

UNCLASSIFIED

**QLI-TR-07-001**

**Integrated Biological Warfare  
Technology Platform (IBWTP)**

**Intelligent Software Supporting  
Situation Awareness, Response, and Operations**

**Final Report**

Contract N00014-02-C-0320

Jan 2007

**Quantum Leap Innovations, Inc.  
Delaware Technology Park  
3 Innovation Way  
Newark, DE 19711**

UNCLASSIFIED

This page intentionally left blank.

**Quantum Leap Innovations, Inc.  
Delaware Technology Park  
3 Innovation Way  
Newark, DE 19711**

---

---

**QLI-TR-07-001  
Jan 2007**

# **Integrated Biological Warfare Technology Platform (IBWTP)**

**Intelligent Software Supporting  
Situation Awareness, Response, and Operations**

## **Final Report**

**ONR Contract N00014-02-C-0320**

---

---

**Sponsored by: Office of Naval Research  
Arlington, VA 22217**

# Abstract

Within the context of the Integrated Biological Warfare Technology Platform (IBWTP) program, Quantum Leap Innovations, Inc. (QLI) was tasked by the Office of Naval Research to develop, evaluate, and demonstrate novel technology supporting early detection of and rapid response to biological or chemical threats. This report provides an overview of the challenges QLI faced, the approach it took to creating the technologies, and some of the specific technological solutions in the areas of Situational Awareness, Course of Action Planning, Command & Control, and Data & Process Integration. It also presents the applicability of the developed technologies to areas other than biological response, such as Department of Homeland Security applications in emergency management, and Department of Defense applications in force transformation, especially regarding Future Naval Capability (FNC) Knowledge Superiority and Assurance (KSA).

# Contents

Abstract .....	ii
Contents .....	iii
List of Figures .....	vi
List of Tables .....	vii
1. Summary .....	1
2. Introduction.....	3
3. Synopsis .....	4
3.1. Problem.....	4
3.2. Approach.....	6
3.3. Solution.....	9
3.3.1. Awareness Capabilities .....	9
3.3.2. Action Capabilities .....	10
3.3.3. Control Capabilities.....	10
3.3.4. Integration Capabilities .....	11
3.4. Applicability .....	11
4. Problem Details.....	12
4.1. Motivation.....	12
4.1.1. The Emerging Threat.....	12
4.1.2. Potential Consequences .....	13
4.1.3. What Is Required.....	13
4.2. Background.....	14
4.2.1. Biological Threats .....	14
4.2.2. National Public Health Surveillance .....	15
4.2.3. Users of Systems for Combating Biological Warfare .....	16
4.2.4. Data Sources .....	17
4.2.5. Data Acquisition and Provisioning.....	18
4.2.6. Analytic Methods and Models .....	18
4.2.7. Planning Approaches.....	20
4.2.8. Interfaces for Collaborative Analysis and Decision Support .....	20
4.2.9. The Integrating Platform .....	20
4.3. Relevant Technologies.....	21
4.3.1. Existing Biosurveillance Systems .....	22
4.3.2. Automatic Integration and Analysis of Data from Multiple Sources.....	23
4.3.3. Continuous Planning and Optimization Models .....	23
5. Methods, Assumptions, and Procedures .....	24
5.1. Integrated Awareness/Action/Control Framework.....	24
5.1.1. Awareness .....	27
5.1.2. Action .....	28
5.1.3. Control.....	29
5.1.4. Integration .....	30

5.2. Evolution of IBWTP Conceptual Framework .....	30
5.3. Software Development Processes and Quality Assurance .....	31
5.3.1. Technology Evolution .....	31
5.3.2. Maturity Levels .....	33
6. Results and Discussion .....	37
6.1. Awareness Capabilities.....	39
6.1.1. Intelligent Data Management (IDM).....	40
6.1.2. Weather Tool Demonstrator .....	44
6.1.3. Dispersion Model Demonstrator .....	46
6.1.4. Environmental Quality Monitoring (EQM) Demonstrator .....	47
6.1.5. Event Attendance Estimator (EAE) Demonstrator .....	47
6.1.6. Dynamic Population Distribution (DPD) Demonstrator .....	48
6.1.7. Probabilistic Reasoning Toolkit (PRT) .....	49
6.1.8. Causal Reasoning Engine (CRE) .....	49
6.1.9. Threat Assessment Module (TAM) Demonstrator.....	50
6.1.10. Dynamic Distributed Data Mining (D3M).....	51
6.1.11. Other Awareness Capabilities .....	52
6.2. Action Capabilities .....	53
6.2.1. Adaptive Optimization® Engine (AOE) .....	53
6.2.2. AOE Utilities and Subpackages .....	53
6.2.3. Resource Placement Framework (RPF) .....	54
6.2.4. Sensor Location Model Demonstrator .....	54
6.2.5. Real-Time Adaptive Planning Framework (RAP) .....	55
6.2.6. Truck Routing Demonstrator.....	57
6.2.7. Distributed Process Execution (DPE) .....	59
6.2.8. Agentacalc Demonstrator .....	60
6.3. Control Capabilities .....	60
6.3.1. Integrated Knowledge Environment (iKE) .....	61
6.3.2. Collman .....	63
6.3.3. Distributed Application Framework (DAF).....	66
6.3.4. Clearboard .....	67
6.3.5. A/V Conference.....	68
6.3.6. Multi-Mouse.....	68
6.3.7. User Modeling.....	69
6.3.8. Menu Manager .....	69
6.3.9. Interactive Knowledge Wall (IKW).....	69
6.3.10. Portable Control Units (PCU) .....	71
6.3.11. Other Control Capabilities.....	73
6.4. Integration Capabilities.....	73
6.4.1. Multi-Agent Development Environment (MADE) .....	73
6.4.2. Policy Management.....	74
6.4.3. Group Communication Service .....	74
6.4.4. Multi-Agent System Management System (MMS).....	75
6.4.5. Agent Glossary .....	75
6.4.6. Other Integration Capabilities .....	75
7. Conclusions.....	77

7.1. Applicability .....	77
7.1.1. BioDefense and Emergency Response.....	77
7.1.2. FORCEnet .....	78
7.2. Publications.....	80
7.2.1. Patent Applications.....	80
7.2.2. Publications .....	81
7.2.3. Technical Reports.....	82
8. References.....	83
9. Index .....	85
Report Documentation Page .....	86

# List of Figures

Figure 1: The Initial IBWTP Approach by QLI .....	7
Figure 2: The Integrated Awareness/Action/Control (AACI) Framework .....	8
Figure 3: Biosecurity Landscape .....	17
Figure 4: The Initial IBWTP Approach by QLI .....	25
Figure 5: The Awareness/Action/Control/Integration Framework.....	26
Figure 6: The Awareness/Action/Control/Integration Technologies.....	26
Figure 7: Evolution of the AACI Conceptual Framework.....	30
Figure 8: Spiral Technology Development from Concept to Deployment.....	31
Figure 9: Evolution of IBWTP Technology Solutions .....	37
Figure 10: Evolution of IBWTP Technology Solutions (cont.).....	38
Figure 11: Data Management Services provided in IDM Framework .....	41
Figure 12: Dataflow among IDM Components .....	42
Figure 13: Different Weather Data Sources.....	44
Figure 14: User's Data Query Interface to Weather Tool.....	45
Figure 15: Application Interface to Weather Tool.....	45
Figure 16: User Input to Dispersion Model .....	46
Figure 17: Application Output from Dispersion Model to iKE.....	47
Figure 18: Dynamic Population Distribution System Overview .....	48
Figure 19: Syndromic Surveillance with the Threat Assessment Module.....	51
Figure 20: User Input to Sensor Location Model .....	55
Figure 21: Output of Sensor Location Model .....	55
Figure 22: General Distributed Realtime Adaptive Planning Architecture .....	56
Figure 23: Distributed RAP Architecture applied to a Damage Control System .....	57
Figure 24: Planned Truck Route with Contingencies .....	58
Figure 25: New Truck Route .....	58
Figure 26: DPE System Architecture Diagram.....	59
Figure 27: DPE Agent Architecture.....	60
Figure 28: Composeable Applications within iKE .....	62
Figure 29: Output of Dispersion and Sensor Location Models .....	63
Figure 30: Socket Connections in the Collman Workbench.....	64
Figure 31: Agent Network Example in the Collman Workbench.....	65
Figure 32: AclOwlGrounding Diagram .....	67
Figure 33: Overview of the Interactive Knowledge Wall Setup.....	70
Figure 34: The Fixed Interactive Knowledge Wall. ....	71
Figure 35: Conceptual deployment of Portable Control Units .....	72
Figure 36: Portable Control Units displaying iKE.....	72
Figure 37: MMS Screen Shot 1 .....	76
Figure 38: MMS Screen Shot 2 .....	76



## List of Tables

Table 1: Goals for Software Maturity .....	33
Table 2: Requirements for SW to achieve Quality Maturity Levels .....	36
Table 3: Filed Patents.....	80
Table 4: Technical Reports .....	82

# 1. Summary

A global threat to the people of the United States has recently emerged, with implications comparable to the nuclear devastation we faced in the cold war era. A combination of easily accessible pathogens, low cost of development and dispersal, and the demonstrated strategy of terrorist adversaries to target civilian and commercial interests, indicates that we should prepare for biological attacks. Though we cannot eliminate the threat of a biological terrorist attack, we can vastly improve the outcome for the US population via a combination of early detection of threats and rapid response to mitigate the damage.

Within the Integrated Biological Warfare Technology Platform (IBWTP) program, Quantum Leap Innovations, Inc. (QLI) developed an integrated decision support framework for defense against chemical and biological warfare. The framework enables the integration of static and dynamic data (e.g. from hospitals, sensors, open source, intelligence), models (e.g. dispersion, exposure, damage), and advanced intelligent computing technologies to create a powerful early detection and rapid response system to identify and respond to potential or existing biological or chemical threats, either man-made or natural. It enables crisis managers to:

- **Monitor** chemical or biological outbreaks,
- **Identify** the cause(s) of the outbreak and its (their) possible sources,
- **Predict** potential exposure,
- **Plan** for effective response, and
- **Alert** appropriate authorities to mitigate the damage (hospitals, local government, law enforcement, military and CDC).

The system supports a variety of potential scenarios and continuously updates real world action plans in a format that improves the efficiency and quality of collaborative decision-making. Collaborative emergency planning and management is facilitated through an interactive knowledge visualization and decision making environment that supports teams of different users (ranging from technical specialists to high-level decision makers) in a single space or distributed across different geographic locations.

The technologies that QLI has developed, also referred to during the project as the Integrated Biological and Chemical Warfare Defense (IBCWD) software, revolve around a comprehensive architectural framework supporting:

- **Situational Awareness:** Provide situational awareness by transforming dynamic, distributed, and heterogeneous data into actionable knowledge in order to identify and localize potential or existing problems and threats as early as possible. Share this knowledge with relevant users and applications as soon as possible.
- **Course of Action Planning, Optimization, and Execution:** Given knowledge about potential or existing problems and threats, simulate different scenarios, formulate courses of action (plans), and trigger actuators (applications or humans) to carry out the courses of action in a distributed environment. Continually plan for contingencies, as the environment is open, dynamic and ever changing.

- **Command and Control:** Support distributed collaboration among decision makers in visualizing relevant information (coming from the Awareness layer), deciding which courses of action to take (coming from the Action layer), and monitoring the execution of the selected plans.
- **Data and Process Integration:** Tie applications, users, and systems within and across the Awareness, Action, Control layers in a dynamic and decentralized fashion with a high degree of scalability, reliability, and security in adherence to given policies and procedures.

Highlights of the developed technology in the corresponding areas include:

- **Awareness**
  - Intelligent Data management enabling access to heterogeneous data from distributed data sources.
  - Probabilistic reasoning enabling knowledge discovery and syndromic surveillance based on causal evidence.
- **Action**
  - Optimal placement of sensors and other resources over dynamically defined geographic regions.
  - Real-time adaptive planning for contingencies.
  - Decentralized coordination of plan execution by distributed autonomous systems.
- **Control**
  - Integrated Knowledge Environment supporting user-driven information sharing and application composition.
- **Integration**
  - Flexible multi-agent framework enabling dynamic service discovery and invocation across a network.

The technologies developed under IBWTP were designed to support early detection and rapid response to biological and chemical threats, and are applicable to decision making and support in these areas in Homeland Security, Defense, and Agriculture. However, as the underlying technologies were designed with the goal of being broadly applicable, the results of the IBWTP project can be used in many domains beyond the initial target of real-time decision making, including applications in simulations, functional exercises, and field exercises in training and preparation for emergency management and emergency response at the urban, regional, state, and federal levels.

Furthermore, the technologies are applicable to other areas of defense, especially towards enhancing the Future Naval Capability (FNC) Knowledge Superiority and Assurance (KSA) in support of the transformation of the Navy to meet the emerging threats in the 21st Century. In particular, the technologies provide enabling capabilities for Composeable FORCEnet, Operational Adaptation, Human Systems Integration, Warfighter Defense, Manpower Reduction, and Joint Battle Management.

Based on the developed technologies, QLI has submitted 11 patent applications, published 5 conference proceeding/journal articles, and has written 11 technical reports.

## **2. Introduction**

This Final Report provides an overview of the research and development effort carried out by Quantum Leap Innovations, Inc. (QLI) under contract to the Office of Naval Research (ONR) during the period from July 2002 to September 2006. Within the context of the Integrated Biological Warfare Technology Platform (IBWTP) program, QLI was tasked by ONR to develop, evaluate, and demonstrate novel technology supporting early detection of and rapid response to biological or chemical threats. This report provides an overview of the challenges QLI faced, the approach it took to creating the technologies, and some of the specific technological solutions in the areas of Situational Awareness, Course of Action Planning, Command & Control, and Data & Process Integration. It also presents the applicability of the developed technologies to areas other than biological response, such as Department of Homeland Security applications in emergency management, and Department of Defense applications in force transformation, especially regarding Future Naval Capability (FNC) Knowledge Superiority and Assurance (KSA).

Section 3, Synopsis, of this report provides a high level overview of the issues facing this country that the project addressed, the approach, overview of the technological solution, as well as the applicability of the solution to other areas. Section 4, Problem Details, provides the motivation as well as a background of currently available technology and technology gaps. Section 5, Methods, Assumptions, and Procedures, describes the evolution of the approach over the duration of the project, including the architectural framework, software development processes, and project work breakdown structure. Section 6, Results and Discussion, provides a more detailed overview of the specific technologies developed during the project. Section 7, Conclusions, presents the applicability of the developed technology to the biodefense domain and domains of Homeland Security and Defense.

## 3. Synopsis

### 3.1. Problem

A global threat to the people of the United States has recently emerged, with implications comparable to the nuclear devastation we faced in the cold war era. A combination of easily accessible pathogens, low cost of development and dispersal, and the demonstrated strategy of terrorist adversaries to target civilian and commercial interests, indicates that we should prepare for biological attacks.

Well-informed studies have observed that we lack the sensor technology to broadly and accurately detect many of the biological weapons that adversaries could wield against us. Additionally, vaccines are lacking for many bioweapons, may have serious side-effects, may not be effective for weaponized variants of natural pathogens, and vaccination compliance is low, even when effective vaccines are freely available. Many potential bioweapons currently lack any rapid detection and some also lack effective treatment.

Though there are similarities between use of bioweapons and the emergence of naturally occurring diseases, there is a fundamental difference in the danger and logistics posed by pathogens that are deliberately introduced into a population. Typical disease epidemiology proceeds gradually, through random contacts, giving healthcare workers and infectious disease specialists time to analyze early victims, alert the medical community, and determine the best course of treatment and prevention. In many cases, low rates of transmission and “herd immunities” reduce incidence and prevalence of diseases. In contrast to the case of naturally occurring pathogens, bioterrorists can, in principal, expose a large number of initial victims over a wide area to diseases for which there is little natural immunity, so that there is little time between detection of an index case and pervasive disease in the population. Exploitation of contagious disease as a weapon magnifies this effect – turning every infected victim into an unwitting ally. In some scenarios, many health facilities will be simultaneously overwhelmed and in need of regional and national assistance. Even the best preparation will be insufficient for such an event.

The openness and mobility of western society make the US population an easy target for foes that do not rely on military weapons, and especially for those who have no particular vulnerability to retaliation. Thus, it is prudent to expect that eventually one or more biological attacks will be successful, and that multiple agencies from local to national levels will be required to act in concert to minimize damage to the population. Resources such as the CDC's Strategic National Stockpile (SNS) are effective only if they are provided to the appropriate population sufficiently early in the progression of disease. Early detection and coordinated response can drastically improve the outlook for affected populations. Also, it is prudent to expect that at least some local agencies will need to act initially on their own, both in detection and treatment of disease, and in trying to uncover the signal events and paths of exposure that predict additional cases.

