worked on at risk (and therefore may or may not be able to be invoiced). Data dependencies imply that some information generated by one activity serves as an input to another activity. For instance, the expenditures for the month should serve as an input to the activity that generates the invoice.

We deviated from classic process-based systems in several aspects. These include the *separation of intent from method*, where intents are defined using terms from an ontology of intents. This enables *incremental refinement* and *refinement during execution*, and the ability to specify the methods used by AEC activities as being done by *humans, tools, or subprocesses*. These unique capabilities of AEC that gives it the agility to better support the coordination of humans in large-scale and cross-organizational teams.

The ability to refine a process incrementally means that a process does not need to be fully-specified before it can be used. In fact, a process may be predefined, may be transformed from a related or slightly obsolete process, or even be defined from scratch. Thus, the knowledge that a user needs before he can begin working with a process-based activity in AEC can range from minimal to complete. Complete knowledge is suitable in situations where there are known business processes or known procedures for doing a specific task. If the characteristics of the task are slightly different from a previous one, then the user can start from either an existing or more abstract process, and refine from there. For example, there are well-defined principles and activities related to producing a proposal to the government. A company can specify a general process involving acquiring a charge number to do the bid and proposal, writing the technical proposal, costing the proposal, incorporating subcontractors, and having a red team review. However, the producing of a particular proposal may involve elaborating on some of these steps. Additionally, there may be a fixed process for, say, acquiring a B&P number, but a less-defined process for, say, writing some section of the proposal. Incremental refinement allows the process to be defined at varying levels of detail at any given point.

The ability to begin executing a process before it is fully defined is another novel aspect of AEC processes. An activity may execute once it is defined enough to begin to execute, it has adequate resources to begin, and its dependencies are satisfied, If an activity is not ready to execute, then AEC will delay starting it to allow the necessary refinement, resource assignment, and precursor activity completion. For long-running, sketchily-defined activities, this capability enables the user to define and begin executing the earlier parts of an activity before he even knows how to tackle the later aspects of the task. Thus, the task coordinator can “learn as he goes”. One additional advantage of this is that the parts of the process geared towards reacting to exceptional circumstances do not need to be addressed unless the exceptional circumstance actually occurs. While standard recovery methods may be available to facilitate recovery, the user need not concern himself with them unless necessary.

Agent-based Systems: Because we though from the start that contexts should be ubiquitous, we began with the belief that contexts should be accessible via an agent-based paradigm. That is, a particular context in the context lattice (jurisdictional, organizational, activity, or team) should be accessible via an agent (or possibly, but not to the same effect, via a web service). Agents are a natural match with AEC contexts, because agents assume that they are autonomous, and agent-based systems are built around the presupposition that each agent is an autonomous unit. Similarly, AEC contexts are assumed to be maintained autonomously by the jurisdiction, organization, team or person that they represent.

One key issue with respect to the similarity between agents and contexts relates to crafting the interactions between contexts such that they are interacting at a semantic level concerning the policies, resources and methods that they convey to each other. We did not have either the time or the opportunity to bring this part of the vision into fruition in the implementation, though it is envisioned clearly in the architecture.

A second area that AEC loosely borrowed, specifically from goal-based agent systems, is the notion of *separation of intent from method*. In a goal-oriented agent-based system, an agent begins with a goal that it wants to attain, then makes a plan that it can execute to attain that goal. In AEC, we loosely adopted this idea. An AEC intent is not exactly a goal, but it does provide a guideline to the system as to what a particular activity is intended to do. You could view the intent as a desire to achieve a goal, so the ideas are very interrelated. AEC factors out the intent because, often, a user knows *what* needs to be done before he can determine *how* to do it (the *method*). Because the intent is separated from the method, incremental refinement can begin as early as the stage where only the intent is known. Thus, the plan for how an activity can be done is capturable within the AEC system at almost all potential levels of refinement, and AEC becomes a unified umbrella under which all planning and execution of an AEC activity may be captured.

Ontologies: Ontologies are intended to represent knowledge in terms of concepts and the relationships between them. The semantic web is a key driver of ontology development, with their web ontology language standard, OWL [OWL]. Ontologies are often used within reasoning systems, but have also been used increasingly in the area of information integration and fusion, e.g. [InfoSleuth]. This is because ontologies allow for information to be related and fused at the concept level, rather than at the schema level. Additionally, ontologies allow for the semantics of an application to be exposed in such a way that it facilitates interaction with other applications that share or can import the same ontology.

AEC uses ontologies to accomplish two separate objectives:

- Expose aspects of concepts within AEC that can be customized to conform to specific domains and applications. E.g. activity types, resource types and semantics, event types.
- Enable the user to formulate statements and requests in terms of concepts, e.g., locate a resource to be used for a specific purpose, subscribe to an alert of a particular type, or declare an intent.

