Using Δ-Federations for On-Demand Crisis Situation Response

R. Baird and R. Gamble
Software Engineering and Architecture Team
Department of Computer Science
University of Tulsa
800 S. Tucker Drive
Tulsa, OK 74104
robert-baird@utulsa.edu

Abstract

Dynamic, service-based systems offer powerful resources for knowledge and information processing in a crisis situation. These systems need efficient strategies to integrate diverse services under the stress of an emergency to capture, organize, and disseminate appropriate information. These strategies require a combination of service discovery, 'mashing', and orchestration given workflow task templates that can be instantiated and deployed on demand. In this paper, we introduce the Δ-Federation reference architecture for on-demand emergency management systems (ODEMS). The Δ-Federation provides the foundation for incorporating these strategies that consider type, availability, and requirements of services as historically understood by users in similar crisis events.

1. Introduction

Crises, caused by human and natural disasters, require immediate and ongoing information processing and dissemination to affected communities. The development and proliferation of web applications has yielded numerous online services capable of gathering, organizing, and deriving the relevant information needed by the public. Unfortunately, dynamically locating and orchestrating these services to complete useful tasks, such as forming evacuation plans, determining shelter availability, and crew allocation, and announcing the arrival of food, water, and other necessities. Thus, crises require a time-sensitive infrastructure that composes a wealth of services, some of which perform redundant tasks, and can manipulate them on demand according to prescribed plans.

Modeling tasks using workflows or plans provides formulaic, programmed responses to specific crisis events. These plans can be acquired by historical needs and previous forms of community involvement. This concept is similar to preparation for a crisis. If an ice storm is predicted, both salt trucks, tree trimming trucks, and utility trucks are readied with their crews. These anticipated workflows must rely on emergency, industrial, weather, radio, and utility services, as well as information from social networking and unexpected sources of needed information but whose service availability may not be predictable.

The infrastructure in which the workflows reside must account for when providers may update services and their availability as well as the contracts they operate under given the emergency. Additionally, the changing crisis scenario requires continued adjustments in acquiring and disseminating information. Thus, patterns of workflows are needed to address the changing situation and update executing tasks to reflect ongoing needs. This influx requires an ability to perform these same tasks redundantly. That is, there must exist multiple workflows resolving the same or closely related goals with higher/lower priority services to allow for shifts in both service availability and in dictated user needs.

In an ideal world all dynamic events would feed through a human-in-the-loop to guide decision-making. However, many systems designed for emergency management do not crosscut all the levels of the system information that must be analyzed. In this paper, we introduce the Δ-Federation reference architecture for on-demand emergency management systems (ODEMS). We assume the existence of two important entity types: user needs for delivering and receiving relevant information given certain crisis situations such as earthquakes, snow/ice storms, and chemical spills, and abstract workflow patterns that are defined by these historical needs to guide the decision process among available service use, invoke those services according to the patterned definition of the task, and dynamically alter the services as dictated the changing scenario.

The Δ-Federation reference architecture represents an amalgamation of (1) services that can be used for most crises, like shelters, and those that are dedicated to specific events, like salt trucks, (2) workflow patterns that can be augmented and instantiated depending on the event, and multiple
strategies for coping with dynamism in service type and availability and changing community needs.

2. IT Capabilities in Emergency Management

Crisis management relies on multiple heterogeneous types of communication infrastructures, including sensor grids, voice communication, data streams, video conferencing, and, more recently, social networking. Services currently fall into three main categories: long-standing stable services, new services, and user generated mash-ups. According to a recent National Academy of Sciences publication, research indicates that, in a disaster situation, Information Technology (IT) has the capabilities to aid people in clearly understanding the situation and their roles so that they make quick decisions based on accurate information [1]. Two of the cited IT capabilities are

- Improved decision support and resource management, including online directories and collaborative software and file sharing
- Improved public engaging, including multimodal reporting and resource contact systems.

Web services serve as a starting point for building the necessary infrastructure to address these challenges. Integrating components, interactive systems, and dynamic user environments on demand within a collaborative forum is plagued with orchestration and quality of service problems, as well as difficulties with addressing the changing needs of users in a disaster situation. Solutions must encompass concerns for (1) functional data and control exchange, (2) non-functional performance, reliability, and security, (3) user QoS priorities and preferences, and (4) situational awareness.