A small number of anthrax-laden letters mailed in November, 2001 resulted in the death of 5 victims, illness of 22 people, disruption of US Postal Activities and many

related businesses for a variable length of time, and roughly \$200MM in costs for decontamination and remediation of US Post Office facilities. The authors of that attack are still at large.

Though the anthrax attack was costly, it is *relatively modest* compared to potential bio-warfare attacks, where some estimates, for communicable infectious diseases such as smallpox, pose 100,000 initial victims leading to 30 million deaths in four months. Clearly, bioweapons are competitive with nuclear weaponry in terms of inflicting death and disruption on society, yet the technology to create a bioweapons is much more accessible (than that to create nuclear weapons) to non-state actors such as Al Qaeda. Delivery of a bioweapons is also very difficult to prevent, as millions of doses of an active pathogen may be easily concealed on the person of a terrorist.

Though we cannot eliminate the threat of a biological terrorist attack, we can vastly improve the outcome for the US population via a combination of early detection of threats and rapid response to mitigate the damage. This challenge calls for the creation of an analytic and decision support system that continually monitors data for patterns that indicate an attack, alerts the appropriate individuals and agencies when an incident reaches some threshold of significance, maintains a current capability of simulation and planning for any locale affected by the attack, supports cooperative analysis and decision making by domain experts and aids in coordinated response, including appropriate stakeholders from emergency management, medical, intelligence, and law-enforcement bodies at local, state, regional and national levels.

At best, local agencies create static “red-book” plans to respond to major types of emergencies. These plans often lack sufficient specificity (such as awareness of daily or seasonal current population distributions, current loading of healthcare facilities, and specific projections given a particular disease event) or concreteness (such as the allocation of particular personnel to neighborhood health centers). A system must be created that supports representation and maintenance of dynamically updated plans that account for important changing conditions.

To satisfy the varied users of such a system, it must provide easy integration of data from a wide variety of sources, such as hospital and physician reports, and pharmacy sales, event notices, weather data, police reports, and national or local threat assessment levels. The system must provide probabilistic assessment of the threats, and permit domain experts to use their tools of choice to perform assessments, run scenarios, consider or generate plans, and effectively communicate their findings.

To exploit local knowledge and awareness, to provide robust operation, and to prevent bottlenecks inherent in monolithic approaches, the system must provide a decentralized but connected network, allowing local users to coordinate with more central ones, while avoiding information overload of any party.

Prior to work on the IBWTP system, no comprehensive technologies existed to provide the continual monitoring, planning and coordination needed to satisfy the previously-stated requirements.

Some of the major elements lacking in existing technologies include:

- Automatic integration of data from multiple sources to feed on-the-fly analysis

- Data provisioning to provide temporally or locally inaccessible data in support of anytime analysis and simulation.
- Continual update of planning and optimization models with current state values and probabilistic threat assessments.
- Distributed decentralized interfaces that allow domain experts to dynamically compose and manipulate appropriate analytic applications, simulations and visualizations
- Anytime planning and optimization within uncertain domains
- Integration and manipulation of arbitrary analytic components through an easy-to-use visual composition system.

### 3.2. Approach

This section provides an overview of the approach taken by QLI to address the aforementioned problems. Further details of the approach are elaborated in Section 5 of this report.

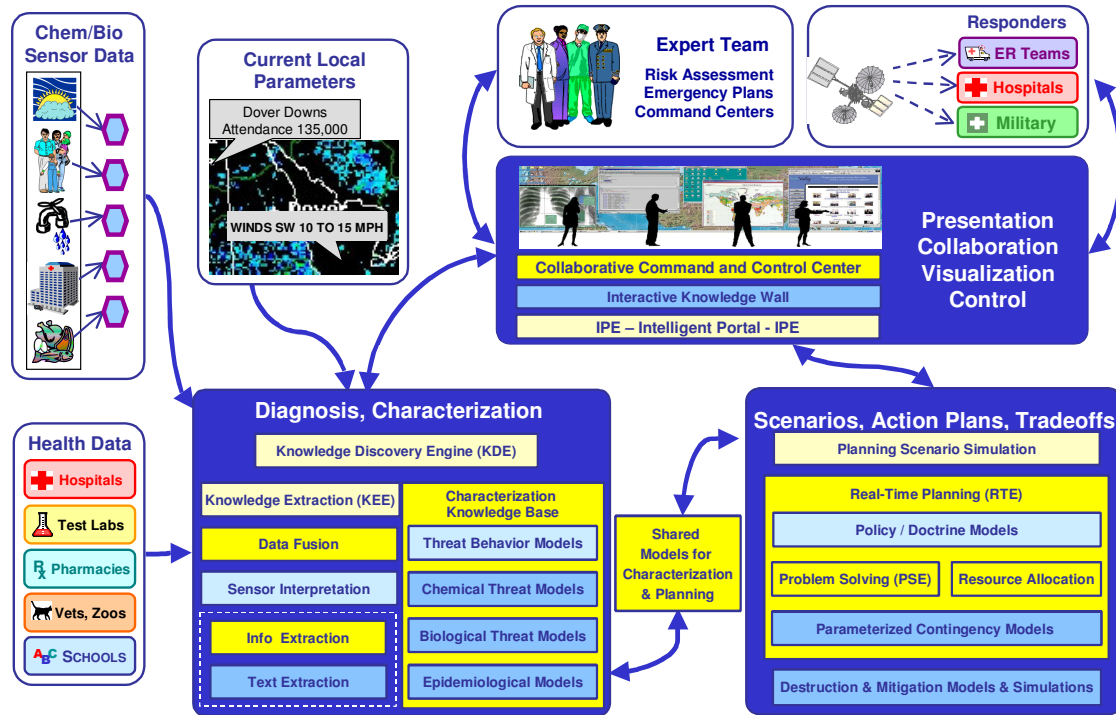
The objective of the Integrated Biological Warfare Technology Platform (IBWTP) program was to develop an integrated decision support framework for defense against chemical and biological warfare. The framework enables the integration of static and dynamic data (e.g. from hospitals, sensors, open source, intelligence), models (e.g. dispersion, exposure, damage), and advanced intelligent computing technologies to create a powerful early detection and rapid response system to identify and respond to potential or existing biological or chemical threats, either man-made or natural. It enables crisis managers to:

- Monitor chemical or biological outbreaks,
- Identify the cause(s) of the outbreak and its (their) possible sources,
- Predict potential exposure,
- Plan for effective response, and
- Alert appropriate authorities to mitigate the damage (hospitals, local government, law enforcement, military and CDC).

The system supports a variety of potential scenarios and continuously updates real world action plans in a format that improves the efficiency and quality of collaborative decision-making. Collaborative emergency planning and management is facilitated through an interactive knowledge visualization and decision making environment that supports teams of different users (ranging from technical specialists to high-level decision makers) in a single space or distributed across different geographic locations.

The approach adopted at the outset of IBWTP is shown in Figure 1. The large, dark blue boxes depict the three major IBWTP system components. In each box, QLI targeted initial development of the technologies represented in the yellow subcomponents, whereas technologies represented in the blue components are required from external sources (COTS, GOTS, and domain expertise). The “Diagnosis and Characterization” component contains the techniques and models required to fuse, analyze, and reason about vast amounts of information. Outputs of this component include the detection and

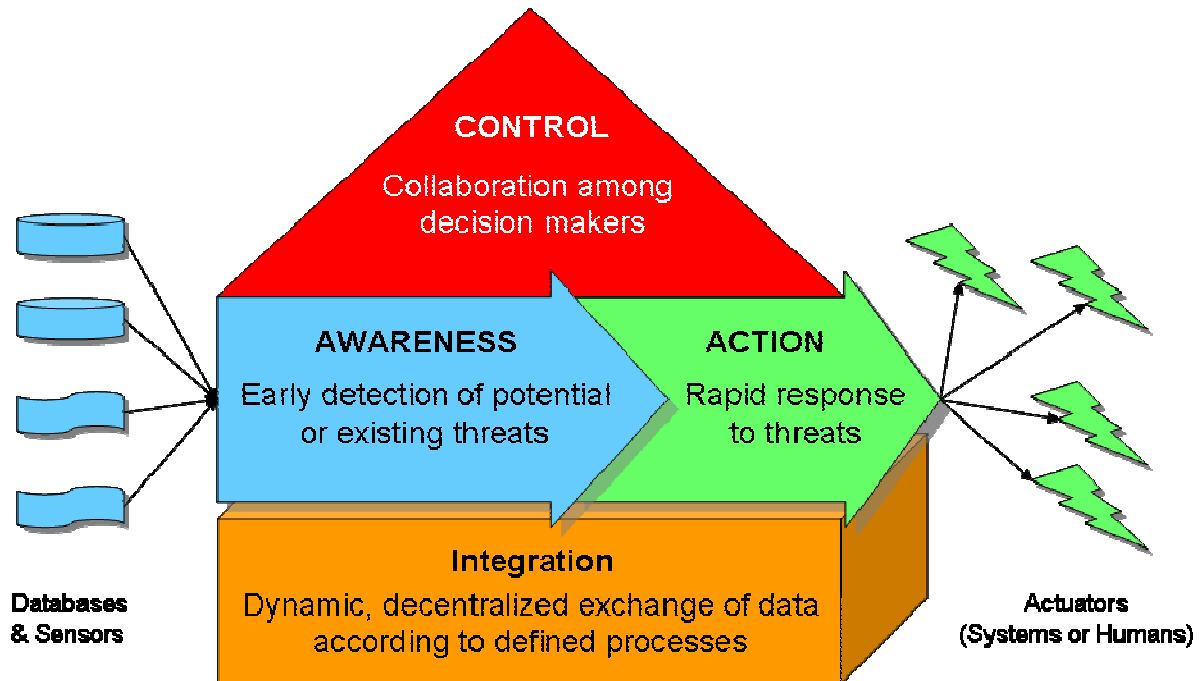
identification of harmful biological or chemical agents. The “Scenarios, Action Plans, Tradeoffs” component contains the techniques and models for proactively planning about the source, exposure, and response to harmful agents. The “Presentation, Collaboration, Visualization, and Control” component contains techniques for presenting the information gleaned from processing huge amounts of data to the decision-maker or technical user, in a format that is easy to understand and manipulate. The goal of IBWTP was to provide an integrated framework tying the various technologies within and across the components.



**Figure 1: The Initial IBWTP Approach by QLI**

As work on IBWTP progressed, it became clear that many of the technologies under development were suitable not only for the biological and chemical defense domain, but were equally applicable to a variety of other domains, especially emergency response in general, netcentric warfare, and intelligent enterprise solutions. With this in mind, QLI developed a more general architectural framework around which the core domain-independent technologies could be developed and integrated. This architectural framework was designated Awareness/Action/Control/Integration (AACI) as it integrates technologies and capabilities from Situational Awareness, Course of Action Planning and Execution, and Command and Control. Using the AACI framework, QLI could then represent and layer domain-specific functionality, knowledge representation, and models, to address specific application areas including but not limited to the original project goal of biological and chemical defense. In particular, the framework can be used to effectively represent and deploy solutions addressing the Naval FORCENet requirements. The AACI framework is depicted in Figure 2.





**Figure 2: The Integrated Awareness/Action/Control (AACI) Framework**

The four areas of the framework are:

- **Situational Awareness:** Provide situational awareness by transforming dynamic, distributed, and heterogeneous data into actionable knowledge in order to identify and localize potential or existing problems and threats as early as possible. Share this knowledge with relevant users and applications as soon as possible.
- **Course of Action Planning, Optimization and Execution:** Given knowledge about potential or existing problems and threats, simulate different scenarios, formulate courses of action (plans), and trigger actuators (either applications or humans) to carry out the courses of action in a distributed environment. Continually plan for contingencies, as the environment is open, dynamic and ever changing.
- **Command and Control:** Support distributed collaboration among decision makers in visualizing relevant information (coming from the Awareness layer), deciding which courses of action to take (coming from the Action layer), and monitoring the execution of the selected plans.
- **Data and Process Integration:** Tie applications, users, and systems within and across the Awareness, Action, Control layers in a dynamic and decentralized fashion with a high degree of scalability, reliability, and security in adherence to policies and procedures.

The goal is to automate the Awareness and Action layers as much as possible, while still keeping the human users and decision makers in the loop through the Control layer.

The majority of effort by QLI in IBWTP was in the development and demonstration of novel technologies in the four areas of Awareness, Action, Control, and Integration. The approach followed a spiral development, where basic research was conducted to

develop the core technology. For the most successful research components, the technologies would then be strengthened to satisfy software quality standards.

In order to effectively support the transitioning of technology from research proof-of-concepts and demonstrators to more substantial prototypes and pilots, QLI developed a unique “Technology Transfer” process allowing for professional software engineers and developers not only to advance the software quality, maintainability, and maturity, but also to develop applications and solutions employing the technology for further validation in successor projects to IBWTP. All software development used the JAVA™ 2 Standard Edition (J2SE) programming language to ensure maximal portability among different operating systems and inclusion into modern programming environments.

### **3.3. Solution**

This section provides an overview of the QLI technology development in the AACI framework. Implementations of the software are also referred to as the Integrated Biological and Chemical Warfare Defense (IBCWD) software. These individual technology solutions are presented in more detail in Section 6.

#### **3.3.1. Awareness Capabilities**

The QLI IBWTP team developed advanced mechanisms for enhancing situational awareness, such as coordinating data analysis, fusion, and probabilistic reasoning systems to more rapidly and accurately alert IBWTP users to potentially harmful agents, such as chemical and biological agents. These systems provide access to heterogeneous data from a variety of distributed static and dynamic data sources (e.g. databases, sensors) as well as provide advanced data analysis mechanisms operating on the data. The systems are integrated by using multi-agent techniques in a grid-like network. This has the advantage of being able to integrate systems across geographical and organizational boundaries while preserving individual autonomy. In particular, the IBWTP team:

- Provided automated discovery and access to comprehensive on-line data sources via multi-agent techniques.
- Investigated mechanisms for automatically analyzing results of data fusion and reasoning engines. This enables rapid triggering of alarms to decision makers as well as triggering further, more exhaustive and comprehensive, analyses.
- Developed mechanisms to combine the behavior of separate input models to provide expanded datasets for drawing better conclusions.
- Enabled cooperative resource, results, and goal sharing among disparate data fusion and reasoning engines, applications, and users.
- Developed multi-phased analysis mechanisms, whereby conclusions based on readily accessible easy-to-process data subsets trigger processing based on more exhaustive (and therefore more expensive-to-process) data sets.
- Analyzed methods of automatic information extraction from unstructured text-based information sources.
- Increased number and scope of data fusion and reasoning engines to encompass traditional data mining techniques and deductive inference mechanisms.

- Demonstrated the technologies in the areas of dispersion modeling, syndromic surveillance of diseases, and integration of weather data.

To accomplish this, QLI drew upon and enhanced core technologies in the areas of probabilistic and Bayesian reasoning, multi-agent techniques, and the semantic web.

### **3.3.2. Action Capabilities**

The QLI IBWTP team developed mechanisms for contingency planning, optimizing, and executing plans by distributed autonomous systems. In particular, the IBWTP team:

- Developed a framework for optimal placement of resources (such as sensors, medical facilities, UAVs) over a geographical region satisfying a number of coverage constraints and targets.
- Incorporated anytime algorithms to be able to provide the best available plans at any time of the execution phase.
- Developed techniques using probabilistic AI to enable real-time planning in uncertain environments.
- Introduced probabilistic representations of the past/present/future world states as well as probabilistic representations of targeted goals.
- Developed optimized decision making algorithms drawing upon the probabilistic representation of world states and goals.
- Incorporated advanced plan representation mechanisms to enable automated shared execution and coordination of plans across multiple autonomous actors.
- Demonstrated the technologies in the areas of sensor placement and route planning in unknown environments.

To accomplish this, QLI drew upon and enhanced core technologies in the areas of optimization and problem solving, business process management, probabilistic reasoning, and multi-agent systems.

### **3.3.3. Control Capabilities**

The QLI IBWTP team developed technology to support collaborative decision making processes required to maintain control over the Awareness and Action levels by users and decision makers. In particular, the IBWTP team:

- Developed an “integrated Knowledge Environment” (iKE) supporting seamless integration of information from many disparate sources into a common shared visualized information space.
- Enabled the integration and user-guided on-demand composition among QLI-developed, legacy, and COTS applications within the iKE to more comprehensively evaluate and process information.
- Designed and constructed fixed and mobile “Interactive Knowledge Walls” supporting distributed interaction among users using iKE.
- Developed and integrated mechanisms into iKE supporting collaboration among distributed teams of users and autonomous agents.
- Demonstrated the technologies in the area of disaster management.

To accomplish this, QLI drew upon and enhanced core technologies in the areas of collaboration, computer supported cooperative work (CSCW), user modeling, tailorability, semantic web, and multi-agent systems.