With respect to customization, AEC provides an upper ontology which specifies generic concepts within AEC. A subset of this ontology is shown in Figure 6. The “Activity” class defines the generic activity semantics, or the set of intents that an activity can have. This can be extended via subclassing to domain-specific activities such as “Check Database” or “Admit After Interview” (from the example in the technical section). The “Resource Semantics” class defines the different semantics that domain-specific resources can have. Again, from the above example, the domain-specific subclasses for resource semantics may include “Interview Report” or “Interviewer”. Finally, the awareness event types are defined as subclasses of the “Event Record” class. An example of this, again from the technical section, would be “Threatening Person Enters Country”.

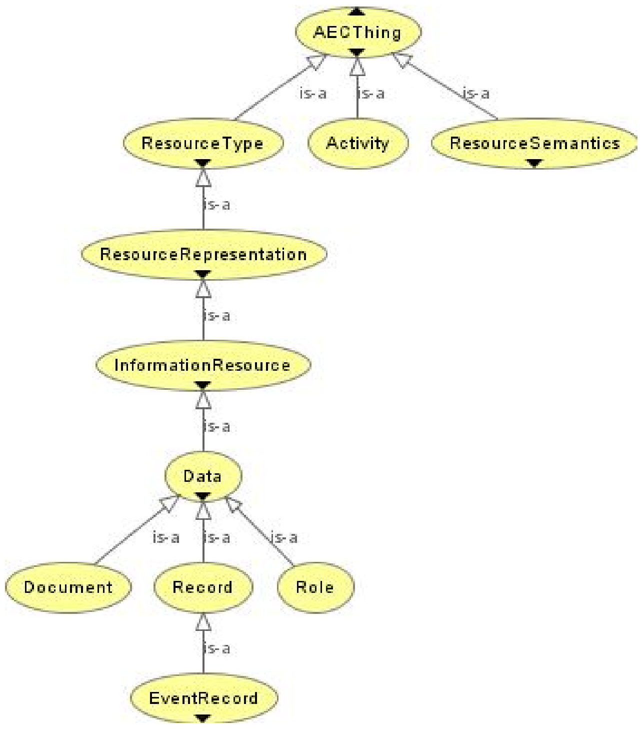


Figure 6: The AEC Upper Ontology (A Subset)

The domain-specific concepts defined in the ontology populate the tools within the AEC user interface (either the AEC GUI or the VBCS web interface). For example, the Activity Ontology Browser, implemented in the AEC GUI, allows for the browsing of possible intents in the domain-specific ontology. Intents may be selected within the Activity Ontology Browser and dragged into an active process, causing an activity with that intent to be inserted into the process.

Ontologies used in the AEC system are defined in OWL [OWL].

Context and contextualization: Context, in general, refers to the circumstances surrounding your current activities. Within AEC, context currently includes, but is not limited to, the following:

- The resources available to the activity,
- The policies that affect the activity, and
- The methods available for performing certain intents within the activity.

In AEC, we use the term "context" to refer not only to the immediate context of an activity, but also to the network of jurisdictional, organizational, and other contexts that reflect the environment in which the activity is taking place. The domains discussed by Uszok et.al with respect to KAoS policies [Us+03] are similar to our jurisdictional and organizational contexts. Agent communication contexts in Ricci et.al. [RVO04], when restricted to a single activity of an agent, would maintain similar information to an agent's personal context in AEC. Also, the information that we place in our context network relates to the types of information you would expect to see at the upper levels of institutions, within the work on representing norms and institutions, e.g. in [Di02].

However, while related projects compact the contextual policies within the domain or institution, AEC creates a composite context containing related individual, and autonomously-updatable, contexts that can be maintained explicitly by the organizations and jurisdictions whose policies and resources they represent. That is, individual contexts associated with specific individuals, activities, teams, organizations and jurisdictions are related together via scoping relationships to form a network of contexts, as was shown in Figure 6. Thus, while the contents of an AEC context are not as rich as those provided in other context models, the context network together provides a powerful model that is easily-updatable and supports dynamic contextualization.

AEC provides mechanisms and tools for both automatic and user driven dynamic contextualization. When an activity is begun, its specific context is defined by its relationships with other contexts in the context lattice. The transitive closure of the activity's context and its related contexts automatically scopes the availability of resources and methods, and the applicability of policies, to exactly those that are needed by the activity. Additionally, changes in one of the contexts within this transitive closure automatically impact the activity, including changing the availability of resources and methods, and changing the policies that apply to the activity. We know of no other system that manages the global context of an activity in this manner.

Policy Representation: Damianou et al describe the concept of policy as a "rule that defines a choice in the behavior of a system" [DDLS01]. Much of the current and recently past policy work has focused on security and access policies over resources or web services, e.g. KAoS [Us+03]. Ponder [DDLS01] and REI [Kag+03]. KAoS also supports the notion of domains for policies, which is similar to AEC contexts. A second set of policy work, a bit more related to AEC, is in the form of conversation policies for agents, which govern how agents interact as they coordinate their efforts towards some task. Examples of this include the FIPA Interaction Protocols [FIPA]. This type of policy defines sequences of messages that are appropriate towards various types of interchange between agents, for instance, asking a question or bidding in an auction. They also tend to be quite restrictive, as they are used for automatic composition of agent behavior. As AEC develops, some of these notions of policy will be incorporated into the domains in which AEC is installed, incorporating the access control policies currently in use in the installation environment.