Recent advances in Web 2.0 technologies, cloud computing, and web application mashups result in a shift from traditional development cycles for integrated systems to what is being called “Dev 2.0” [2]. In Dev 2.0, new technologies are making services available to users as the software evolves, even enabling users to participate in formulating the desired task and fulfillment technologies. Dev 2.0 can use workflows to combine procedural rules with specific services to facilitate the transmission of tasks and information among participants.

Several examples of Dev 2.0 for disaster situations exist. In 2007, the Amazon.com computational infrastructure for Human Mechanical Turk nearly instantaneously searched satellite imagery for the crashed airplane of Steve Fossett [3]. The approach combined web services and satellite imagery to examine photographs searching for a crashed airplane. Another Dev 2.0 crisis example is the 2009 flooding of the Red River in Fargo, ND. A web presence assisted in disseminating critical emergency information. Water levels of the Red River were published hourly to Twitter [4] enabling community members to view the information asynchronously and evacuate accordingly.

Though Web 2.0 is a worthwhile development tactic, not all interoperability and dynamism issues are easily resolved. With more complex development where workflows form the backbone behind opportunistic mashups, failures in availability and communication require a decentralized approach for their management [5]. Unfortunately, few approaches tackle service failure with a comprehensive understanding of workflow management. Even though mashups are popular, they lack the sophistication to provide crucial, on-demand services integration [6]. Still, IT and the recent rise in WS popularity should conspire to meet the demand of emergency management situations. Traditional service-oriented architectures cannot provide needed the infrastructure because they are static, do not have adaptive workflow templates that provide for redundant task executions in a dynamic environment, and cannot orchestrate and ‘mash’ services on demand.

3. The Δ-Federation Architecture

Imagine a crisis situation centering on a chemical spill and subsequent fire. Historically, this scenario often requires dynamic plume modeling. Our example uses geographic information system (GIS) applications for mapping within a cordoned area. The ODEMS must integrate mapping services with chemical databases and with services that provide current weather information. In this scenario, community representatives initially formulate workflows to track the spread of chemicals.

A successful relationship between IT and crisis situations must overcome several challenges rooted in the core instantiation of the ODEMS from the Δ-Federation. Traditionally, these systems operate in closed networks using intra-organizational communication, including phone and liaison workers, that an appropriate level of pre-planning can accommodate [7]. The Δ-Federation allows addressing these challenges and others that the community encounters during runtime when it needs inter-organizational communication by:
allowing community representatives to direct information processing and dissemination by changing existing, preplanned workflows,

• separating community involvement from workflow execution, while allowing workflow refinement given task results and input from affected people,

• offering redundant service access to allow community representatives to prioritize alternatives, even as they become aware of them, to better align the processing with community needs

• allocating service providers a dynamic means to upgrade and deploy additional services for community use

• promoting dynamic changes while minimizing downtime.

Shown in Figure 1, each layer of the Δ-Federation aggregates information to assist in workflow instantiation, deployment, and reconfiguration. The architecture facilitates dissemination to social networking sites by integrating with the various web architecture at each layer are tasked with the responsibility of data aggregation to uniformly provide information and processing that drives management the reference architecture includes a traversal process that crosscuts the layers of information to isolate and track the current state of community needs.

![Traversal Process](image)

**Figure 1. Δ-Federation Reference Architecture**

Inside the Δ-Federation, the Users layer collects community service preferences, historical information needs, potential workflow tasks given expected service offerings, and perspectives for workflow results. The users of the Δ-Federation include both developers of the ODEMS as well as users of social networking sites. Concerns specify the preferences that the Δ-Federation has over task composition, execution, and updates based on the community it serves. An ontology (i.e., hierarchy of connected terms) unifies concerns expressed by different communities to semantically reason about the needs for tasks and workflows. Providers aggregate service interface definitions and context information from the differing provider access technologies (WSDL, job schedulers, and REST web services, for example). Contracts describe the policies and requirements that each community associates with usable services. Contracts include operational and security requirements of the federation, as well as any Service Level Agreements (SLAs) or Quality of Service (QoS) specifications that exist. A final Connectors layer provides a toolkit of software integration enablers that resolve common interoperability problems during integration and execution.

Isolating each layer of the Δ-Federation furnishes the ODEMS with a mechanism to monitor multiple sources of dynamism separately. Assigning components to each layer enables the community to adjust the ODEMS monitoring on a situation-by-situation basis. Reporting of events must be passed to a Traversal Process that segregates the management of the different types of dynamism.