### **3.3.4. Integration Capabilities**

The QLI IBWTP team developed and deployed a flexible decentralized platform for integration and coordination of heterogeneous applications and data sources. This enables rapid dynamic composition of applications required to support and integrate the Awareness, Action, and Control capabilities. It also enables the seamless composition of the capabilities with each other. In particular, the IBWTP team:

- Developed and tested the Multi-Agent Development Environment for developing, deploying, and coordinating interaction among autonomous systems
- Implemented techniques facilitating monitoring and evaluating attributes of large scale distributed multi-agent systems.
- Investigated mechanisms for policy management in multi-agent systems, especially in order to support authorized execution of data queries, data provisioning, and application invocation.
- Developed mechanisms for automated composition of applications based on rich semantic descriptions and discovery.
- Developed a unified glossary of terms related to service oriented architectures.

To accomplish this, QLI drew upon and enhanced core technologies in the areas of multi-agent system technology, service oriented architectures, and the semantic web.

## **3.4. Applicability**

The technologies developed under IBWTP were designed to support early detection and rapid response to biological and chemical threats, and are applicable to decision making and support in these areas in Homeland Security, Defense, and Agriculture. However, as the underlying technologies were designed with the goal of being as broadly applicable as possible to other areas, the results of the IBWTP project can be used not only in real-time decision making but also in simulations, functional exercises, and field exercises in training and preparation for emergency management and emergency response at the urban, regional, state, and federal levels. User requirements and specific targeted application areas were obtained from discussions with the Delaware Emergency Management Agency (DEMA) and Delaware Department of Health and Social Services (DHSS).

Furthermore, the technologies are applicable to other areas of defense, especially towards enhancing the Future Naval Capability (FNC) Knowledge Superiority and Assurance (KSA) in support of the transformation of the Navy to meet the emerging threats in the 21st Century. In particular, the technologies provide enabling capabilities for Composeable FORCEnet, Operational Adaptation, Human Systems Integration, Warfighter Defense, Manpower Reduction, and Joint Battle Management.

Section 7.1.2 provides a more detailed analysis of some of the ways in which the technologies developed under IBWTP are applicable to FORCEnet.

## 4. Problem Details

### 4.1. Motivation

#### 4.1.1. The Emerging Threat

A global threat to the people of the United States has recently emerged, with implications comparable to the nuclear devastation we faced in the cold war era. A combination of easily accessible pathogens, low cost of development and dispersal, and the demonstrated strategy of terrorist adversaries to target civilian and commercial interests, indicates that we should prepare for biological attacks.

Well-informed studies have observed that we lack the sensor technology to broadly and accurately detect many of the biological weapons that adversaries could wield against us [1]. Additionally, vaccines are lacking for many bioweapons, may have serious side-effects, may not be effective for weaponized variants of natural pathogens, and vaccination compliance is low, even when effective vaccines are freely available [2]. Many potential bioweapons currently lack any rapid detection and some also lack effective treatment [3].

Though there are similarities between use of bioweapons and the emergence of naturally occurring diseases, there is a fundamental difference in the danger and logistics posed by pathogens that are deliberately introduced into a population. Typical disease epidemiology proceeds gradually, through random contacts, giving healthcare workers and infectious disease specialists time to analyze early victims, alert the medical community, and determine the best course of treatment and prevention. In many cases, low rates of transmission and “herd immunities” reduce incidence and prevalence of diseases. In contrast to the case of naturally occurring pathogens, bioterrorists can, in principal, expose a large number of initial victims over a wide area to diseases for which there is little natural immunity, so that there is little time between detection of an index case and pervasive disease in the population. Exploitation of contagious disease as a weapon magnifies this effect – turning every infected victim into an unwitting ally. In some scenarios, many health facilities will be simultaneously overwhelmed and in need of regional and national assistance. Even the best preparation will be insufficient for such an event.

The openness and mobility of western society make the US population an easy target for foes that do not rely on military weapons, and especially for those who have no particular vulnerability to retaliation. Thus, it is prudent to expect that eventually one or more biological attacks will be successful, and that multiple agencies from local to national levels will be required to act in concert to minimize damage to the population. Resources such as the CDC's Strategic National Stockpile (SNS) are effective only if they are provided to the appropriate population sufficiently early in the progression of disease [4]. Early detection and coordinated response can drastically improve the outlook for affected populations. Also, it is prudent to expect that at least some local agencies will need to act initially on their own, both in detection and treatment of disease, and in trying to uncover the signal events and paths of exposure that predict additional cases.

#### 4.1.2. Potential Consequences

A small number of anthrax-laden letters mailed in November, 2001 resulted in the death of 5 victims, illness of 22 people, disruption of US Postal Activities and many related businesses for a variable length of time, and roughly \$200MM in costs for decontamination and remediation of US Post Office facilities [5]. The instigators of that attack are still at large.

Though the anthrax attack was costly, it is *relatively modest* compared to potential bio-warfare attacks, where some estimates, for communicable infectious diseases such as smallpox, pose 100,000 initial victims leading to 30 million deaths in four months [6]. Clearly, bioweapons are competitive with nuclear weaponry in terms of inflicting death and disruption on society, yet the technology to create a bioweapons is much more accessible (than that to create nuclear weapons) to non-state actors such as Al Qaeda [7]. Delivery of a bioweapons is also very difficult to prevent, as millions of doses of an active pathogen may be easily concealed on the person of a terrorist.

#### 4.1.3. What Is Required

Though we cannot eliminate the threat of a biological terrorist attack, we can vastly improve the outcome for the US population via a combination of early detection of threats and rapid response to mitigate the damage. This challenge calls for the creation of an analytic and decision support system that continually monitors data for patterns that indicate an attack, alerts the appropriate individuals and agencies when an incident reaches some threshold of significance, maintains a current capability of simulation and planning for any locale affected by the attack, supports cooperative analysis and decision making by domain experts and aids in coordinated response, including appropriate stakeholders from emergency management, medical, intelligence, and law-enforcement bodies at local, state, regional and national levels.

At best, local agencies create static “red-book” plans to respond to major types of emergencies. These plans often lack sufficient specificity (such as awareness of daily or seasonal current population distributions, current loading of healthcare facilities, and specific projections given a particular disease event) or concreteness (such as the allocation of particular personnel to neighborhood health centers). A system must be created that supports representation and maintenance of dynamically updated plans that account for important changing conditions.

To satisfy the varied users of such a system, it must be provide easy integration of data from a wide variety of sources, such as hospital and physician reports, and pharmacy sales, event notices, weather data, police reports, and national or local threat assessment levels. The system must provide probabilistic assessment of the threats, and permit domain experts to use their tools of choice to perform assessments, run scenarios, consider or generate plans, and effectively communicate their findings.

To exploit local knowledge and awareness, to provide robust operation, and to prevent bottlenecks inherent in monolithic approaches, the system must provide a decentralized but connected network, allowing local users to coordinate with more central ones, while avoiding information overload of any party.

## 4.2. Background

It has been observed that the US cannot provide a specific defense for all of the varied biological agents, especially those which are engineered to resist drug therapy and vaccines [8]. Simultaneously, we have come to the realization that biological warfare (BW) attacks of far greater magnitude than the November Anthrax attack of 2001 can occur at any time. Existing disease-prevention processes and agencies, designed to combat naturally-occurring threats, are likely to be too slow and too limited to forestall a large-scale deliberate attack.

To detect biological threats, characterize their nature and consequences, and respond quickly and effectively, we need to advance the practical application of software technologies in the areas of distributed knowledge discovery, continual (real-time) planning, cooperative analysis and control, and in the integration of technical components such as data analysis, simulations, visualization, and planning components.

### 4.2.1. Biological Threats

The CDC lists three categories of biological agents that terrorists might be expected to use [9]. Category A agents pose a risk to national security because they are easily introduced or transmitted to the population, produce high mortality, have major impact on public health, and may cause severe disruption of society. These agents include:

- Anthrax
- Botulism (*Clostridium botulinum* toxin)
- Plague (*Yersinia pestis*)
- Smallpox (*variola major*)
- Tularemia (*Francisella tularensis*)
- Viral hemorrhagic fevers (filoviruses [e.g., Ebola, Marburg] and arenaviruses [e.g., Lassa, Machupo])

Category B agents are the second highest priority, as they are relatively easy to introduce, have moderate morbidity rates and low mortality rates. These agents include:

- Brucellosis (*Brucella* species)
- Epsilon toxin of *Clostridium perfringens*
- Food safety threats (e.g., *Salmonella* species, *Escherichia coli* O157:H7, *Shigella*)
- Glanders (*Burkholderia mallei*)
- Melioidosis (*Burkholderia pseudomallei*)
- Psittacosis (*Chlamydia psittaci*)
- Q fever (*Coxiella burnetii*)
- Ricin toxin from *Ricinus communis* (castor beans)
- Staphylococcal enterotoxin B
- Typhus fever (*Rickettsia prowazekii*)
- Viral encephalitis (alphaviruses [e.g., Venezuelan equine encephalitis, eastern equine encephalitis, western equine encephalitis])

- Water safety threats (e.g., *Vibrio cholerae*, *Cryptosporidium parvum*)

Category C agents are the third highest priority, and include emerging pathogens that could be engineered for mass dissemination in the future because of their availability, ease of production, and potential for high morbidity, mortality and health impact. These agents include:

- Emerging infectious diseases such as Nipah virus and hantavirus

Clearly, there are more than enough threats to provide nightmares for public health officials, but even this list is a gloss of the true depth of the threat. There are many existing strains of almost every agent, and the strains may be combined or augmented to reduce the effectiveness of therapy, or even to stymie differential diagnosis.

Each of the BW threats is associated with varied etiologies and symptoms, and unfortunately, most of them lack decisive unambiguous signs in the prodromic stages.

A related form of bioterrorism is agroterrorism, which is aimed at livestock populations. This approach to bioterrorism can be economically devastating, and in the case of zoonotic disease, offer a second avenue to attack the human population [10]. The agro-BW agents must also be considered and monitored in any comprehensive system.

#### **4.2.2. National Public Health Surveillance**

The CDC's goals for public health surveillance clearly omit consideration of intentional use of pathogens, as the surveillance activities are aimed at naturally occurring disease, view very long time-lines (months to decades) and are aimed at improving health practices rather than detecting an unfolding attack [11]. Stated goals of the CDC health surveillance are:

- Estimate magnitude of the problem
- Determine geographic distribution of illness
- Portray the natural history of a disease
- Detect epidemics/define a problem
- Generate hypotheses, stimulate research
- Evaluate control measures
- Monitor changes in infectious agents
- Detect changes in health practices
- Facilitate planning

An overview of the epidemiologic indicators of a biological attack (from The U.S. Army Medical Research Institute of Infectious Diseases) lists factors such as intelligence information and lack of genetic diversity of strains that are not typically considered in public health analysis of disease patterns [12]:

- The presence of a large epidemic with a similar disease or syndrome, especially in a discrete population
- Many cases of unexplained diseases or deaths
- More severe disease than is usually expected for a specific pathogen or failure to respond to standard therapy



- Unusual routes of exposure for a pathogen, such as the inhalational route for diseases that normally occur through other exposures
- A disease that is unusual for a given geographic area or transmission season
- Disease normally transmitted by a vector that is not present in the local area
- Multiple simultaneous or serial epidemics of different diseases in the same population
- A single case of disease by an uncommon agent (smallpox, some viral hemorrhagic fevers)
- A disease that is unusual for an age group
- Unusual strains or variants of organisms or antimicrobial resistance patterns different from those circulating
- Similar genetic type among agents isolated from distinct sources at different times or locations
- Higher attack rates in those exposed in certain areas, such as inside a building if released indoors, or lower rates in those inside a sealed building if released outside
- Disease outbreaks of the same illness occurring in noncontiguous areas
- A disease outbreak with zoonotic impact
- Intelligence of a potential attack, claims by a terrorist or aggressor of a release, and discovery of munitions or tampering

It is clear that an analysis and decision support system aimed at minimizing the effect of a BW attack must consider types of information (syndromic information, events providing mass exposure, police reports, veterinary reports, and weather information, to name a few) far beyond the typical purvey of disease monitoring, and that the system must provide continual local analysis in order to flag suspicious incidents as early as possible.

#### **4.2.3. Users of Systems for Combating Biological Warfare**

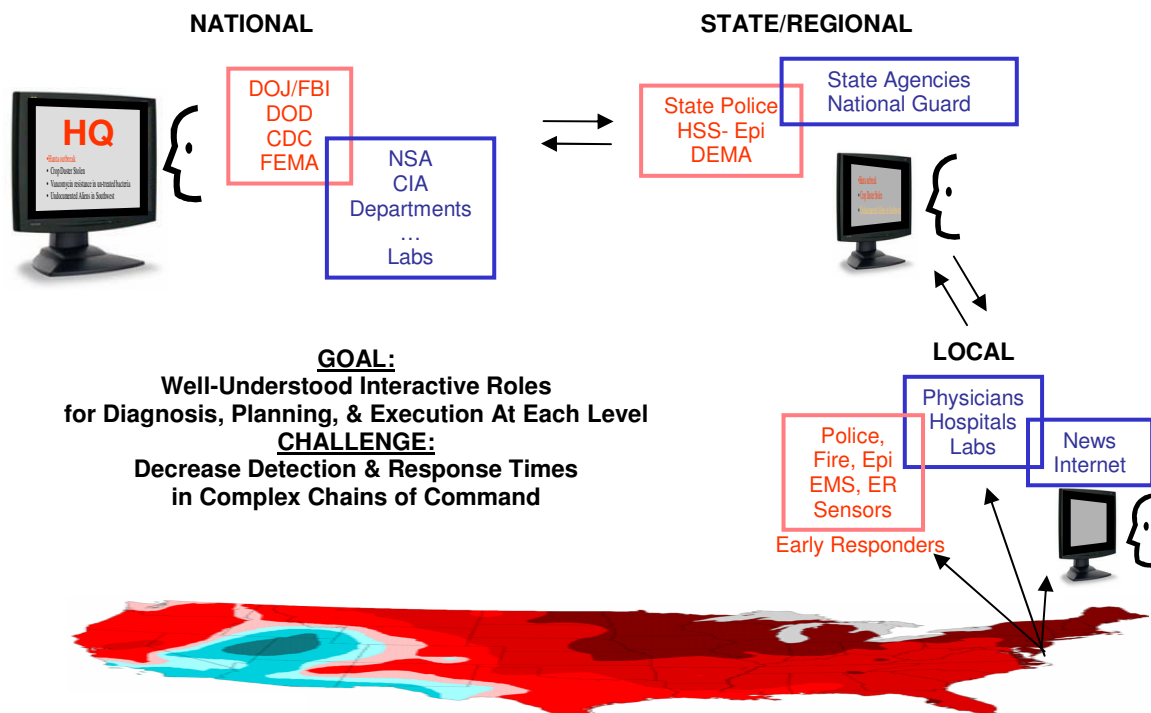
Many distinct groups of users at various levels provide the natural audience of a system aimed at mitigating the effectiveness of BW attacks, including:

- Local and regional health-care facilities
- Local, state, and national public health officials
- Local, metropolitan, and regional emergency responders
- Local, metropolitan, state, and national emergency management
- Defense personnel and leadership
- Local, metropolitan, state, and national law enforcement
- Intelligence Agencies
- Providers of critical commercial infrastructure (food, energy, transportation, communication)

Because of the diversity of users, and their disciplines, the system must be flexible enough to incorporate a wide array of analytic tools, yet provide natural integrating

platform and interface. The system must be highly configurable to accommodate the roles of many users and agencies, and must support interlocking networks of expertise.

The landscape of biosecurity users (cf. Figure 3), especially at the local and metropolitan levels, is constantly changing. Thus, a centralized architecture would not be tenable. It is also a mistake to assume that all analysis can be usefully performed at national level. The peculiarities of each site, from student illness during finals week to livestock deaths during a heat-wave, are much better comprehended at a local level than a central (regional or national) one. Conversely, relatively rare expertise in disease forensics, diagnosis and treatment of novel or rare diseases, and intelligence analysis is more likely to be found at national centers of excellence. Thus, an effective system must permit coordinated analysis by local and distant experts, each of whom contribute their unique insight, and all of whom can communicate salient data, findings, models, projections, and plans via shared visual representations. The system must exploit decentralized development of local analyses, while supporting oversight and specialized expertise from regional and national agencies.



**Figure 3: Biosecurity Landscape**

#### 4.2.4. Data Sources

A great volume and diversity of data has potential value in the discovery of BW attack, making a data-warehouse approach untenable, and demanding great flexibility in acquisition, meta-data representation, and provisioning. Some of the broad areas of data that are relevant to the system include:

- Reports of illnesses from local hospitals and other health providers
- Absences from schools and businesses
- Pharmaceutical sales – which can indicate a surge of symptoms

- Environmental reports -which can account for clusters of symptoms such as respiratory difficulties
- Police reports – which might mention suspicious individuals, or activities later associated with an outbreak.
- Intelligence reports, especially those warning of specific dissemination approaches
- Veterinary reports, both of pets and livestock, especially those involving zoonotic disease.
- A history of recent mass events such as football games or “Black Thursday” sales.
- Geographic information on which to overlay various data, hypotheses, simulations and plans.