Unlike the works cited above, the *subject* of AEC policies is on activities as opposed to resources and their security. Thus, AEC policies appear to be more related to constraints on processes. 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. These are still policies, because they govern the behavior of the activities as opposed to fixing that behavior.

Some of the policy types defined in AEC that differ or overlap somewhat with existing policy types, include:

- *Coordination policies.* These govern how activities must relate to one another in a process, for instance by proscribing control dependencies, ordering dependencies, or the existence of some activity when another is present in the process.
- *Resource utilization policies.* These govern what resources must be used by an activity. For instance, these may constrain a user to use a particular tool or have to fill out a particular form. Another use is in role assignment – for instance, a person that fills a particular role in an activity must have specific authorizations within his organization.
- *Information sharing policies.* These are procedures or processes that must be followed in order for information to be communicated from one user to another or from one activity to another. Examples of this include declassification processes such as redaction or other forms of censorship, the need to log the transmittal of information in a special log, or the need to communicate through a liaison.
- *Method selection policies.* These guide end-users in the selection of methods for unrefined activities. A method selection policy may require, suggest, discourage, or prohibit a method from a method catalog to be used in a given circumstance. This allows organizations to control how end-users refine flexible processes for their particular circumstances and gives end users more information about the choices of methods available to them for their particular circumstance.

Thus, while AEC applies the principal of policy in a similar manner to other uses of policy, it applies a policy-based approach to different aspects of the coordination problem.

Publish/Subscribe: The publish/subscribe paradigm allows a *selective push* interface for information. In this situation, information germane to the activities and tasks of a user is pushed to the user as it becomes available, changes, or becomes unavailable. This form of selective push is crucial to such military applications as the Joint Battlespace Infosphere [JBI] and the Communities of Interest work that is currently ongoing. In these systems, users locate and select information (documents, tables in database, etc.) that are of interest to them. As these items change, the user is informed of the changes.

AEC differs from these publish/subscribe systems in that subscriptions are specified with respect to *concepts*, not data. In AEC, subscriptions are specified with respect to *types of events*, and with respect to algebraic compositions of multiple events of a given type. Subscribing to types of events, at the lowest level, has a similar feel to subscribing to the set of data in a database table – as rows are added and deleted, the user is informed of these changes. However, when subscribing at the concept level, this means that the one subscription may cover all tables and other data artifacts that contain the requested information – at the subscription level the user does not need to be concerned with where the information is actually located, or in what format. This approach is similar to that taken in InfoSleuth [InfoSleuth, NFK00].

The second issue where AEC differs is that it provides an algebra to compose events into more complex events that represent higher-level concepts. For instance, activities such as renting trucks, purchasing fertilizer, and purchasing timing devices look benign when viewed as individual events – however, when viewed in combination may be indicators of a more malicious intent of which the user may want to be aware. AEC allows the user to specify patterns of normally benign events that together combine into an event of interest, and to monitor for those patterns. This frees the user from the information overload of receiving and processing the lower-level events, many of which are insignificant to his task.

This approach is similar to the approach taken in the Video Event Awareness System [VEAS], which is a part of DTO's VACE Phase II program.

Finally, none of these technologies provide models and mechanisms for customizing situation and work related awareness to serve the needs of each user although various tools provide awareness on the status and content of shared resources and information, (e.g., [GGR96]). AEC allows this by the ability to extend the ontology to define new types of awareness event and to customize event operators and the composition of different event streams and event operators to define domain-specific computations of given awareness events.

Stream-based Query Processing: Stream-based query processing techniques served as an initial basis for processing awareness events in AEC. Stream databases [STREAM, ACC+03] enhance SQL, a relational query language, with time-based windows, allowing for real-time queries over multiple data streams. Operators are based on generic relational algebra operators including selection, projection, join, aggregates, and group-by. To perform reactive information and event processing, steam databases assume that there is no late arriving information. Optimization assumes that events/information is readily available – that is, it does not consider either the cost of extracting the information in any of the information streams, nor does it consider delays that may be caused, for example, because the information needs to be transferred between agencies or organizations.

In AEC, awareness is produced in a stream-like manner. However, the set of operators for events can be tailored to include domain-specific or application-specific types of event operators. Also, because AEC event processing is intended to be able to combine events potentially generated from multiple, geographically-distributed agencies, it performs incremental, rather than purely stream-based computation. This means that event information is kept around for long enough that late-arriving information and events can still be factored into the computation of a more complex event.

2.1.2 Design Criteria Established

AEC has a broad vision of ubiquitous contexts and a coordination system supporting correct, large-scale coordination of activities that cross national, organizational, and cultural boundaries. Because of this, it is possible in the long run that it will interact with a wide variety of applications, people, and systems. This factor greatly influenced the design criteria that we used in developing AEC. Specifically, this vision impacted the design of AEC in the areas of usability, scalability, openness and robustness.