The Δ-Federation handles considerations for both data and control exchange since task execution requires that all entities collaborate fluidly without interoperability problems. When a chemical fire occurs, emergency response teams immediately seek feedback on plume contents, location and direction [8]. Though this feedback generally comes from a stable set of services, ODEMS frequently experience difficulty in automatically accessing and orchestrating them. Additionally, the number and type of alternative resources, including user provided information, is initially unknown. Finally, new services and information resources, such as transportation for evacuation, arise that may be crucial to community needs.

The Traversal Process (shown vertically in Figure 1) combines context-awareness and dynamic resource management within the Δ-Federation. These core modeling constructs weave the Δ-Federation layers together to achieve the processing. A Task phase starts the traversal process and derives the high-level functional properties of execution including access rights, invocation limits, and connectivity between ODEMS users. Primary tasks in our example would
be to investigate the fire and direct search and rescue efforts, to model a chemical plume based on weather information, and to deliver evacuation notices as the situation demands it.

Composition prepares the arrangement of services and service compositions (workflows); including redundant compositions should certain services become unavailable. The reference architecture contains well-known solutions to ODEMS situations for which abstract workflow types are available to prepare, select, and adapt to particular tasking needs. Service selection uses the compositions and community policies to orchestrate a specific workflow for execution. This phase takes the abstract workflow and generates a concrete workflow invoking services such as the National Weather Service for current wind speed. The Contracts layer provides context information, including service availability, access rights, and current response times. The weather service can impose restrictions on the number of invocations per hour. Thus, a context manager ensures that community concerns and acceptable metrics are accounted for. To formulate executable workflows, a Link construct obtains the functional requirements regarding the service interfaces. The Connectors layer supplies the link construct with resolution strategies, inserting routers, controllers, and translators into the workflow. Execution and Reconfiguration are performed by a reconfiguration management system. The system integrates a user support for changing goals and preferences of the community users.

The Δ-Federation consolidates the necessary entities to instantiate services and workflows to create an ODEMS. Its feasibility stems from the available technology of workflow patterns, service QoS from which priorities can be derived, Web services communication standards and two implemented technologies: Contract Templates and a toolkit that performs dynamic Workflow Reconfiguration.

4. Integration for Web Services

To incorporate heterogeneous service types within the Δ-Federation given the ODEMS domain, we extend our prior XML-based service specification Contract Templates [9] to be Crisis Service Level Agreements (CSLAs). Once instantiated, CSLAs provide the ODEMS with accurate service provider use requirements to resolve functional and semantic interoperability problems. The profile captures resource information, including current state, attributes, and functions, as well as its metadata to adapt it within a particular context (i.e., crisis, goal, etc.). CSLAs assist with syntactic and semantic unification of user directives by providing a standardized mechanism to collect heterogeneous systems within the ODEMS domain. Three distinct levels separate each contract, as shown in Figure 2, with increasing association from the Δ-Federation to the service provider. The levels are: (1) the community’s predefined semantics, (2) expected user directives, and (3) CSLA capabilities.

The first two levels of the contract are instantiated prior to the Traversal Process in order for service selection, integration, and execution to occur. The first level contains semantic information, using the ontology for that crisis situation, describing the properties of services, such as component type, functions, or methods they expose. The second level includes user directives, parameter values or command line arguments that control the service execution. Service providers must expose this information for inclusion into the system.

Earlier versions of the Contract Template tied direct invocation details to a third level within the XML structure. The final level incorporated by the CSLA directly encodes SLA details linking the service provider with the contracts it agrees to operate under. The Traversal Process automatically fills this level during the Link phase maintaining current up-to-date information, including if the service is operating according to the agreements it publishes.

To generate CSLAs for the chemical plume example, we identify four pieces of semantic information an ODEMS representative would require in order to quickly classify a given service: Service Type, Data Type, Access Type, and Encoding Type. This initial, partial specification does not represent the full CSLA, relying on an automated refinement process to complete the specification. As the crisis unfolds, the ODEMS community will require current WS QoS properties, (e.g. reliability and response time), and the immediate integration with new services.
service types (e.g. an alert distribution system).