#### **4.2.5. Data Acquisition and Provisioning**

Technologies required for accessing relevant data and making that data available to analytic components include:

- Text Extraction
- Information extraction
- Sensor interpretation
- Data transformation
- Data preprocessing
- Semantic integration
- Data fusion
- Knowledge Extraction
- Intelligent Caching
- Interpolation, Extrapolation
- Data quality assessment

Moreover, for reasons mentioned previously, it is not feasible to achieve all of these facets in a centralized, monolithic system. Any such system would lack the flexibility and scalability to handle the large and growing catalog of potentially relevant data. Thus a successful approach must allow distribution over a wide network of servers and data sources, placing much of the data acquisition and provisioning as close to the source as possible, and allowing the incorporation of expert analysis at multiple levels.

Additionally, the data acquisition and provisioning system must support both demand-driven and data-driven modes of operation, providing relevant data when requested by analytic components and/or users, while propagating new relevant data automatically to the appropriate analytic models and users.

#### **4.2.6. Analytic Methods and Models**

Because of the severe consequences of a biological attack, it is preferable to investigate many “false positives” rather than to fail to detect one “true positive”. The decision support system needs to integrate as much of the relevant information and models as possible, and to support probabilistic weighting of plausible attack scenarios. Many specific analytic methods are applicable to the BW area, and can be broadly

viewed as “supervised” approaches (in which expert users provide judgment on a set of positive and or negative examples, or projected quantitative outcomes) and “unsupervised” approaches – where the methods seek to uncover unvoiced relationships, so suggest new knowledge to the expert users.

Some of the supervised prediction and classification approaches that experts typically apply include:

- Bayesian Belief Nets
- Radial Basis Functions and Artificial Neural Networks
- Entropy/Mutual Information-based Learning
- Auto-Regressive Integrated Moving Averages (ARIMA)
- Iterative Dichotomiser (ID3) And Related C4.5 Approaches
- Association Rule Inference
- Causal Reasoning
- Inductive Logic Programming
- Dynamic Time-Warping Methods
- Frequent-Pattern Tree Approaches
- Hidden-Markov Models
- Linear and Logit Regression
- Regression Tree Approaches
- Kernel Methods And Support Vector Machines (SVMs)

Relevant unsupervised techniques include:

- Self-Organizing Maps
- K-Means and Hierarchical K-Means Clustering
- Latent Semantic Indexing
- Multi-Resolution Grid Clustering
- Distance-Based Outlier Detection

In many cases, pre-existing models and simulations are needed to make sense of information that is given, learned, or predicted. For the BW domain, some of the important models include:

- Threat Behavior Models
- Biological Threat Models
- Epidemiological Models
- Models of “normal” disease reports and variation
- Atmospheric Propagation models

The BW decision support system must provide an integrating framework for the many analytic approaches and domain-specific models mentioned above, along with additional approaches that experts find to be relevant. To be computationally and

organizationally tractable, the system must also support distribution of processing over locally distributed networks (LANs) and wide area networks (WANs).

#### **4.2.7. Planning Approaches**

Because of the severe consequences of a BW attack, and because of the extensive search space of plausible response plans, a pragmatic decision-support planner must be maintained in a “warm-start” mode at all times. That is, information and scenarios concerned with significant threats (as determined by human experts and or analytic models) along with relevant state-of-the-world information must be continually propagated through potential a plan analysis system, to reduce the time spent in purely reactive planning. Likelihood and consequence information from threat scenarios must be exploited to ensure that useful plans are available when needed.

To prevent wasted re-examination of plan components, the system requires an intelligent caching mechanism that preserves the most relevant plans, and the most reusable components of plans. The system must also ensure that plans achieve an appropriate level of concreteness.

One of the great challenges in developing a pragmatic system is that the majority of organizations charged with emergency planning have developed static “red-book” plans to respond to major types of emergencies. These plans are often too abstract to provide a concrete guideline, or too inflexible to adapt to ever-changing conditions. The planning system must permit planners to start with a representation of these static plans, but to extend and parameterize the plans so that they become continuously relevant.

#### **4.2.8. Interfaces for Collaborative Analysis and Decision Support**

Ultimately, the analytic and planning capabilities of the system must be monitored, aided, and directed by teams of human experts. To tie the many agencies and users together, and provide them with a consistent view of likely threats, plausible plans, and the ensuing trade-offs. The BW system will need to exploit ideas from Computer Supported Cooperative Work (CSCW) and Intelligent User Interface (IUI) design. It is especially important that the system provide a large visual state which has been shown to serve as an aid to individual cognition and which serves as a unifying reference for multiple local and distant collaborators [13]. At the same time, some users such as Emergency Medical Technicians (EMTs) may be mobile and may only be able to accommodate limited communication devices, such as PDAs and smart phones. The same collaborative infrastructure must be able to accommodate such uses and to provide acceptable interplay between large display stationary multi-user views and highly constrained individual devices. It is a significant technical challenge to permit many users to simultaneously manipulate a shared visual representation, while updating that representation with acceptable speed.

#### **4.2.9. The Integrating Platform**

Much has been said about the need for supporting distributed, decentralized data access, analysis, planning, and user interfaces. Additional requirements of flexibility, robustness and universality argue that traditional methods of multi-component integration are unsuitable for the desired system. For instance, many multi-application systems are constructed around sets of relational database tables – but such an approach would force

centralization and would be slow to adapt for new components. The ability to easily compose software components has long been a goal of many computer systems. In some senses, operating systems provide some of this ability, as do applications that offer component hooks or plug-in strategies. Unfortunately, these approaches fall far short of the goal of enabling typical users to confront new and unanticipated challenges.

In the case of operating systems, beyond the simplest level of multi-component use, such as flat files piped among simple Unix filters, there is little system-wide agreement on the semantics of data sources and data sinks – leading to small clusters of functionality which, in general, do not communicate with each other. Recent approaches to providing component integration, such as Component Object Model (COM), Common Object Request Broker Architecture (CORBA), High Level Architecture (HLA), and JavaBeans provide schemes to ease the programming difficulty of integration, but do little for the non-developer end-user who wishes to combine multiple processing components on the fly. Some software components require a long initialization period before they become useful. In such cases, it would be convenient to be able to link the running system into a larger software context, or to unlink that component when it is no longer of immediate utility.

Multi-Agent Systems (MAS) provide an avenue to both very flexible integration of components and support of distributed decentralized processes and users. Because MAS components operate at arms-length, using messaging to communicate and accomplish coordinated action, they can be constructed to survive failure of any individual component, and can efficiently incorporate new components as they become available.

### **4.3. Relevant Technologies**

Prior to work on the IBWTP system, no comprehensive technologies existed to provide the continual monitoring, planning and coordination needed to satisfy the previously-stated requirements.

Some of the major elements lacking in existing technologies include:

- Automatic integration of data from multiple sources to feed on-the-fly analysis
- Data provisioning to provide temporally or locally inaccessible data in support of anytime analysis and simulation.
- Continual update of planning and optimization models with current state values and probabilistic threat assessments.
- Distributed decentralized interfaces that allow domain experts to dynamically compose and manipulate appropriate analytic applications, simulations and visualizations
- Anytime planning and optimization within uncertain domains
- Integration and manipulation of arbitrary analytic components through an easy-to-use visual composition system.

Many existing technologies share some of the aims of IBWTP components, but are not directly applicable because of one or more intrinsic limitations. In some cases, such as existing biosurveillance systems, information produced via technological approaches can be incorporated into IBWTP in a fairly transparent way.

### 4.3.1. Existing Biosurveillance Systems

#### *Real-time Outbreak and Disease Surveillance System (RODS)*

Developed by Carnegie Mellon and the University of Pittsburgh the RODS system collects and analyzes relevant data automatically and in real-time, including emergency room registration data, microbiology culture results, reports of radiographs, and laboratory orders [14]. The system strives to recognize patterns of infectious disease, especially sudden and frequent outbreaks of cases involving flu-like symptoms, respiratory illnesses, diarrhea and paralysis. The system is available as open source, and has been used widely in Pennsylvania, and in several other states and municipalities in the United States, Canada and Taiwan.

#### *National Retail Data Monitor (NRDM)*

The National Retail Data Monitor (NRDM) is a public health surveillance tool that collects and analyzes over-the-counter healthcare product sales from eight large retail chains that sell over-the-counter (OTC) medications. These chains own 18,600 retail stores across the country [15]. The NRDM has over 540 users in 47 states. 1, 2 Studies of sales of OTC medications during outbreaks demonstrate that monitoring OTC sales can provide timely detection of disease outbreaks. The NRDM is a centralized data-warehousing system that incorporates new data hourly, and provides cached time-series information via a web services interface.

#### *Electronic Surveillance System for the Early Notification of Community-Based Epidemics (ESSENCE) and ESSENCE-II*

The ESSENCE Ambulatory Data System (ADS) diagnoses from 104 primary care and emergency clinics within a 50 mile radius of Washington, DC [16]. The diagnostic codes are grouped into "syndromic clusters" consistent with emerging infections including bioterrorism. Through the daily data downloads, traditional epidemiologic analyses using historical data for baseline comparisons, and more cutting edge analytic methods such as geographic information system approaches (GIS), the feasibility of the ESSENCE methodology was established. Currently ESSENCE downloads each day outpatient data from 121 Army, 110 Navy, 80 Air Force, and 2 Coast Guard installations around the world. Over 2700 syndrome- and location-specific graphs are prepared each day and automatically analyzed for patterns that suggest a need for further investigation. Beyond these centralized assessments, the graphs are available daily to approved DoD public health professionals on a secure web site.

ESSENCE-II is a DARPA-funded project including joint work by Johns Hopkins, George Washington University, Carnegie Mellon University, Cypcorp, and IBM. A key element of the approach is the exploitation of non-traditional sources of information on human and animal behavior during the early onset of symptoms. If abnormalities are found, supporting data like weather, regional disease states etc. will be mined and exploited to reduce false alarms. ESSENCE II will automatically perform active surveillance 24/7, alert and notify when abnormal conditions exist thereby relieving public health, epidemiologists, and preventive medicine personnel of the routine tasks associated with surveillance. Once operators have been notified of abnormal conditions, a suite of reasoning, data mining, and visualization tools will be provided to investigate

the potential outbreak in a timely fashion. ESSENCE II will be built around an agent-based architecture with communications occurring on an ontological level.

#### **4.3.2. Automatic Integration and Analysis of Data from Multiple Sources**

The Joint Battlespace Infosphere(JBI), pioneered by the Air Force Research Laboratory (AFRL) provides a flexible publish/subscribe data management and dissemination system that achieves some of the goals of decentralized data management and provisioning [17]. However, the system has no built-in capacity to distribute analytic models to data sources, and does not tackle the semantic integration issues that are especially prevalent among disparate organizations, and among users with varied domain expertise.

#### **4.3.3. Continuous Planning and Optimization Models**

Very few general purpose systems are aimed at the difficult problem of continual sensing, analysis and planning. SRI International's System for Interactive Planning and Execution (SIPE-2) provides one approach to this challenge [18]. While SIPE has some of the desired capabilities needed by a BW mitigation system, it is lacking in several important ways. SIPE is not constructed to easily incorporate new analytic components, probabilistic assessments or collaborative use.

Another existing system, the GRASP planner, developed at UMASS, has been used in simulations of adversarial planning environments, and has some relevance to the problem area [19]. This planner has advantages in using a supervenient hierarchy – which allows plan sub-components to be developed semi-independently, but, like SIPE, it does not exploit probabilistic information streaming from the analytic components.

Many MAS approaches also support anytime processing and design-to-time processing, which is applicable to continual planning [20]. In fact the IBWTP approach exploits this anytime capability in many facets of sensing and planning.



## 5. Methods, Assumptions, and Procedures

The methods for addressing the problem and challenges outlined in Section 4 were to perform software research and development in novel technologies of intelligent computing within the frameworks described in Section 5.1. The procedures supporting the chosen methods consisted of a spiral specification, development, and evaluation lifecycle transitioning from core research to mature software and are described in more detail in Section 5.2. The assumptions underlying the methods and procedures undertaken within IBWTP were that this approach would result in viable software that could be used to demonstrate and deploy the technology in applications addressing early detection and rapid response to biological and chemical threats, as well as emergency management and DoD force transformation applications.

### 5.1. Integrated Awareness/Action/Control Framework

*“Regarding survival of species*

*it is not the biggest, strongest, nor fastest that survive –  
rather, those that can adapt the fastest.”*

*- Charles Darwin*

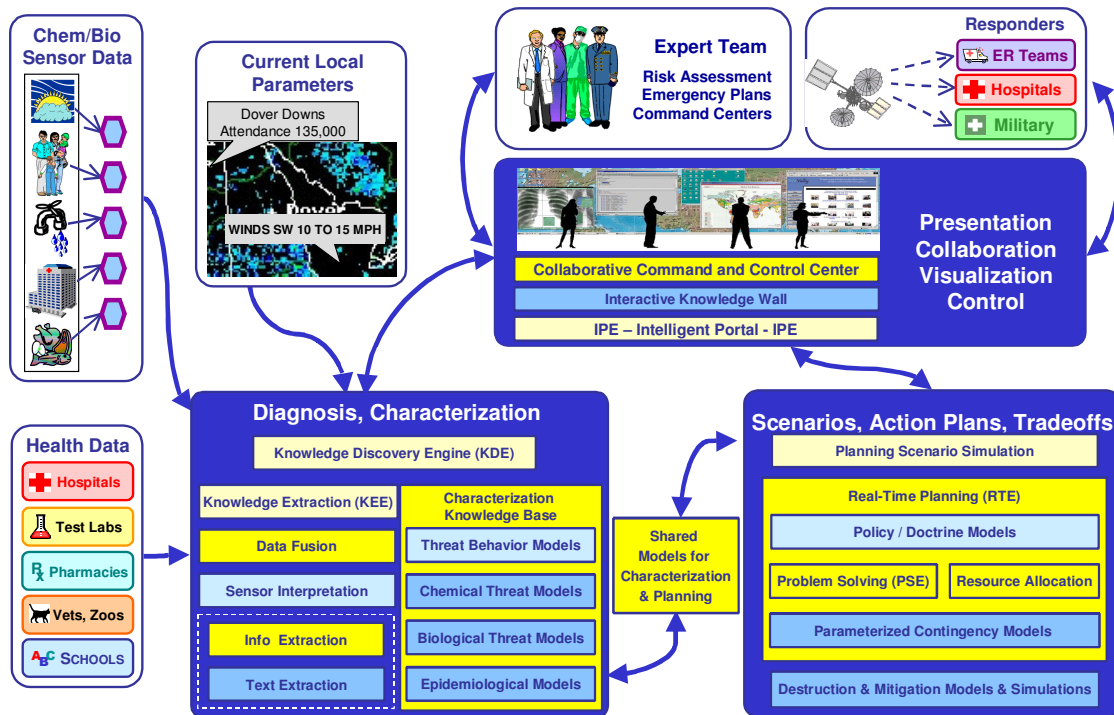
The objective of the Integrated Biological Warfare Technology Platform (IBWTP) program was to develop an integrated decision support framework for defense against chemical and biological warfare. The framework enables the integration of static and dynamic data (e.g. from hospitals, sensors, open source, intelligence), models (e.g. dispersion, exposure, damage) and advanced intelligent computing technologies to create a powerful early detection and rapid response system to identify and counter potential or existing biological or chemical threats, either man-made or natural. It enables crisis managers to:

- Monitor chemical or biological outbreaks,
- Identify the cause(s) of the outbreak and its (their) possible sources,
- Predict potential exposure,
- Plan for effective response, and
- Alert appropriate authorities to mitigate the damage (hospitals, local government, law enforcement, military and CDC).

The system supports a variety of potential scenarios and continuously updated real world action plans in a format that improves the efficiency and quality of collaborative decision-making. Collaborative emergency planning and management is facilitated through an interactive knowledge visualization and decision making environment that supports teams of different users (ranging from technical specialists to high-level decision makers) in a single space or distributed across different geographic locations.