Our principal design criteria with respect to usability included the following:

- Work as hard as possible to make AEC easy and natural to use.
- Reduce the new concepts required to understand what the system can do for the user to a few intuitive concepts – in this case, these included contexts, processes, policies and events.
- Provide capabilities that allow the user to get a quick return on his investment to learn AEC by such capabilities as capturing processes and methods, and keeping track of pedigree and accountability information automatically.
- Allow for the user to specify and customize what he needs at the level of detail that he needs. Do not present a system that is either overly constraining or inadequate in its support.
- Provide mechanisms by which part of an activity can be done within AEC and part not within AEC, albeit while sacrificing the ability for AEC to enforce policies and maintain accountability information for activities done outside of AEC.
- Provide very complete user documentation, while at the same time providing user interfaces that are intuitive to learn.

Our principal design criteria with respect to scalability included the following:

- Develop a scalable architecture, with respect to number of contexts and people.
- Factor the contexts such that each organization, jurisdiction, team or individual has control over its own contextual information. This allows actual changes to be scoped and localized, and the effects of a change to be propagated more globally, as they apply to the different activities.
- Be able to incorporate increasingly heterogeneous applications and resources. Provide methodologies and tools to facilitate this.
- Keep the vision that AEC tasks may be international, inter-organizational, and inter-cultural.

Our principal design criteria with respect to openness include:

- Use and work with existing applications, services and resources as much as possible. This criterion led us to use OWL as our ontology definition language and Jena as our reasoning language within the ontology service.
- Be conformant to existing standards, especially those in use within the Intelligence Community.

- Provide an open architecture, along with paradigms and methodologies that allow users to incorporate their own tools, applications and other resources into AEC activities.

Finally, our principal design criteria with respect to correctness, reliability and security included:

- Use a policy-based approach to govern application correctness and the sharing of resources in a way that maintains its classification and confidentiality.
- Design the architecture such that it can deal with situations where the environment becomes more unreliable, e.g. contexts become unreachable.
- Be able to incorporate access control and other existing security approaches into the overall AEC access control and security mechanisms.

Note that, while these criteria all were considered in the design and architecture work for AEC, not all of them were actually implemented.

2.1.3 Procedures and Processes

Because we were moving into uncharted territory, Telcordia began the work by researching the literature, looking at related work from the internet, and drawing on our own vast experience to try to characterize the problems that we needed to solve and to locate approaches that we might bring to bear on the research problem itself. We then spent the first year of the project putting together the overall concepts and architecture that we felt best addressed the requirements that we had developed. This resulted in a white paper and a prototype demonstration at the end of the first year.

Once the white paper was complete, including the basic architecture and design, we took a spiraling development model to bring the product to increasing levels of sophistication in each subsequent year. Since many of the concepts in AEC are interrelated, and work synergistically to provide the most impact on the end users, we felt that the spiraling development model was a good approach.

Starting with the first, proof of concept demonstration at the end of the first year, one of the annual goals of the AEC project is to produce annual demonstrations showing off the accumulated features of AEC. This resulted in the Terrorist Intercept demonstration at the end of the second year of the project (which was also kept up-to-date for the remainder of the project) and the Short Coronado demonstration at the end of the third year of the project.

3. Results

3.1. Observations

Overall, we made a huge amount of progress over the three years of the contract. The problem that we tackled originally really is a set of interlocking problems that must be solved in a more interleaved fashion – thus, our research challenge was greater than that of many other projects because we needed to figure out how to proceed forward with several thorny problems simultaneously. Some examples include:

- When someone is given a particular task to do, they sometimes have a really good idea of how to do parts of it while at the same time have little insight into how to do other parts. How do you support this with process-oriented concepts, given that parts of the process are very well-defined while other parts are unknown?
- How can a company ensure that certain proscribed policies and procedures are followed as their employees go about their work? How can they make their employees aware of when they are inadvertently violating a policy?
- How can you represent an environment in which a single, focused change (with respect to collaboration) in the real world is reflected in a single, focused change in the collaboration system that will be reflected in current and ongoing collaborative tasks?

For many of these problems, we were able to think through the issues and develop new technical approaches that ameliorated them. For the above three problems, for instance, we came up with the following three ideas which, as far as we know, are completely novel to AEC:

- Flexible processes and the separation of intent from method in process definitions. When a user knows or has a known procedure for some activity in his task, he can insert both the intent and the method for that activity with one drag-and-drop operation. When a user only knows what needs to be done for a part of the task, he can insert only the intent, and then refine the method incrementally as he learns how to do it.
- The specification of coordination policies that constrain the form of processes and the use of resources within activities in processes. We note that KPAT [Us+03] supports some of these ideas, but the full notion of coordination policy is only present in AEC.
- The representation of the environment as a network of jurisdictional, organizational, and other contexts, each of which is (preferably) maintained by the jurisdiction, organization, etc. that it represents. The intent of this is to make contexts ubiquitous, with each entity that maintains a context in control of the information that it represents and makes available.