Within the CSLA the value for service type broadly categorizes the functionality a service offers. For the example, expected values include services that provide weather information, plume mapping toolkits, and databases of chemical information. Data type similarly describes the type of data that a service exposes or expects to interact with. This information reduces the miscommunication of information exchanged, which often leads to errors in decision-making. For example, the difference between a service delivering wind speed information in miles per hour (mph) versus (kilometers per hour) can cause a miscalculation of a chemical plume model. The use of the CSLA resolves this issue.

Services often require different access technologies, restricting the use within an ODEMS. Access type refers to the mechanism with which the workflow must interact in order to invoke a service. For example, web services can expose functionality using WSDL documents, or they can function as REST-based services. Access control technologies including passwords or key-based tokens can restrict sensitive information. Encoding type refers to the data encoding mechanism the provider accepts at its interface. Data may have a similar type (weather information wind speed, for example) but the mechanism the service uses to encode it can differ between implementations. Valid encoding types can include RSS XML feeds, comma separated files (CSV), PDF documents, or HTML for example.

5. Reconfiguration inside the Λ-Federation

The Execution and Reconfiguration portion of the Traversal Process in the Λ-Federation supports dynamic runtime changes to workflows due to updates in services and community goals. This construct reuses the workflow reconfiguration portion of our fully implemented Next generation Workflow Toolkit (NeWT) [10]. Figure 3 overviews the processing functions NeWT such that it can work with the multiple, redundant workflows generated and stored by the workflow task patterns. As a two-part extension to a traditional workflow management system, NeWT adds functionality for definition and instance reconfiguration with changing goals and service preferences. It accepts that multiple, competing Web services will be available and should be interchanged as dictated by need. As such NeWT employs the Contract Template specifications to fully support the needs of Λ-Federation participants.

Service failure in an ODEMS can occur for a variety of reasons, and emergency situations may not always yield services that present availability with a clear-cut Boolean value. For example, a chemical sensor may be slow to respond to a change in air quality, yet still appear to provide correct information. Thus, the processing time for service requests may vary depending on system load, the time of day, or number users active on the system. During an emergency, communications may be lost at any time due to network, power, and demand issues.

Users, historical information, and current service providers denote which services have priority over their redundant counterparts in which workflow tasks. By allowing this allocation of priorities among similar service types, the Λ-Federation, with the help of the NeWT technology can house multiple workflows for the same task. Thus, when preferences or service availability changes (when the crisis forces a primary weather service offline), NeWT reconfigures all executing instances of currently deployed workflows, such as those using the primary weather services. The reconfiguration brings in alternate services predetermined for replacement in the case of unavailability, allowing tasks to immediately complete with the new services in place. Critical information processing is, therefore, not interrupted. NeWT reports to an ODEMs administrator workflow usage, services QoS, status of running instances, information about the community WSs, and information about reconfiguration occurrences. This feedback mechanism increases situational awareness for making alterations in tasks and goals, as well as provides additional historical information for use in the next crisis.

Inside the Λ-Federation, the CSLAs and Execution and Reconfiguration as performed by NeWT can be directly applied to the chemical plume crisis. The ODEMS example relies on the following services that fulfill the emergency situation requirements: Warehouse Inventory, Weather Service (primary), Chemical Simulator, and a Weather Service
(backup), which is an available, alternative service with the same functions as the primary. Table 1 presents a sampling of the CSLAs service providers instantiate for participation in the ODEMS. Differences between access mechanisms and encoding types are present to demonstrate the variety in service types available in web service architectures. Although a real workflow to perform the necessary tasks during an emergency requires several additional services, this basic set allows us to investigate a critical workflow failure and reconfiguration.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Service Type</th>
<th>Data Type</th>
<th>Access Mechanism</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warehouse Inventory</td>
<td>Database</td>
<td>Chemical Properties</td>
<td>SQL</td>
<td>CVS text</td>
</tr>
<tr>
<td>Chemical Simulator</td>
<td>Planning Tool</td>
<td>Plume Map</td>
<td>WS-wrapped legacy system</td>
<td>PDF Image</td>
</tr>
<tr>
<td>Weather Service (primary)</td>
<td>Weather Information</td>
<td>Wind Speed (mph)</td>
<td>http website</td>
<td>XML encoding</td>
</tr>
<tr>
<td>Weather Service (backup)</td>
<td>Weather Information</td>
<td>Wind Speed (kph)</td>
<td>Twitter feed</td>
<td>CVS text</td>
</tr>
</tbody>
</table>

For Composition in the Traversal Process, the community interfaces with NeWT to generate a workflow specification using these services. Using the workflow formulation functionality and the Service Selection construct, community representatives generate a workflow specification and their priorities for alternative service use to invoke services that construct a chemical plume map. The workflow gathers data about chemicals from the warehouse, collects current weather information, and generates a plume using the chemical simulator service. The workflow is displayed graphically on the left-hand side of Figure 4.