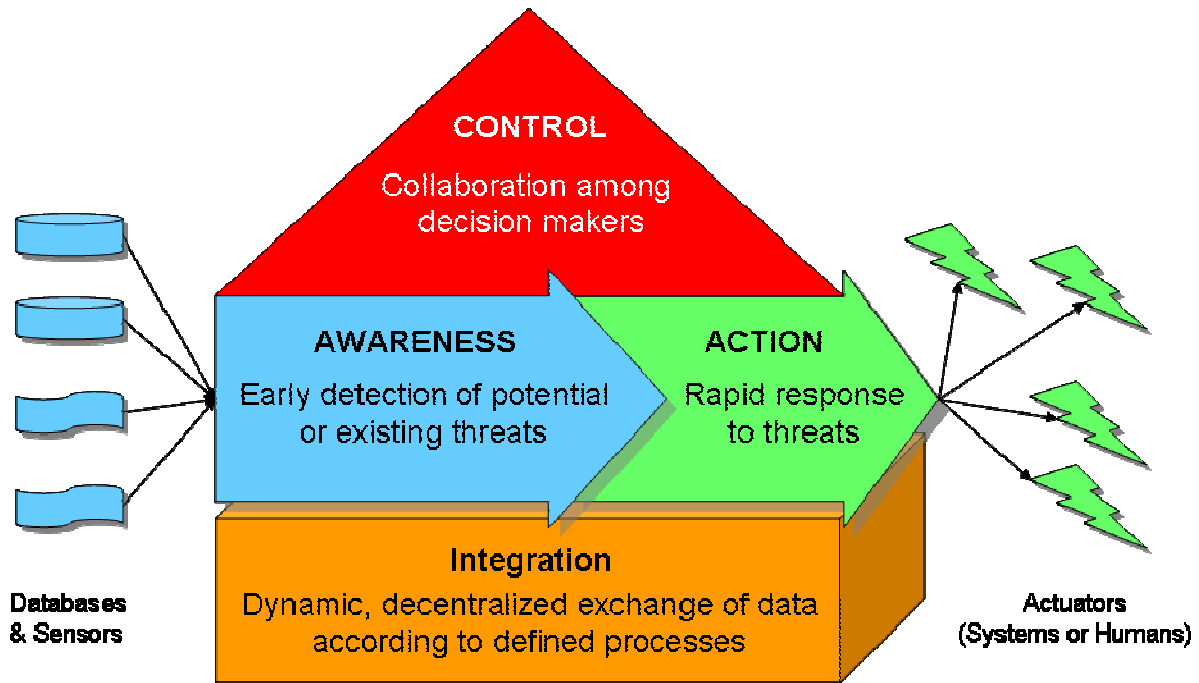
Figure 4 shows the approach adopted at the outset of IBWTP. The large, dark blue boxes depict the three major IBWTP system components. In each box, the yellow components were developed by QLI and the blue components come from external sources (COTS, GOTS). The “Diagnosis and Characterization” component contains the techniques and models required to fuse, analyze, and reason about vast amounts of

information. Outputs of this component include the detection and identification of harmful agents. The “Scenarios, Action Plans, Tradeoffs” component contains the techniques and models for proactively planning about the source, exposure, and response to harmful agents. The “Presentation, Collaboration, Visualization and Control” component contains techniques for presenting the information gleaned from processing huge amounts of data to the decision-maker or technical user, in a format that is easy to understand and manipulate.



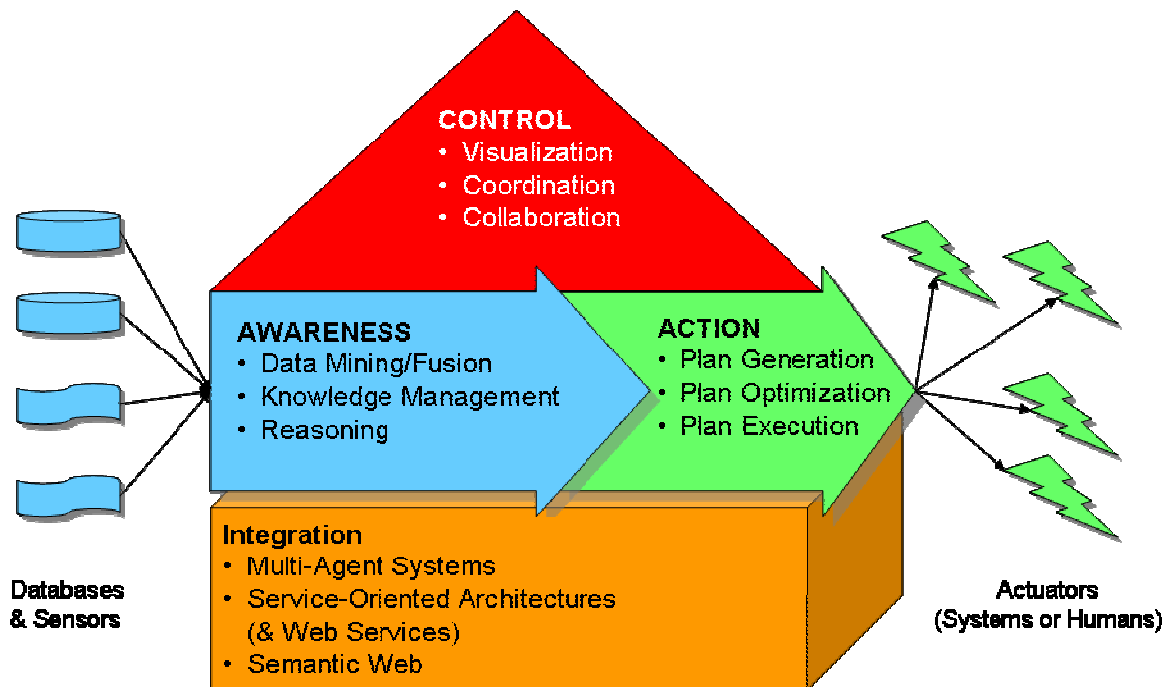
**Figure 4: The Initial IBWTP Approach by QLI**

As work on IBWTP progressed, it became clear that many of the technologies under development were suitable not only for the biological and chemical defense domain, but were equally applicable to a variety of other domains, especially emergency response in general, netcentric warfare, and intelligent enterprise solutions. With this in mind, QLI developed a more general architectural framework around which the core domain-independent technologies could be developed and integrated. This architectural framework was designated Awareness/Action/Control/Integration (AACI) as it integrates technologies and capabilities from Situational Awareness, Course of Action Planning and Execution, and Command and Control. Using the AACI framework, QLI could then represent and layer domain-specific functionality, knowledge representation, and models, to address specific application areas including but not limited to the original project goal of biological and chemical defense. In particular, the framework can also be used to effectively represent and deploy solutions addressing the Naval FORCENet requirements. The AACI framework is depicted in Figure 5. The approach fits naturally within the Observe, Orient, Decide, Act (OODA) loop, where Observe and Orient are in the domain of Awareness, Decide is in the domain of Control and Act is in the domain of Action.



**Figure 5: The Awareness/Action/Control/Integration Framework**

Figure 6 depicts the relevant software technology areas associated with the components of AACI. The bulk of the work performed by QLI within IBWTP focused on researching, developing, and enhancing technologies in these areas.



**Figure 6: The Awareness/Action/Control/Integration Technologies**

### 5.1.1. Awareness

*90% of the solution is in knowing the problem.*

Any organization, in particular, emergency response and management agencies, must be constantly aware of what's going on in order to rapidly detect (and respond to) potential and existing problems and threats. This involves monitoring many data and information sources – internal and external – to understand what has happened in the past, what is currently happening, and what might happen in the future.

With the advent of the Internet and information technology, there is an overabundance of available data – the problem is not that of collecting data, rather that of collecting the right data, how to make sense of this "InfoGlut" and, especially, how to identify relevant areas of concern. An important problem is to identify potential causes and indicators of problems based on observed effects.

A wide variety of data mining and data fusion techniques are available for extracting information out of data and knowledge out of information. Within IBWTP, QLI developed new approaches to data mining and data fusion based on probabilistic and causal reasoning. This allows for patterns in the effects of causes to be determined and thereby point to potential causes more rapidly than previously possible.

Furthermore, QLI developed technology within IBWTP to bring the power of different analysis techniques together, to cooperatively achieve better understanding than any one technique on its own. This enables data mining across disparate data sources (even potentially across organizational boundaries) without having to aggregate all data within a single data warehouse. This has advantages of flexibility, scalability, and allows organizations to maintain their individual autonomy.

An important component of Awareness is knowledge management – maintaining it as well as distributing it in a timely fashion to those who need to know. Targeted active dissemination of knowledge to relevant parties is crucial.

The QLI IBWTP team developed advanced mechanisms for enhancing situational awareness, such as coordinating data analysis, fusion, and reasoning systems to more rapidly and accurately alert IBWTP users to potentially harmful agents, such as chemical and bio-agents. These systems automatically share data, information resulting from data analysis, and goals. The systems are integrated by using multi-agent techniques in a grid-like network. This has the advantage of being able to integrate systems across geographical and organizational boundaries while preserving individual autonomy. Within IBWTP, the team:

- Provided automated discovery and access to comprehensive on-line data sources via multi-agent techniques.
- Investigated mechanisms for automatically analyzing results of data fusion and reasoning engines. This enables rapid triggering of alarms to decision makers as well as triggering further, more exhaustive and comprehensive, analyses.
- Developed mechanisms to combine the behavior of separate input models to provide expanded datasets for drawing better conclusions.

- Enabled cooperative resource, results, and goal sharing among disparate data fusion and reasoning engines, applications, and users.
- Developed multi-phased analysis mechanisms, whereby conclusions based on readily accessible easy-to-process data subsets trigger processing based on more exhaustive (and therefore more expensive-to-process) data sets.
- Analyzed methods of automatic information extraction from unstructured text-based information sources.
- Increased number and scope of data fusion and reasoning engines to encompass traditional data mining techniques and deductive inference mechanisms.
- Demonstrated the technologies in the areas of dispersion modeling, syndromic surveillance of diseases, and integration of weather data.

To accomplish this, QLI drew upon and enhanced core technologies in the areas of probabilistic and Bayesian reasoning, multi-agent techniques, and the semantic web.

### 5.1.2. Action

*“...The best-laid schemes o' mice an 'men  
Gang aft agley...”*

*- Robert Burns*

With Awareness comes knowledge and understanding of problems facing an organization. Once an organization is aware of any current problems it is facing along with their corresponding causes, it must take action to resolve those problems. Similarly, if an organization is aware of any potential future problems it might be facing, it needs to take action to prevent those problems. In both cases, plans must be formulated and executed to achieve the organization's goals. Within IBWTP, QLI developed advanced planning & reasoning technology to help organizations take action.

Plans may be generated using the following techniques:

- Scour a search space of possible plans, guided by heuristics, if available,
- Goal-directed reasoning using explicit domain-specific representation of preconditions and effects of tasks, and
- Establishment, modification, selection, constraint addition, and constraint relaxation by human planners drawing upon their experience and know-how.

However, it is crucial that plans adapt to an ever dynamic and evolving environment. It is not enough to plan based upon complete knowledge (or belief) about the current world state, as this knowledge may be inaccurate. Planning must also take different possible future world states into account (contingency planning). This takes not only the probability of events happening but also the importance of events into account. A trade-off must be made between the optimality of a plan and the time required to generate it (real-time planning). Finally, during execution of a plan, an organization must be able to rapidly react to any changes in the world state that affect the plan (on-demand planning).

Once a plan that must be performed is identified, it must be scheduled, resources (personnel and equipment) allocated, and a timeline set up for execution. Quantum Leap developed a flexible scheduling mechanism to accomplish this by a sophisticated model incorporating tasks to be done, available resources, and hard and soft constraints. The model is solved by Quantum Leaps patented Adaptive Optimization® Engine, which is a

flexible technology employing over thirty different problem solving techniques in a cooperative-competitive mechanism.

Within IBWTP, the team:

- Developed a framework for optimal placement of resources (such as sensors, medical facilities, UAVs) over a geographical region satisfying a number of coverage constraints and targets.
- Incorporated anytime algorithms to be able to provide the best available plans at any time of the execution phase and demonstrated these in the area of route planning in unknown environments.
- Developed techniques using probabilistic AI to enable real-time planning in uncertain environments, including probabilistic representations of past, present, and future world states as well as probabilistic representations of targeted goals.
- Incorporated advanced plan representation mechanisms to enable automated shared execution and coordination of plans across multiple autonomous actors.

To accomplish this, QLI drew upon and enhanced core technologies in the areas of optimization and problem solving, business process management, probabilistic reasoning, and multi-agent systems.

### **5.1.3. Control**

A fundamental component in any organization is the support of human decision makers in visualizing the information and knowledge gained by the awareness component, in analyzing and deciding on plans of action, and in directing and monitoring operations in real time. In large organizations, these decision makers are distributed across time and space. Furthermore, such operations involve the inclusion of many support personnel, such as technical specialists. Within IBWTP, the team:

- Developed an “integrated Knowledge Environment” (iKE) supporting seamless integration of information from many disparate sources into a common shared visualized information space.
- Enabled the integration and user-guided on-demand composition of both QLI-developed, legacy, and COTS applications within the iKE to more comprehensively evaluate and process information.
- Designed and constructed fixed and mobile “Interactive Knowledge Walls” supporting distributed interaction among users using iKE.
- Developed and integrated mechanisms into iKE supporting collaboration among distributed teams of users and autonomous agents, including clearboard, decision logging, chat, and video streaming.
- Demonstrated the technologies in the area of disaster management.

To accomplish this, QLI drew upon and enhanced core technologies in the areas of collaboration, computer supported cooperative work (CSCW), user modeling, tailorability, semantic web, and multi-agent systems.

### 5.1.4. Integration

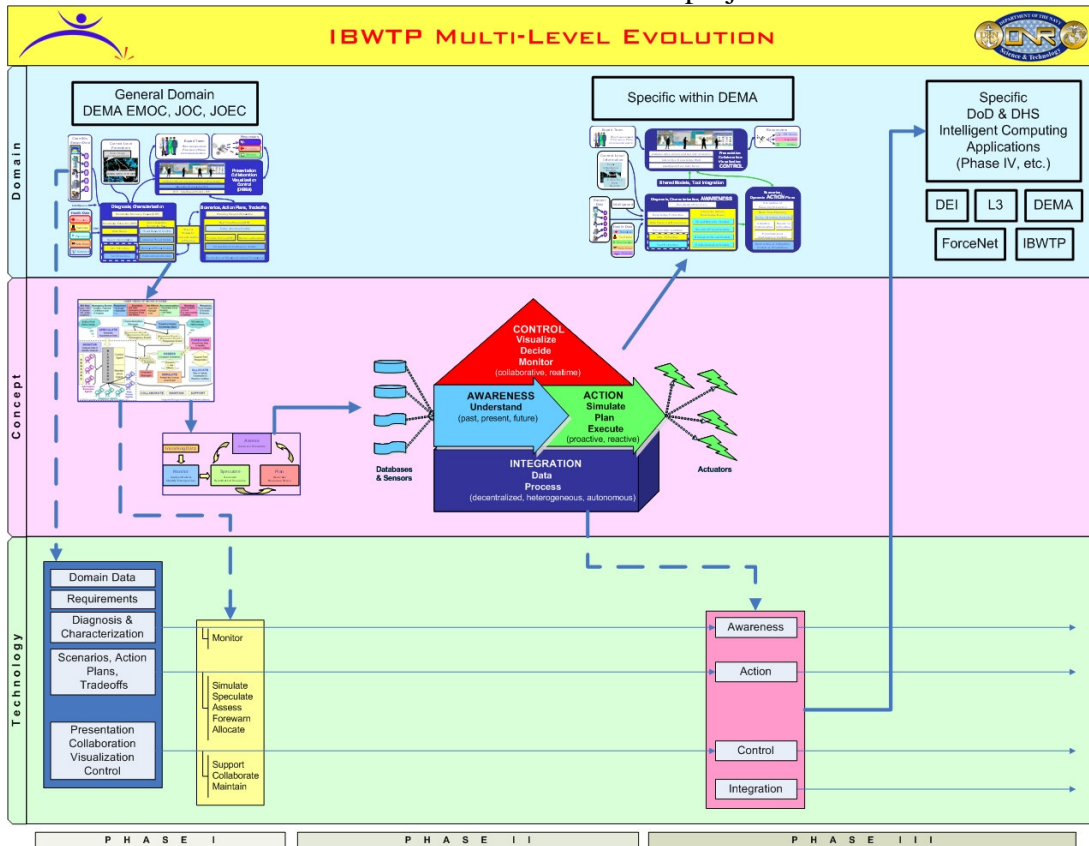
Underlying these technology components is the requirement to seamlessly and dynamically integrate a wide number of applications, systems, and human users into a cohesive, unified whole. Within IBWTP, QLI used multi-agent system technology, service oriented architectures, and the semantic web to develop a flexible platform (Multi-Agent Development Environment – MADE) supporting:

- Decentralized data management and process execution,
- Collaboration among systems and human users,
- Zero configuration networking and automated service discovery,
- Integration with legacy systems and applications, and
- Policy management for authorized access and execution of services.

The QLI team developed and deployed a decentralized platform for integration and coordination of heterogeneous applications and databases enabling rapid dynamic composition of applications required to support the Awareness, Action, and Control capabilities. It also enables the seamless composition of the capabilities with each other.