It has been clear both to us and to the program managers that we have made a lot of technical progress over the three years of the project. Project observers such as the members of the ISAI team were beginning to see the various application areas in which the AEC functionality would be useful and add to the efficiency of the teams. Also, parts of the functionality, particularly the ability to capture human process, recently had begun to capture the interest of people in some of the government agencies.

We have validated our ideas in implementations and in demonstrations. We produced two demonstration scenarios in somewhat different, but related, application areas (border crossing and interception of terrorist attacks). In a realistic scenario (long Coronado), we have shown AEC to be more than capable of managing the collaborative process. As directed by the program manager, we have provided our software to the RDEC where it has been tested and has been demonstrated to members of the intelligence community.

We were responsive to the directives of the program manager and the personnel at Rome Labs. These included working with the subject matter experts at ISAI, and providing an interface to the VBCS (which was to serve as the analyst desktop). The scenarios provided by the subject matter experts served as a good basis for the AEC annual demonstrations. We used a variant of the first scenario they produced, the Coronado scenario, as a basis for our 2005 annual demonstration, and were planning a series of 2006 demonstrations based on the Target Abu Jihad scenario. We also found them to be a good source of expertise for providing the details that we needed to add verisimilitude to our demonstrations, so we could bring the material closer to the actual experiences of the people to whom we were demonstrating it.

Another directive that we received was to keep the architecture open and our integration process loose. Our integration with the VBCS was definitely a loosely-defined interaction, allowing us also the ability (with small modifications) to work with other web servers. We actively demonstrated or tested integrations that involved opening Microsoft Word files, making reminder phone calls, and sending emails. Additionally, we were working on a demonstration within the RDEC that integrated some of the Topsail tools orchestrated under the aegis of an AEC flexible process.

Finally, we were responsive to the directives with respect to VBCS. Telcordia was one of the first real systems in Topsail to follow the directive to provide a user interface through VBCS. Because of the nature of AEC, this meant developing a VBCS user interface. Our team developed a user interface, and provided feedback in the form of a white paper that could be used by other Topsail team members as they developed their VBCS user interfaces. This effort took considerable time, which may in retrospect have been better spent working on demonstration development and marketing.

To sum things up, we regard AEC as a successful project that did not have the opportunity to come to completion. We are, however, grateful for the support of the sponsors for the three years that the project was funded, and are looking for other opportunities to carry the work to completion.

3.2. Problems

Perhaps the key problem that we encountered has been the changing nature and focus of the Topsail program, from our perspective. Topsail began as an ambitious program with a focus on improving collaboration, information sharing and “connecting the dots” among agencies in the intelligence community. Its current focus appears to be more in the area of productivity support and tools to facilitate the work of the intelligence analyst. The AEC project is inherently geared towards structuring collaboration among peers, which was a good match with the initial focus, but not with the current focus, as we perceive things.

The second key problem had to do with the management of the tension between the technical progress of the project and the technical transfer activities throughout the project. This problem when coupled with an uncertain development time frame and funding situation, made it difficult to develop a coherent technical development and transfer plan.

2.1.1 Blue-Sky Research Problems

There were a number of problems that we had in AEC simply because we started out more in the line of blue sky research. We had a big, futuristic vision of how collaboration should work, with many milestones along the path to getting there. We believe that this was consistent with the motivation for the project at its conception.

One issue with this has to do with the general nature of government-funded research and the length of time it takes to make real breakthroughs. Futuristic research is more risky, requires more work to bring the project to fruition, and may be characterized by false starts or setbacks. However, the requirements and support for government-funded research fluctuate over time. A problem that captures the attention in this year’s budget development does not necessarily capture the attention of next year’s budget development. Changes in program or agency personnel require you to spend considerable effort re-justifying your work. This does not provide a stable foundation in which you have the time to put in the necessary thought to be able to make leaps in your research progress.

Secondly is the funding issue. Budgets are front loaded so that funding declines at the end of the project. Also, in the current climate, funding is also predicated on the fact that you are making measurable progress. Thirdly, if there is a risk, the program managers may provide less money until you can prove that what you are doing will be beneficial. These factors provide a disincentive to take any risks. If you take a risky approach, you may hit a setback at a time when your progress is being measured, your funding may decline before you reach a point where you are ready to transition your project out of the research arena, or you may be under funded to the point where you cannot actually complete the research. Thus, it is more appealing to do short-term, predictable, easily measurable research.

A third issue with more blue sky research problems we have already alluded to, and that is that it is hard to explain what you are doing to someone. What you are doing is new, and therefore there is little shared experience on which to base your conversation. You may be able to state very high-level objectives, but often there is no common language or terminology that you can use when you want to go down a couple of levels, where you can really differentiate what you are doing – either that, or, as described before, the use of familiar terms is as likely to mislead as to clarify things.