![Figure 4. Chemical Plume Workflow](image)

After specification the community uses the execution functionality of NeWT to launch instances of an associated workflow specified via the Business Process Execution Language (BPEL). As each workflow executes, the workflow engine invokes the individual services to perform their tasks. Assume that the primary service for weather information goes offline. The service failure could stem from lost power, service overload, or a software update. The ODEMS must respond to the failure to maintain community trust. Though continual instance and state inspection, NeWT becomes aware of the failure and provides feedback as to the current problem.

By pre-storing redundant service connections and the user preferences on the order of priority for invocation, NeWT automatically generates an alternate workflow specification (shown in the right-hand side of Figure 4) when the service failure occurs. NeWT manages the reconfiguration of workflows and migration of instances according to their current state of execution.

The Δ-Federation Link construct of the Traversal Process inserts the correct integration enablers when CSLA information indicates it is necessary. For example, the backup weather service has different data types and encoding mechanisms to store current weather information. Thus, the Link addresses this by inserting and instantiating the proper translator for seamless execution of the service within the workflow. Finally, the Execution and Reconfiguration construct of the Traversal Process works with NeWT to complete the reconfiguration management that includes reissuing BPEL partner links establishing proper workflow orchestration. Migrated instances then continue their execution to provide the most accurate information in the current scenario.

During execution representatives routinely receive information from the affected community, by way of system usage, email, and social networking sites, that direct a shift in workflow processing as reflected in ongoing situational awareness. For instance, querying fewer sensors can mean a faster workflow response time, but less accuracy. However, at a certain point, accuracy may be less important if the public must quickly evacuate. Different software vendors can offer separate plume modeling services that range in computational complexity. These services may be chosen based on priority, availability, or weather conditions. After instituting the change the community representatives can review the workflow feedback in conjunction with the user feedback in the form of concerns to ensure the ODEMS is in a proper state to continue to address the crisis event.

6. Relevant Research

A variety of commercial and research toolkits exist for first responders and emergency operations centers. Frameworks for EMS focus on tasks related to surveillance and monitoring environment properties, the evaluation of risk in the community, and contingency planning for when primary planning is insufficient [11]. Decision support and knowledge management feature prominently with little concern.
for integration and change management. Ad hoc wireless sensor networks interface with existing middleware technologies to resolve networking issues [12], however current approaches lack support for the adjustment of goals or service availability.

The Open Platform for Emergency Networks (OPEN) [13] is a standards-based approach to establish software interoperability in crisis situations. Its benefits are limited due to lack of adoption from local communities and service providers. The Web Service Federation Language [14] created to assist with brokering security tokens across federation boundaries. WS-Federation facilitates the movement of identities, attributes, and authentication between realms, but does not specify non-security concepts. Traditional research in federation engineering focuses on the aspect of integrated services performing in adherence with SLAs as contracts for expected performance [15]. Automated reconfigurations based on SLA violations are only possible given predefined (static) alternative workflows. The wider range of community input is not part of the framework.

7. Conclusion

The Δ-Federation reference architecture demonstrates the potential for rapid configuration of an ODEMS, working with collaborative information, fault-tolerance, and adaptability. Workflow reconfiguration adapts tasks dynamically to community needs, service provider availability, and integrates with CSLAs to rapidly changing crisis events. By blending the different sources of information from the layered architecture, the traversal process continuously updates workflows while simultaneously integrating new services as their specifications enter the Δ-Federation. Developing and expanding the Δ-Federation architecture can assist in domains other than emergency management with similar success. Ultimately we envision the approach can be applied to emergent SOA applications such as cloud computing, where contracts and availability play key roles in service provisioning.

Acknowledgement. This material is based on research sponsored in part by the Air Force Office of Scientific Research (AFOSR) under agreement number FA-9550-09-1-0409 and in part by the Air Force Research Laboratory (AFRL) under agreement number FA8750-10-2-0143. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied of the AFOSR, AFRL, or the U.S. Government.

8. References