### 5.2. Evolution of IBWTP Conceptual Framework

Figure 7 provides an overview of the progression of the concept and high level architectural framework over the course of the IBWTP project.



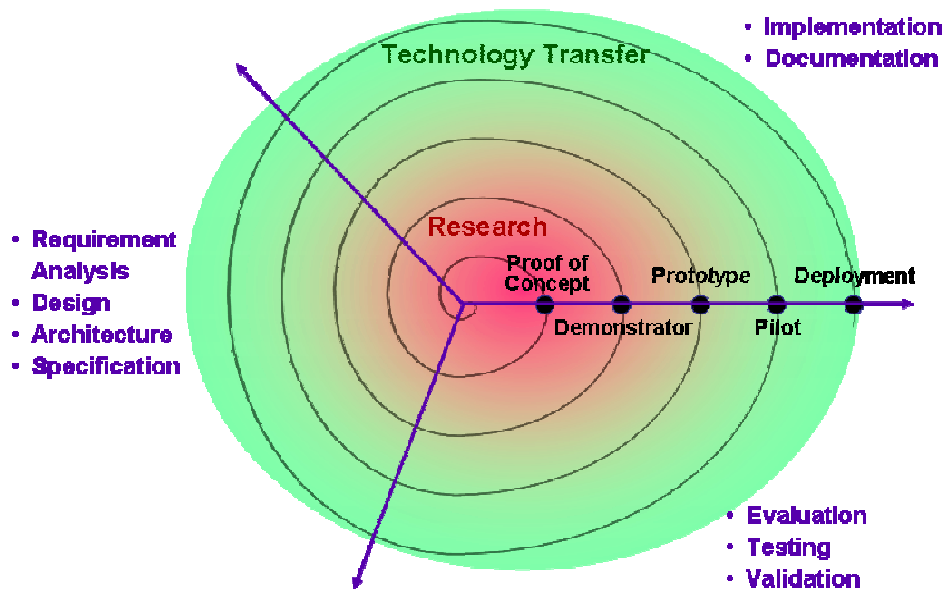
**Figure 7: Evolution of the AACI Conceptual Framework**

User requirements and specific targeted application areas were obtained from discussions with the Delaware Emergency Management Agency (DEMA) and Delaware Department of Health and Social Services (DHSS) as well as with representatives from US Department of Defense, Office of Naval Research and Department of Homeland Security.

### 5.3. Software Development Processes and Quality Assurance

#### 5.3.1. Technology Evolution

QLI's goal in IBWTP was to not only perform research into novel technologies supporting the aims of the project but to also progressively evaluate and develop particularly promising technology into mature software that can be embedded in deployed solutions. To this end, QLI adopted a spiral R&D development with research performed by software scientists creating the core technologies followed by technology transfer performed by software engineers and developers developing robust and maintainable implementation of the technology. The technology evolved in the following stages (cf. Figure 8):



**Figure 8: Spiral Technology Development from Concept to Deployment**

- Proof of Concept (Research)
  - Basic research
  - Concept development
- Demonstrator (Research)
  - Simulated data
  - Simulated environment
- Prototype (Research & Technology Transfer)
  - Real world data
  - Simulated/controlled environment



- Pilot (Technology Transfer)
  - Integrated into real world environment
- Deployment (Technology Transfer)
  - Fully deployed

Software Scientists were primarily responsible for the Proof of Concept and Demonstrator phases, and Software Engineers were primarily responsible for the Pilot and Deployment phases. The transition from Research to Technology Transfer usually happens at the Prototype phase, where Scientists hand off the technology to the Engineers.

Throughout the phases of the technology, QLI used a common software development and maintenance environment, called the Quantum Leap Uber Build System (QLUBS). QLI developed a variety of mechanisms to monitor the progression of technology development across teams in order to ensure that the technologies could be used by other teams and integrated with the other technologies under development.

The following describes the approach taken in the development of the technologies and maturation of the software. All software was developed modularly in the JAVA™ software programming language, using the Eclipse Integrated Development Environment. Each technology is represented in one or more Java “packages.” All packages are registered with a central archive called the nexus that has an easy-to-use web-based interface to access virtually all information about the package.

In order to ensure maximum reusability, any package can be dependent on any other package in the nexus or a third party library. The nexus provides a variety of reports about the software code, such as Lines of Code, results of Unit Testing, adherence to coding standards, etc.

Packages are managed by the QLUBS build system that automatically

- Manages dependencies among packages
- Stores code in a version control system
- Launches and loges compilation of newly generated code
- Performs automated testing
- Generates reports
  - JUnit test coverage
  - Checkstyle
  - Lines of Code Count
  - Javadoc

QLUBS makes use of a number of open source software management tools, such as MAVEN, ANT, Damage Control, CVS, JUnit, etc. The design of the system is such that new software management components can be updated and integrated into the overall system.

### 5.3.2. Maturity Levels

When first developing software for research purposes, QLI did not want to spend an excessive amount of time ensuring the quality of the software. However, as the software progressed along the spiral development path, it became increasingly necessary to ensure its quality. QLI developed its own Software Quality Maturity Level process with the following goals in mind:

Table 1: Goals for Software Maturity	
Goal	Justification
<b>Research</b>	Pioneering ideas are most likely to be discovered when researchers can work in a flexible environment. Restraints placed on code that is exploring ideas and concepts should be minimal.
<b>Low Overhead</b>	Developers should not have to spend more than 5% of their time fulfilling process requirements. Creation and maintenance of tests and documentation of code is not considered overhead.
<b>Uniformity</b>	Software projects should follow standards in order to allow easier understanding of the code.
<b>Communication</b>	All levels of the organization should be able to easily get up to date information about a software package that is relevant to them.
<b>Bullet Proof</b>	When the software is deployed there should be no doubt that it will work.
<b>Cost Effective</b>	The commitment of resources to software production should be as cost-effective as possible.

The Maturity Level process defines four levels of software maturity as well as the steps required for software to advance from one level to the next. Each package has a Maturity Level (ML). To advance in maturity level, the package must pass an audit performed by the Quality Assurance lead or authorized representative. The requirements and how the software performs against the requirements can be easily seen based on the reports generated by the build system.

#### *Quality Maturity Level 1: Explorable*

At this level, developers have free reign to experiment and try out ideas. Packages should leave this state when something useful has been developed. While there are no specific standards for packages at this level, developers are strongly encouraged to follow published standards to ease the transition to higher levels.

#### *Requirements*

- Package must be registered with the nexus
  - The Nexus registers the package with various back-end infrastructure components, and organizes all software in the company.

#### *Permissions*

- Package may have IBWTP resources committed to it.  
By ensuring that only registered packages are allowed to be worked on, we guarantee that the development organization is aware of what its developers are doing.
- Package may be used by other packages within its originating subtask.  
When a package is initially created, it should be able to freely interact with other packages that are part of the same development effort. However for the package to be used outside of its first subtask, it must be promoted to Quality Level 2.

### *Quality Maturity Level 2: Sharing*

At this level a software package is ready to be used throughout the program. The sharing of software packages allows for code to be combined in ways that may not have been originally envisioned by the creator. This interaction provides valuable feedback on how to increase the utility of the packages.

#### *Requirements*

- Must produce a dedicated entry (web page) in the nexus  
Creates a well known repository for information about packages, and provides a standard expectation for documentation about the package. Documentation about a package must be easily accessible in order for the package to be successfully used.
- Package web page must contain example code and usage documentation  
Packages can only be successfully shared if their usage is clearly documented. Usage documentation should highlight the primary API classes, and direct the reader to their javadoc. It should also detail, if appropriate, how the package can be invoked via the command line, as a web-service or through other inter-process communication mechanisms.
- Less than 1 checkstyle error per 5 source statements  
Checkstyle ensures that the coding standard is followed.
- Less than 1 PMD error per 5 source statements  
PMD exposes poor coding practices in java code.
- Less than 20% duplicate code  
Duplication increases the effort involved in maintaining code.
- Unit test must cover more than 50% the code  
Unit testing is a fast reliable way to ensure that a package maintains its functionality.
- Package must be maintainable  
Having a usable and consistent maintenance infrastructure is necessary for making package modifications and improvements as simple as possible.

#### *Permissions*

- Any packages may be dependent on this package  
Enforcing a minimal set of quality on a package before it can be shared will increase the likelihood that developers can successfully use the package.

### *Quality Maturity Level 3: Usable*

At this level a software package is ready to be used in prototypes and internal applications. The code should be well documented, and maintainable by someone other than the original authors. Standards should be strictly followed, and unit testing should be fairly rigorous.

#### *Requirements*

- Less than 1 checkstyle error per 25 source statements  
Checkstyle ensures that the coding standard is followed.
- Less than 1 PMD error per 25 source statements  
PMD exposes poor coding practices in java code.
- Less than 5% duplicate code  
Duplication increases the effort involved in maintaining code.
- Unit tests must cover more than 75% of the code  
Unit testing is a fast reliable way to ensure that a package maintains it's functionality.
- Passes Unit Test Review  
Unit tests prove that code is operating as the author intended, and allow modifiers to quickly access whether changes had unintended side effects.
- Passes Javadoc Review  
Javadoc is the primary means of communicating how to use code to other developers.

#### *Permissions*

- May be released externally.  
Code at this level is ready to be given to external partners for evaluation or maintenance.
- May be used in internal tools and application.  
Using internally developed code on internal application (Eating your own dogfood) provides valuable information on usability, as well as provides a good environment for discovering defects.

### *Quality Maturity Level 4: Reliable*

At this point the code is as good as it is going to get. The code is fully documented, and unit tested. Additionally, the code is well logged, enforces software contracts, and clearly communicates errors. Software at this level is maintainable, reliable, and configurable.

### Requirements

- Less than 1 checkstyle error per 100 source statements  
Checkstyle ensures that the coding standard is followed.
- Less than 1 PMD error per 100 source statements  
PMD exposes poor coding practices in java code.
- Less than 1% duplicate code  
Duplication increases the effort involved in maintaining code.
- Complies with productization standards  
Includes logging, error handling, internationalization.
- Unit tests must cover more than 95% of the code  
Unit testing is a fast and reliable way to ensure that a package maintains its functionality.
- Passes Unit Test Review  
Unit testing is a fast and reliable way to ensure that a package maintains its functionality.
- Passes Javadoc Review  
Javadoc is the primary means of communicating code's functionality to other developers.

### Permissions

- May be deployed in product and field environments.  
Code at the level represents the best we have to offer.

Table 2 provides an overview of the requirements of the various Maturity Levels.

<b>Table 2: Requirements for SW to achieve Quality Maturity Levels</b>				
<b>Metric</b>	<b>ML1</b>	<b>ML2</b>	<b>ML3</b>	<b>ML4</b>
<b>Checkstyle Error</b>	No requirement	< 1 out of 5 statements	< 1 out of 25 statements	< 1 out of 100 statements
<b>PMD Error</b>	No requirement	< 1 out of 5 statements	< 1 out of 25 statements	< 1 out of 100 statements
<b>Duplicate Code</b>	No requirement	< 20%	< 5%	< 1%
<b>Test Line Coverage</b>	No requirement	> 50%	> 75%	> 95%
<b>Unit Test/Java Doc Review</b>	No requirement	No requirement	Yes	Yes, More Stringent

## 6. Results and Discussion

This section describes the results of IBWTP. Figures 9 and 10 show the evolution of the technologies developed over the duration of the project.

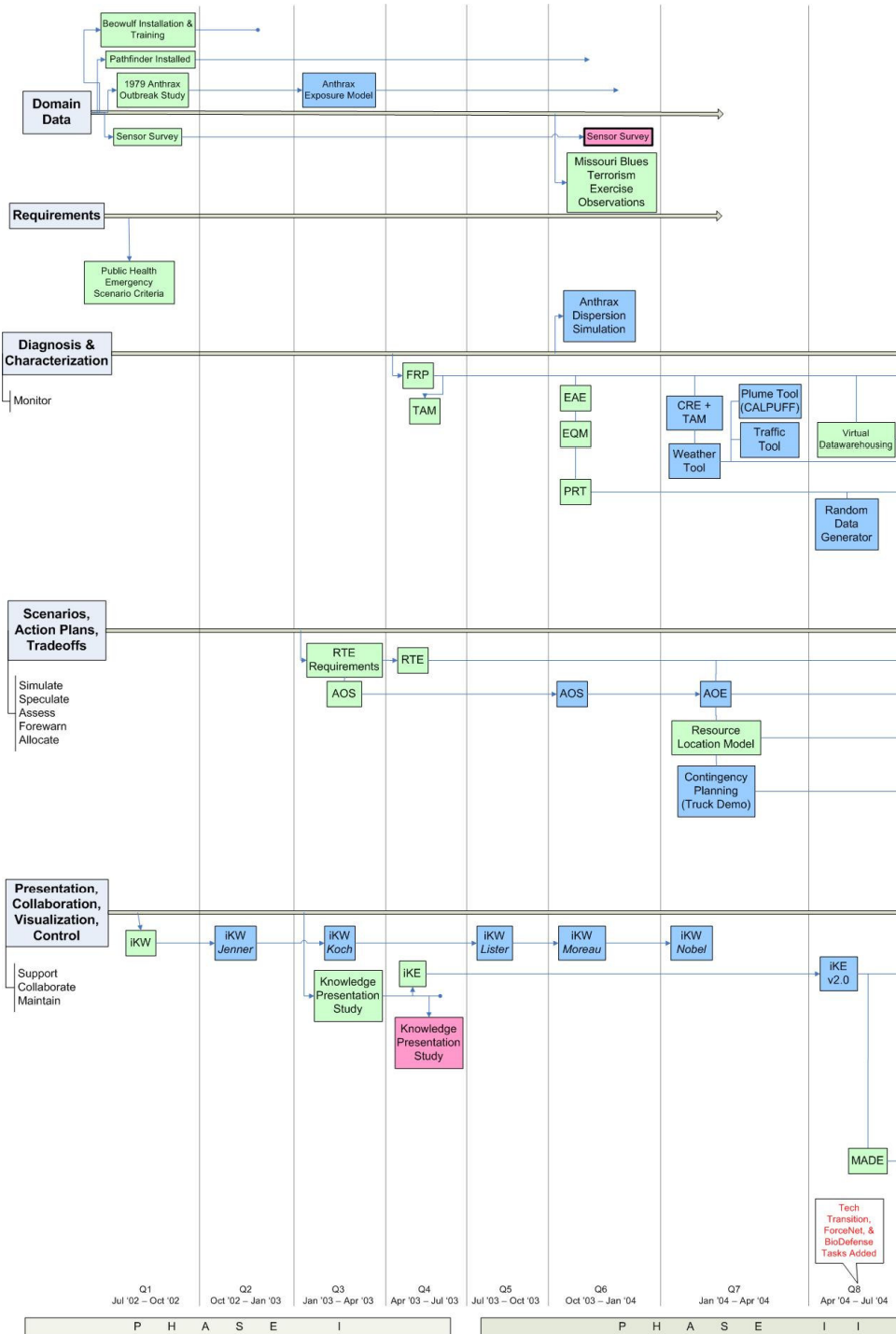


Figure 9: Evolution of IBWTP Technology Solutions

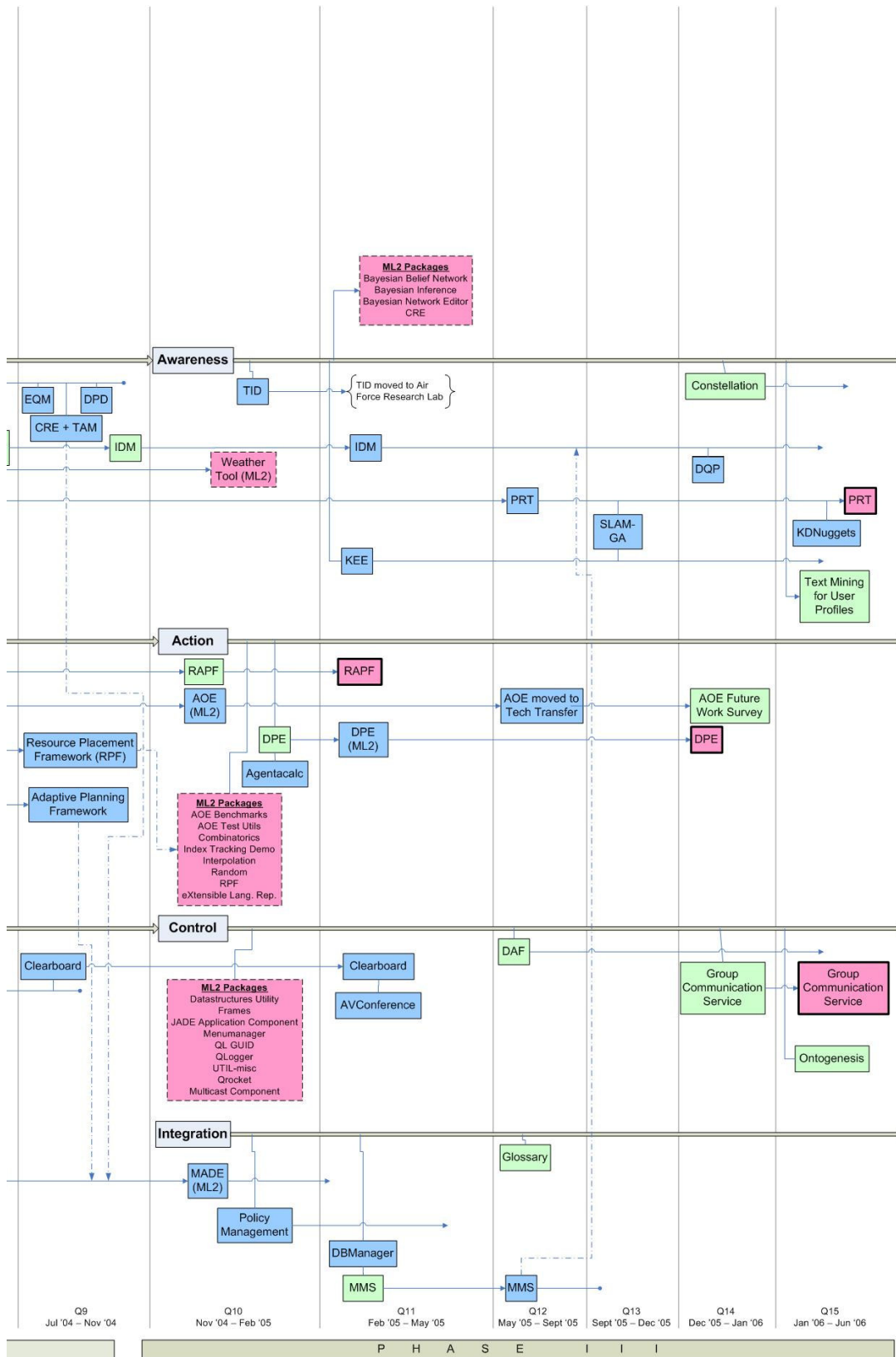


Figure 10: Evolution of IBWTP Technology Solutions (cont.)