These things said, not all of them impacted AEC all of the time. The program manager generously funded us for a second year even when our first year products consisted only of a white paper outlining how we thought we could solve the problem, and a proof of concept demonstration. As the program manager caught our vision more in the second year, he increased our funding to allow us to make more substantial progress in the third year. Additionally, the program managers provided us with opportunities to try to describe what we were doing and to talk to analysts and other IC members during the PI meetings, which was a great help in sharpening our terminology and to demonstrate our product. All of these helped smooth our research considerably.

2.1.2 AEC-Related Problems

AEC addressed a wide problem scope and made a large amount of technical progress. A major contributing factor to its termination at the end of three years had to do with the fact that the project was under funded from the start, yet Telcordia was not able to scale back its objectives and efforts in a manner that reflected the funding shortfall. The consequence of this was that the team paid much more attention to technical requirements and to producing technical functionality, and less attention to the equally important issues of usability testing and technology transfer. This was especially bad due to the nature of the project, and the complications and issues discussed in the sections on issues related to collaboration-oriented research and blue-sky research in general.

A second major problem was that it was hard to differentiate AEC from other collaboration products verbally or in writing, though the effects were easier to see when AEC was demonstrated. For example, consider the key concepts of AEC as described – context, policy, process and awareness. All of these terms are already overloaded in general, in the software application world, and in the collaboration and distributed systems worlds. For instance, "context" can mean the variables accessible to a program (as in classic programming languages), the local representation of the work environment in which an application is being run (as in the work on norms and institutions), or sets of documents and emails that logically relate to a specific task (as in Groove and the ISX CAST application). Furthermore, even within AEC, the word "context" can mean either the set of policies, resources, tools and suggested methods associated with a single organization, jurisdiction, etc., or it can be used with respect to an activity to mean the set of policies, resources, etc. that apply to the activity, because they are within some "context" either directly or transitively related to the activity's "context". Unfortunately, Telcordia did not do a particularly good job of disambiguating and defining its uses of these words, with the result that it was easy to misunderstand our descriptions of our system.

This terminology issue was a key factor in our inability to market AEC and to do technology transfer. It was very hard to describe AEC in terms of differentiators from other available products. Other systems do processes, so we needed to describe how our notion of process was more suitable for supporting teams of humans. Other systems provide context, so we needed to describe how our notion of context was intended to provide channels for scoping policy enforcement, resource location, and for determining how to do something – as opposed to other collaboration systems such as those whose notion of context was meant to organize documents and resources according to topic. Other systems provide policies, but they are mainly access control policies and do not deal with how to organize and structure team activities. Similar issues existed for awareness, resources, and other key concepts. This meant that, in order to differentiate AEC from some of these other systems, we needed to provide a deeper explanation of system capabilities than is normally present in a marketing presentation. There was just no shared vocabulary and assumptions from which we could start.

2.1.3 VBCS Issues

The VBCS integration was plagued with problems. Working through these problems became a complex task that took time away from other research activities more directly relevant to the AEC research.

During our VBCS integration, we identified and reported several problems with VBCS, both with respect to its use within AEC and within the Topsail program as a whole.

The first set of problems stemmed from the fact that the VBCS interface is a basic web interface, which means that some functionality of dedicated user interface will be either not available, or severely limited. In particular, the following features have problems when ported from a highly-interactive, dedicated GUI to a web-based user interface:

- Graphical design tools and wizards. The only reasonable way to implement such interactive tools in web-based systems is via applets (or similar downloaded programs). As far as we know, VBCS only supports the capability to provide a link to an applet. There is no mechanism, for instance, for enforcing a consistent approach to applet implementation.
- Integration and local execution of programs and tools on the user's desktop machine. Often, an activity in AEC invokes an external tool, such as Microsoft Word ™, a specialized form-filling program, or a link analysis tool. The tool is invoked on the user’s behalf, and on the user’s machine. Web-based interfaces are, by design, prohibited from such invocation. Standard workarounds (such as downloaded "executor" programs) detract from the usability of an integrated interface. Additionally, they have security issues that may be unacceptable to the intelligence community.
- Callbacks. These are desirable in situations where the user must be informed immediately of significant changes or alerts. A web-based interface relies on a pull mechanism instead of a push, and thus usually depends on either a user refreshing the screen or a timed polling mechanism; alerts may no longer be timely or convenient.
- Cache and performance. There are some concerns that the caching mechanism is not sufficiently tuned to a web portal system, however, its impact is rather limited.

An additional issue, identified at the time of the VBCS integration, is that the VBCS system requires an additional database on which to run, even if you are not using the document repository aspects of the VBCS. This makes it very heavy weight as a user interface mechanism.

One final issue that we identified as we used the VBCS interface in the process of implementing tests and demonstrations is a scalability issue. When, for example, there are a lot of classes in the AEC activity ontology, or many different methods that can be selected for doing a particular intent, it became hard to represent this plethora of choices in a manner that was appropriate for web page display. Rather, we needed to provide some form of ability to browse ontologies and catalogs; a functionality that is impacted by the local execution issues described above.