In accordance with the approach outlined in Section 5, QLI developed the IBWTP technologies, also called Integrated Biological and Chemical Warfare Defense (IBCWD) software, according to the following Work Breakdown Structure.

- **Task 1: Program Management**
- **Task 2: Development of ‘Awareness’ Capabilities**  
Develop, demonstrate, and enhance technologies supporting early detection of potential or existing biochemical attacks
- **Task 3: Development of ‘Action’ Capabilities**  
Develop, demonstrate, and enhance technologies supporting rapid response to identified situations of a potential or given biochemical attack
- **Task 4: Development of ‘Control’ Capabilities**  
Develop, demonstrate, and enhance technologies supporting an integrated command and control environment enabling same-time, different-place support of information visualization, application linking, and decision support to decision makers assessing and responding to potential or given biochemical attacks
- **Task 5: Development of ‘Integration’ Capabilities**  
Develop, demonstrate, and enhance technologies supporting dynamic integration of and collaboration among applications and humans in a distributed network
- **Task 6: Domain Requirements and Application**  
Evaluate and demonstrate applicability of technologies developed in Tasks 2-4 to application areas of biological and chemical warfare, emergency management, and Navy force transformation.
- **Task 7: Technology Transfer**  
Progress relevant and promising technologies developed in Tasks 2-4 through the software maturity lifecycle for further prototype and pilot development.

The following sections outline the results and developed technologies accomplished in each of these tasks.

## **6.1. Awareness Capabilities**

Work in Task 2, ‘Development of Awareness Capabilities’ concentrated on developing, demonstrating, and enhancing technologies supporting early detection of potential or existing biochemical attacks.

The effort was primarily composed of two parts:

- Data Management & Data Fusion
  - Intelligent Data Management (IDM)
  - Anthrax Dispersion Demonstrator
  - Weather Tool Demonstrator
  - Environmental Quality Monitoring
  - Event Attendance Estimator
  - Dynamic Population Distribution
  - Targeted Information Dissemination (TID)



- Data Mining through Probabilistic Reasoning
  - Probabilistic Reasoning Toolkit (PRT)
  - Causal Reasoning Engine (CRE)
  - Threat Assessment Module (TAM) Demonstrator
  - Dynamic Distributed Data Mining (D3M) Concept

The rest of this section outlines the primary work accomplished in the scope of this task.

### 6.1.1. Intelligent Data Management (IDM)

The ever increasing amount of data available from a wide variety of sources makes it impractical, if not impossible, to collect data into a single data repository for later retrieval and analysis. However, at the same time, the ever-expanding clusters of data on web sites also increase the need to have combined views of data from these heterogeneous sources. This requires the technology to retrieve, extract, integrate and compose data from several distributed, heterogeneous data sources.

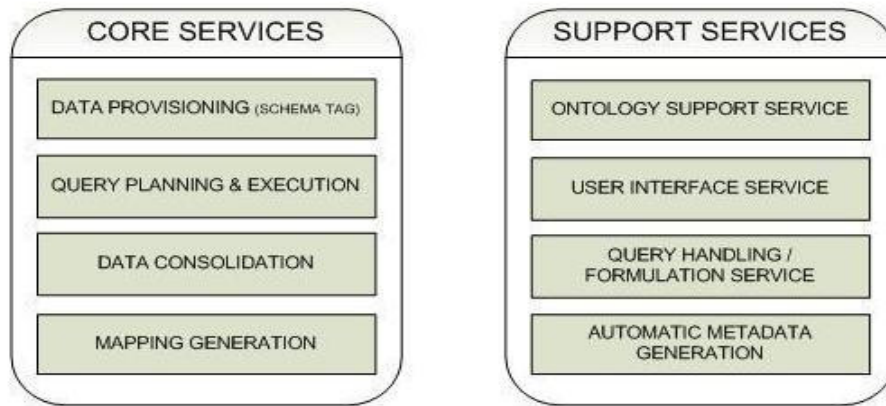
QLI developed the Intelligent Data Management (IDM) framework in order to enable uniform and easy access to disparate and heterogeneous sources of data that may be distributed across networks. The framework imparts a layer of abstraction to the individual data access mechanisms of the data sources. The IDM framework draws upon a multi-agent paradigm for realizing a “data access abstraction layer” resulting in a dynamic, flexible environment consisting of data sources that are registered and accessed as standard agent services. The IDM provides seamless integration of new sources of data as and when they become available or on demand into the existing data environment. IDM can be dynamically updated and evolve in terms of the different databases and data sources that it can handle. The IDM framework relies upon metadata for handling access requests for structured (relational) as well as semi-structured and unstructured data. QLI designed IDM to employ efficient techniques for extracting/imparting and exposing the metadata of the data sources to the user. The asynchronous design of IDM’s keyword based data access mechanism is highly user-driven, in that it selects data based upon the user’s selection and constraints on metadata parts. QLI also designed IDM to utilize distributed caching mechanism to further enhance the efficiency of the framework. In future projects, QLI will incorporate semantic analysis techniques in IDM to impart meaningful and useful relationships among metadata and hence data, and utilize the semantic network of metadata to enable keyword based data querying capability.

QLI developed the Intelligent Data Management framework in order to enable the virtual integration of disparate data sources into a “Virtual Data Environment” (VDE) thereby consolidating their data at runtime to enable retrieval of combined/ fused data from the data sources.

QLI designed and implemented the IDM framework as a multi-agent system with agents providing a variety of system functionalities as **services** (cf. Figure 11). These services include:

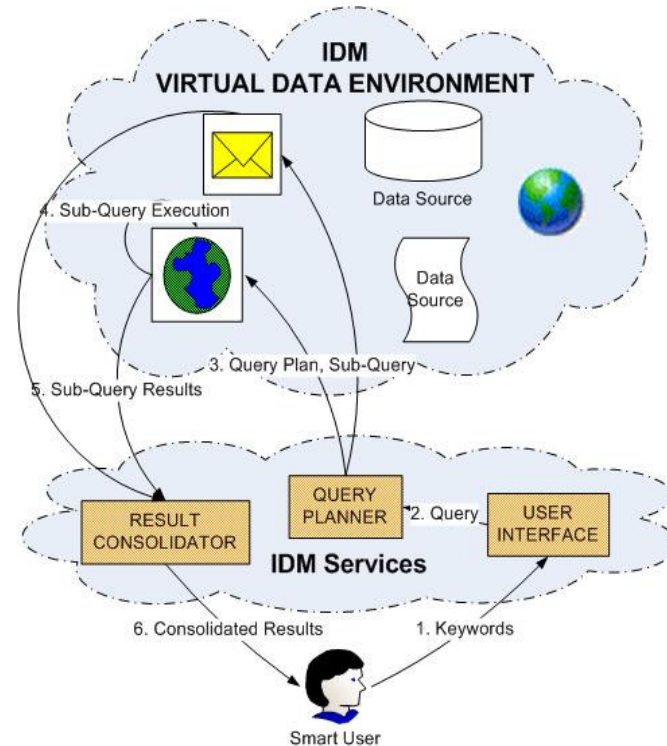
- **Data Consolidation Service** (DC Service): This service combines two or more sets or components of data together in a manner consistent with domain specific rules.

- **Data Provisioning Service (DP Service):** This service is typically provided by data sources to provide access to their data. The description of Data Provisioning Service includes the type of data being provided and how this service can be invoked among other information.
- **Metadata Generator Service (MG Service) (Optional):** The service provides for automatic generation of semantically rich metadata (ontologies) for structured data. This service is closely tied with the software implementation of the data source, e.g., such service for MySQL relational data source is specific to generating ontologies from database, table, and column information.



**Figure 11: Data Management Services provided in IDM Framework**

- **Ontology Support Service (OS Service):** This service is responsible for maintaining a repository of the various ontologies being supported in the system by data sources (Data Provisioning Service providers). The service maintains a list of mappings (and mapping points – resources) among these ontologies.
- **Query Handling Service (QH Service):** This service creates a machine processable version of the user input along with optional formatting and/ or completion. This formal, machine processable representation in the system is called a Query (Description).
- **Query Planning & Execution Service (QP Service):** The Query Planning Service provides for stipulating, distributing, and executing a plan to solve the given query. Query plans consist of sub-queries that are formed by resolving the given query to smaller sections that are sent to individual data source services for their results. The query plan also has information about the Data Consolidation Service providers that combine the results of the sub-queries as per domain specific rules.
- **User Interface Service (UI Service):** This service provides for handling users' interaction with the system including bringing up a UI, responding to the users' input and delivering any responses back to the user.



**Figure 12: Dataflow among IDM Components**

### *System Components*

- Service providing agents
- Data Source: A data source represents and describes any piece of software that maintains real world data. Examples of a data source include relational databases such as MySQL, applications or web-pages that maintain transient data, file-systems that maintain data in files, and messaging systems that hold data in messages/ emails.

### *Agent Deployment Framework*

QLI used the Multi-Agent Development Environment (MADE) used for creating and deploying agents providing the IDM services. The agents draw upon the deployment environment's default mechanisms for agent discovery, communication, cooperation, and easy system configuration for service deployment and usage. The agent behaviors simulating particular IDM services conform to the standards of those services.

### *Data Access/ Query Interface*

The IDM design includes two modes of data access: Data source selection-based and keyword-based. In the first mode, the system is aware of the exact data sources from which the user requires data. QLI designed this option to allow the user to add constraints to specific metadata components of the data sources. In the keyword-based mode, the user does not know a priori about the available data sources and is initially prompted to input keywords relevant to the data they are looking for. In this case, the system is designed to perform word similarity analysis and eventually semantic analysis of the

metadata of available data sources in the system to select the most relevant data sources to fetch data from as results.

The design of the second mode allows the user to input keywords at two levels:

- **Keywords indicating the name of or the ontology of the candidate data.** In this case the user can also specify the actual Unique Resource Identifier (URI) of the ontology. For this keyword the system components try to match the keyword with names/ URIs of the existing ontologies. The user can interact with the system to enter keywords for two or more ontologies.
- **Keywords indicating the name of a particular attribute (resource) that must be defined in the ontology of the candidate data.** In this case, the user can also specify numbers or words as values (-range) constraints on the attributes.

In the scope of similarity analysis process, the keywords entered by the user are matched with the data-source ontologies being supported in the system. As an initial attempt, the first matched data sources (ontologies) will be selected. The data sources supporting these ontologies will be queried for a predefined unit of their data as specified by the user where prompted. Examples of predefined unit are 10 units of data or data in the last 12 hours. If more than one keyword for ontology names are specified and specific ontologies corresponding to those keywords are supported in the system the system is designed to discover common (semantically equivalent) resource(s) defined in the ontologies and attempts to consolidate the data based upon the common resource as desired by the user.

### *Data Structures*

The IDM framework uses the following data structures to support the various data services.

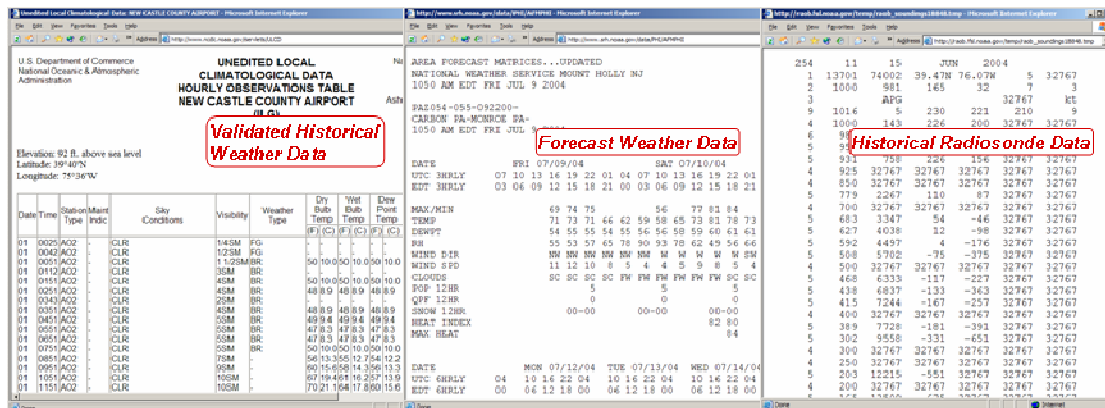
- **Data Source Profile:** Profile of a particular data source including its representative factors along with certain quality related features as provided by the user at the time of integrating the source into the system. The implementation of DSP includes the following features:
  - Name
  - Type
  - Location
  - Metadata
  - Data Type
  - Data Format
  - Quality
  - Frequency
  - Supported Ontologies
- **Processing Technique Profile:** The data about a processing technique including its capability, input/output parameters, technique setup data, etc. as provided by the user at the time of adding the technique into the system. Not designed yet.

- **Metadata:** Metadata includes specific concepts for which the data source has data. It is a part of Data Source Profile and is also used in semantic analysis.
  - Entities
  - Entity Data Type
  - Description
  - Entity Relations
  - Properties
    - Property Data Type
    - Constraints
  - Format
  - Data
- **Data Source Configuration:** The collection of data sources and corresponding data sets as selected by the user at a time.
- **Processing Technique Configuration:** The collection of processing techniques and corresponding profile parameters-values used for processing a set of data.

### 6.1.2. Weather Tool Demonstrator

As an initial demonstrator of IDM functionality, QLI developed the Weather Tool that provides complete weather data between arbitrary dates at a given update frequency as collected and consolidated across several weather data sources. The weather tool collects and/or makes available historical, forecast, and upper air data to other applications (e.g. the Dispersion Model and the Sensor Placement Tool) or to end users. For specific update frequency requirements, weather tool also interpolates over the available weather data with certain assumptions in place.

The sources have weather data in different formats, with different update frequencies and of different quality (cf. Figure 13). In the demonstrator, the Weather Tool has a data scraper each for collecting forecast data from the NOAA forecast web site (<http://www.srh.noaa.gov/data/PHI/AFMPHI>), historical validated data from NOAA's NCDC website (<http://ncdc.noaa.gov>) and radiosonde data from NCDC and FSL's website (<http://raob.fsl.noaa.gov/>).



**Figure 13: Different Weather Data Sources.**

The Weather Tool provides a unified interface to the user presenting data gathered from different sources, as seen in Figure 14.