2.1.4 Collaboration-Oriented Research Problems

In addition to the AEC-related problems described above, there are some problems that characterize projects in collaboration in general. Collaboration is by nature widespread, involving many people. Because of this, you cannot sell a collaboration product to just one person, but instead must convince whole groups of people to try it. This means that, for technology transfer to succeed, you either need to transfer through one high-level manager that is committed to giving it a fair trial, or you need to locate a cohesive group of champions that work together on some common task. If the collaboration is meant to be large-scale or cross-agency, and its capabilities shine best in these areas, then the problem of finding early adapters becomes even more challenging. Contrast this to the task of locating early adapters for a focused tool – in this situation, you really only need to start with locating one person to be trained in it and then try it out. That single person can then adequately advocate the usefulness of your tool and hopefully become a champion for it.

A second issue (related to AEC experimentation in the RDEC) had to do with metrics and measuring how your product improves what it is intended to improve. This type of information is very useful when looking for domains in which you can sell a product, or for statistics that you can use during the actual marketing of the product itself. The first problem in this area is that no two groups work in the same way. This makes experimental design difficult. Therefore it is hard to figure out how to measure progress or ROI for a collaboration tool unless you do a large study involving known sets of groups doing known sets of tasks. It is hard in this setting to determine what ground truth is; nor is it easy to control the experiments in such a way that you can isolate out the information that you want to measure. Few known metrics tend to apply in this type of setting. Therefore, it becomes a challenge to determine how to measure what you would like to measure. This issue became important for us within the RDEC testing, where our system was not included in any experiments.

3.3. Lessons Learned

This section details the lessons that we learned during the three years that we participated in the Topsail program. It covers project planning and management lessons, research lessons, and finally (and probably most importantly) marketing lessons.

2.1.1 Project Planning and Management

We began our work on AEC with a very ambitious agenda. The original award included an appropriate amount of funding over a period of five years. Unfortunately, actual funding we received was insufficient to achieve a reasonably complete set of functionality at the end of the third year, when the project was terminated. In other words, there was a disparity in development resources, and a changing planning horizon; these impacted our ability to plan well. Because of this experience, in the future we will structure our projects and proposals differently. New approaches include:

- Propose projects that are smaller and more focused, or which consist of a set of relatively independent, decomposable parts that are smaller and more focused. This makes it easier to scale the project down when the funding level is less than expected.
- Aim for a prototype at the end of the first year and something well on the way to a COTS product by the end of the second year. This gives you more opportunity to socialize your idea with others and get some good feedback, and to start working on technical transfer early.
- Scale the project to the funding. If the funding is cut in half from what you proposed, then cut the functionality in half as well. Don’t try to squeeze in more, as this will impact later tech transfer and marketing efforts.

Note that scaling a proposed project down from what you envision in the long term is hard, and may be at odds with the ambitions nature of the BAAs. You need to start with some part of the proposed concept that appeals to a number of people. For example, HTML-like systems were first developed to enable organizations to organize documents in house. It was only later that HTML was used more broadly in the Internet, which was the beginning of the World Wide Web.

2.1.2 Research

As mentioned earlier, AEC started out with a broad research agenda, with several thrusts that needed to move forward in parallel in the areas of context, policy representation and validation, process modeling and enactment, and in streaming-like queries for awareness specification and delivery. We proposed, and implemented, a spiraling development model, pushing the research forward on multiple fronts each year. This approach proved in the long run to be a real strength from the prospective of moving the research forward technically, but a handicap with respect to explaining the focus of our research, demonstration, and technology transfer. This also was an issue with respect to metrics, because multiple experiments, each requiring metrics focused on a single technical area, are needed to test such a system. This was quite infeasible with the current RDEC setup. Finally, when the project terminated, we were left with some research threads in an incomplete state.

One possibility to ameliorate this process is to tackle more focused research projects that push forward only one area of research at a time. Instead of using a general spiraling development, we could have pushed one area forward at a time, then tested and demonstrated only that one area. Unfortunately, this approach would have slowed down the AEC research in the long run. Our approach enabled us to avoid rewrites and to converge on a good answer on the first try, Taking smaller, focused steps would probably have engendered rewrites as we would not have been able to think through the broad implications of our design.

2.1.3 Marketing

Marketing an intelligence-related product is hard, especially when you have little or no experience in this area. Our major lesson learned with respect to the AEC project and marketing is that we need partnerships with multiple individuals that have experience in this area to partner with us in marketing and in the development of our marketing message. These partners also need connectivity into the IC to help supply contacts within agencies, and to facilitate meetings with these contacts. Finally, with respect to a product that potentially facilitates integration; the partners need to provide us with information on what products any potential customer is actually using, and on what problems the potential customers are encountering with their current suite of technical tools and products.

It really helps to have a focused purpose and message to the customer, suitable for marketing, at the proposal stage. This message should be guided with input from individuals experienced with the targeted customers. AEC did not have a focused purpose and message – it was (at different times) billed as a collaboration platform, an integration platform, an enforcer of policies within processes, etc. This lack of marketing message was complicated by the fact that a culture of collaboration was not inherently present within the intelligence community – this was an identified problem at the beginning of the Topsail program. Thus, the marketing of a collaboration product such as AEC runs against agency culture. Rather than giving a broad message, we could have focused on just one good differentiator of AEC (such as policy), and used that as a consistent marketing message to the IC. We could have developed marketing material early, start selling our ideas early, and follow up aggressively on all marketing contacts. This would have given us more contact within the IC, and enabled us to get feedback from as many potential users as possible. However, the downside of this is that an early focus on one or two customers may have caused us to narrow our focus and hampered our ability to capture the bigger picture.

Finally, with respect to the facilitation of the technology transfer process, it is useful to aim for a COTS-like product within three years. This may be a subset of what the entire project might intend to do over its lifetime. The availability of a COTS-like product would enable us to marked something concrete at an early stage, would give something to deliver at an early stage, would give us a toehold into the community with an existing product, and would provide lots of opportunity for feedback and guidance as the product continues to be developed.

4. Conclusion

Awareness-Enabled Coordination was an ambitions project that made significant research progress and had a number of technical breakthroughs. These included:

- A context network that allows independent entities to manage their context information autonomously, but then make it useful to the community as a whole. This is one step towards a more global notion of context.
- The ability to dynamically contextualize activities so that they have the resources and procedures, and are affected by the policies, that are appropriate to the organizations, jurisdictions, and teams that have current authority over them.
- The ability for a team member to incrementally refine an activity, starting as early as only knowing what he intends to do, and adding detail as he or she determines what is needed and what policies need to be followed.
- The ability to begin execution of earlier activities of a process even when later activities in the process remain unspecified or incompletely specified.
- The ability to represent and enforce policies that constrain how different aspects of an activity must be executed, while not forcing fixed and specific compliant processes to be the ones that are used.
- The ability to provide a team or team member with awareness information that may impact the work of the team or the team member, in a targeted manner.

On the issues side, the major issues that affected the Awareness-Enabled Coordination project related to funding. Our response to this was to try to focus on the technical development side. As a result, we may not have paid adequate attention to issues of testing, metrics, and technology transfer.

5. References

[ACC+03] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. "Aurora: A New Model and Architecture for Data Stream Management". *VLDB Journal*, 12(2), August 2003.

[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", *Int'l Journal of Cooperative Information Systems*, March 2002.

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

[BPEL] "Business Process Execution Language for Web Services Version 1.1". http://www-128.ibm.com/developerworks/library/specification/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". *Proc. Workshop on Policies for Distributed Systems and Networks*, Springer LNCS, 2001.

[Di02] F. Dignum. "Abstract Norms and Electronic Institutions". *Proc. 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.

[FIPA] "Foundation for Intelligent Physical Agents". http://www.fipa.org/.

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

[GSC00] D. Georgakopoulos, H. Schuster, D. Baker, and A. Cichocki. "Managing Escalation of Collaboration Processes in Crisis Mitigation Situations". *Proc. 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". *Proc. Int'l Joint Conference on Work Activities Coordination and Collaboration (WACC'99)*, 1999.

[InfoSleuth] "The InfoSleuth Agent System". www.argreenhouse.com/infosleuth/ .

[JBI] “JBI – Joint Battlespace Infosphere”. http://www.rl.af.mil/programs/jbi/ .

[Jena] “Jena Semantic Web Framework”. http://jena.sourceforge.net/.

[Kag+03] Lalana Kagal *et al*. "A Policy Based Approach to Security for the Semantic Web", *Proc. 2nd International Semantic Web Conference (ISWC2003)*, 2003.

[NFK00] M. Nodine, J. Fowler, T. Ksiezyk, B. Perry, M. Taylor, A. Unruh. “Active Information Gathering in InfoSleuth” *Int’l Journal of Cooperative Information Systems,* Vol. 9, No. 1, 2000,

[OWL] “OWL Web Ontology Language Overview”. http://www.w3.org/TR/owl-features/ . 2006.

[Protégé] “The Protégé Ontology Editor and Knowledge Acquisition System”. http://www.protege.stanford.edu. 2006.

[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”. *Proc. Workshop on Agent Organizations: Theory and Practice*, AAAI Press, 2004.

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

[STREAM] Stanford University. “STREAM: The Stanford Stream Data Manager. http://www-db.stanford.edu/stream/. 2006.

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

[VEAS] D. Georgakopoulos and D. Baker. “Providing Physical Security via Video Event Awareness”. Whitepaper on file at Austin Research Center, Telcordia Technologies, Austin, TX. 2006.

[Vignette] 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”. *Proc. 31st Hawaii Int. Conf. on System Sciences,* Software Technology Track , 1998.