WEATHER TOOL					
	dateTime	skyCondition	temperature	windDirection	windSpeed
New Castle County	03/25/04 at 3:00:00 AM	Cloudy	44.0	S	10.0
	03/25/04 at 6:00:00 AM	Mostly Cloudy	48.0	S	6.0
	03/25/04 at 9:00:00 AM	Mostly Cloudy	52.0	SW	9.0
Kent County	03/25/04 at 3:00:00 AM	Mostly Cloudy	43.0	S	10.0
	03/25/04 at 6:00:00 AM	Mostly Cloudy	48.0	S	8.0
	03/25/04 at 9:00:00 AM	Mostly Cloudy	52.0	SW	9.0
Sussex Inland	03/25/04 at 3:00:00 AM	Mostly Cloudy	43.0	S	10.0
	03/25/04 at 6:00:00 AM	Mostly Cloudy	47.0	S	8.0
	03/25/04 at 9:00:00 AM	Mostly Cloudy	52.0	SW	10.0
Sussex Coastal	03/25/04 at 3:00:00 AM	Mostly Cloudy	43.0	S	14.0
	03/25/04 at 6:00:00 AM	Mostly Cloudy	46.0	S	10.0
	03/25/04 at 9:00:00 AM	Mostly Cloudy	49.0	SW	13.0

Pick time range for data to show

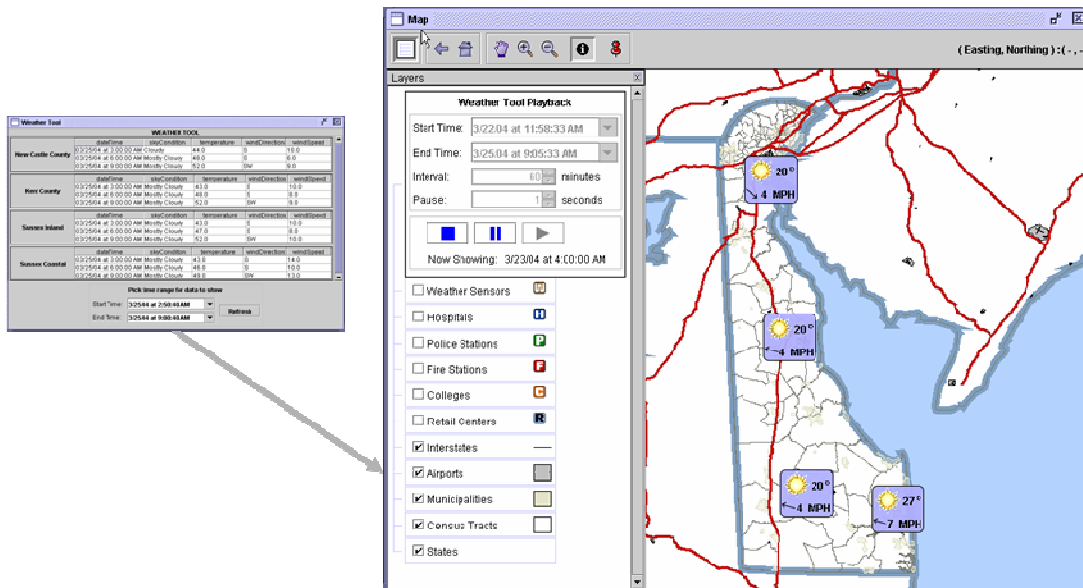
Start Time: 3/25/04 at 2:50:40 AM

End Time: 3/25/04 at 9:00:40 AM

Refresh

**Figure 14: User's Data Query Interface to Weather Tool.**

Furthermore, the Weather Tool offers a sophisticated API to other applications, allowing them to query for data. Figure 15 shows the integrated Knowledge Environment's (iKE) map display connecting to the Weather Tool. Notice, the iKE provides a separate user interface control in the legend, for the user to specify the desired date ranges. This UI is independently maintained by iKE from the UI maintained by the Weather Tool itself.



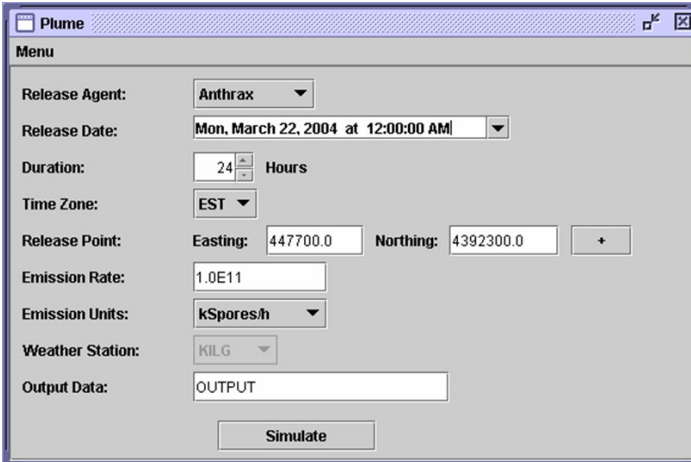
**Figure 15: Application Interface to Weather Tool.**

### 6.1.3. Dispersion Model Demonstrator

QLI surveyed a number of particle dispersion modeling tools prevalent in the area of emergency command and control including CALPUFF, HYSPLIT, HYPACT, HPAC, and ICE-AERMOD. QLI selected CALPUFF for simulating sample plumes situations for supporting detection, visualization, and course of action planning. CALPUFF is a Gaussian puff dispersion modeling system (with complex terrain algorithms, plume fumigation, etc) that also includes a meteorological modeling package and a set of post processing programs.

QLI developed a software wrapper to CALPUFF both at the input and at the output ends in order to trigger generation of dispersion models based on given input criteria and display the ensuing representation to other applications, such as map and sensor location. The power lies in that different tools can be integrated into the overall environment, allowing the user to choose the tool, rather than having the choice be dictated by the capabilities of the underlying IT system.

On the input side, QLI provided a wrapper to easily process weather data from relevant sources and prepare them for input to CALPUFF (cf. Figure 16). This work was done in conjunction with QLI's subcontractor, the Environmental and Occupational Health Sciences Institute (EOHSI). Input processing integrated current, historical, and radiosonde data using the MENTOR/SHEDS program and bringing them together in a single input file for use by CALPUFF. QLI developed new software modules for estimating mixing heights and preparing meteorological inputs from unprocessed monitor data for use in short term air quality model applications. In particular, the modules are robust enough to handle missing information in unprocessed weather data, especially radiosonde data. Change in the formats of the data on the website need not be handled by the modules. Also, the modules must be documented at source code level.

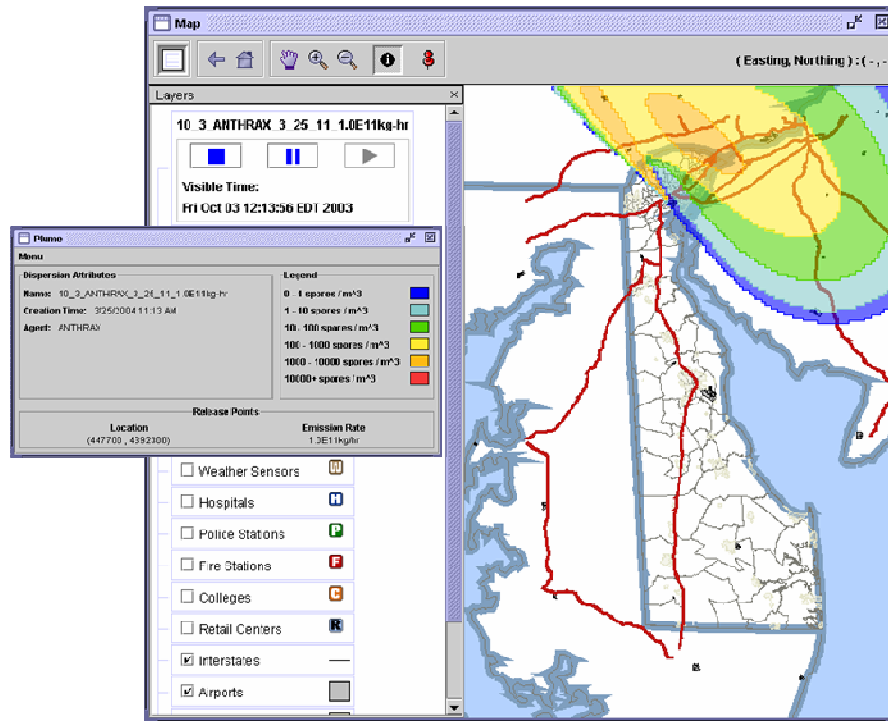


The screenshot shows a software window titled "Plume" with a "Menu" bar. The interface contains several input fields and dropdown menus for configuring a dispersion model simulation. The fields are: "Release Agent" (set to "Anthrax"), "Release Date" (set to "Mon, March 22, 2004 at 12:00:00 AM"), "Duration" (set to "24" hours), "Time Zone" (set to "EST"), "Release Point" (with "Easting" at 447700.0 and "Northing" at 4392300.0), "Emission Rate" (set to 1.0E11), "Emission Units" (set to "kSpores/h"), "Weather Station" (set to "KILG"), and "Output Data" (set to "OUTPUT"). A "Simulate" button is located at the bottom of the window.

**Figure 16: User Input to Dispersion Model**

On the output side, QLI developed software modules for processing the textual plume output from CALPUFF into a format that is more efficient for showing the plume data on a map application (cf. Figure 17). We also developed software modules for converting our internal efficient format into GIS acceptable shape files that can also be used by commercial map simulation software for showing the plumes produced by CALPUFF.





**Figure 17: Application Output from Dispersion Model to iKE**

#### **6.1.4. Environmental Quality Monitoring (EQM) Demonstrator**

The EQM analyses data from Delaware’s Department of Natural Resources and Environmental Control (DNREC) to help identify syndromes found in patient data that are linked to environmental pollution. Other than monitoring the six criteria pollutants (ozone, particulates, carbon monoxide, nitrogen dioxide, sulfur dioxide and lead) DNREC also monitors the levels of pollen count from trees, grasses, weeds and molds.

Health symptoms caused by these pollutants in the population are mentioned on the EPA’s website. DNREC data is analyzed to predict the health problems that might be mistaken for chemical or biological threat.

#### **6.1.5. Event Attendance Estimator (EAE) Demonstrator**

The Event Attendance Estimator (EAE) estimates the number of people attending an event based upon its description. Everyday events, as posted on websites such as [www.DelawareOnline.com](http://www.DelawareOnline.com), are downloaded by running a script. The title and descriptions of these events are scanned to check for the presence of the location of the event by comparing them against a list of venues as maintained by an expert. For events whose descriptions have a 100 percent match for a venue in the list, the attendance is the capacity of the venue.

The estimator tries to estimate the attendance for other events by determining the type of event - Indoor vs. Outdoor, Entertainment vs. Academic vs. Sports vs. Religious event - based upon the description of the event.



### 6.1.6. Dynamic Population Distribution (DPD) Demonstrator

Dynamic population distribution (DPD) is a system which estimates the changes of the population distribution in census area over time. Based on the fact that people move around for various events and reasons, QLI developed an event-driven model providing a generic architecture for dynamically estimating the population in an area at any given time (cf. Figure 18). The initial state of the population distribution is set to the static census data. The dynamic population distribution is then calculated considering the different events happening during the time.

Population estimation plays an important role in the population census work. As the census is a large task involving a lot of labor and resources, it is conducted only every 10 years by the U.S Census Bureau. In the interim, population is only estimated approximately. Many algorithms have been proposed in the literature to estimate population changing over time. However, these algorithms only work well in estimating the population distribution over time scales of years or months. No prior method has been developed to estimate the population motion during a short amount of time, such as population changes during a weekday from morning to evening. However, this is particularly critical for emergency response planning. QLI developed a generic event-driven population updating model, able to incorporate different events, such as one time events (concerts, sporting events), daily events (commute, school in session), and yearly or seasonal events (holiday shopping, summer weekend at the beaches).

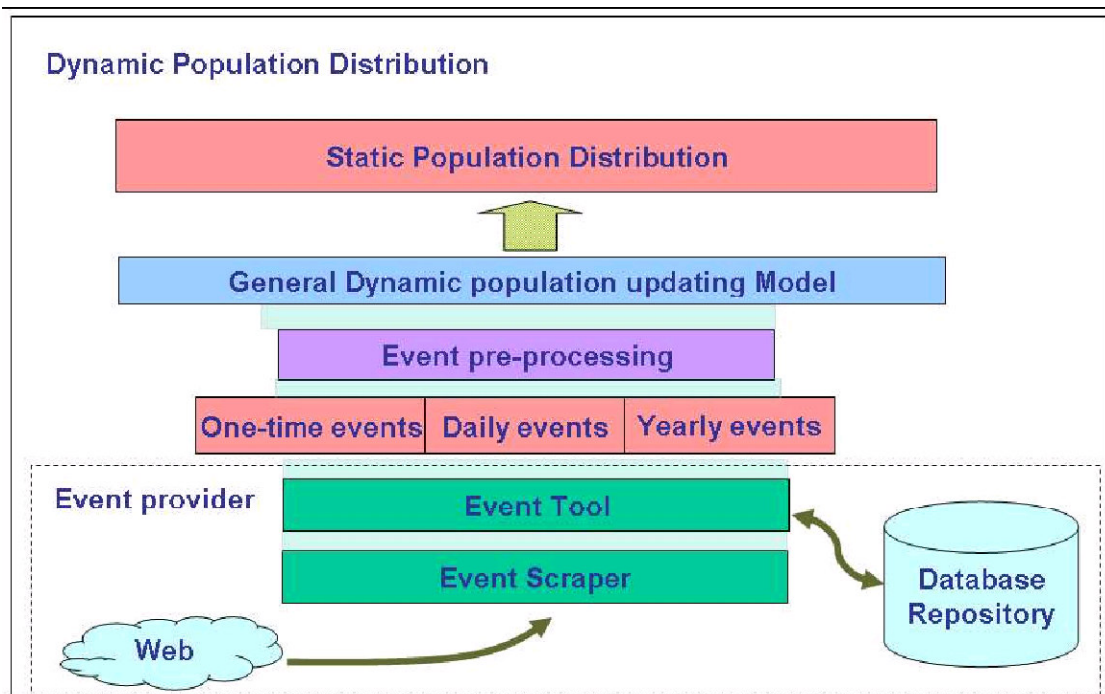


Figure 18: Dynamic Population Distribution System Overview

### **6.1.7. Probabilistic Reasoning Toolkit (PRT)**

An important component of Awareness is the ability to represent and reason about uncertainty. A technology that has become increasingly viable is Bayesian inferencing.

A Bayesian Belief Network (BBN) is a collection of variables and causal connection between the variables. A variable can be thought of as an observable object or event that can take on at least two different values, or states. A causal connection means that variable “A” directly affects variable “B”. A BBN stores probabilities of each variable’s states given the parents of that variable. A BBN network is depicted by a graph of nodes and edges, where each node denotes a variable and each edge denotes causality from the node the edge originates from to the node the edge points to.

Bayesian networks complement certain aspects of the data fusion process. Bayesian networks provide a certain sense of data fusion simply because they allow many kinds of data to be considered while still being able to “reason” with the diverse range of data sources. For example, time of day, season of year, pollen count, anthrax threat, and event attendance can be handled and reasoned about all in the same network.

Bayesian Belief Networks are primarily used for prediction and diagnostic inference. Within the scope of IBWTP, QLI developed it’s own BBN representation and inferencing mechanism, called the Probabilistic Reasoning Toolkit (PRT). The PRT allows users to quickly construct and explore Bayesian Networks, and to inject test cases into those networks. The PRT editor presents a user with an intuitive graphical user interface and an efficient method of drawing Bayesian networks and entering probabilities. The editor is tied in with both the inference algorithm and a synthetic data generator.

The inference algorithm implemented in the PRT is the junction tree algorithm described by Lauritzen and Spiegelhalter. This is an exact-inference algorithm that is common in commercial use. The junction tree algorithm is 100% accurate but relatively slow for very complex networks. This algorithm allows users to instantiate “evidence” nodes, or nodes that the user knows what the values are, and have the algorithm produce the posterior probabilities for query nodes. QLI’s implementation has been tested against industry-standard products, such as Hugin, and has been demonstrated to provide a correct implementation. Beyond the typical use of BBNs, the PRT is especially useful for quickly encoding and visually manipulating logical nodes (ANDs, ORs) that may provide a more transparent encoding of some forms of domain expertise.

One of the benefits of a Bayesian network is having the ability to generate scenarios to create several “what-if” situations. The synthetic data generator does just that; it allows the user randomly generate values for each variable in a realistic manner and consider the scenario.

### **6.1.8. Causal Reasoning Engine (CRE)**

The Causal Reasoning Engine (CRE) is a multipurpose engine for diagnosing probable causes from observed effects and predictors (non-effects that may be positively correlated with a cause). It can be used to provide the posterior probability of each cause, or to generate *scenarios* (combinations of causes) that explain the observations. It dynamically learns what attributes of the data other than symptoms may be predictive of a cause.

The CRE uses valid Bayesian inference methods, so it does not have the pathological behavior of many ad hoc systems. It is based on an intuitive framework for causal modeling, which we call *explanation-based*. A symptom has two possible distributions: one if it is explained (that is, if any of its causes is present), and another if it is unexplained (none of its causes is present). This allows for a very simple representation of the model and very efficient algorithms for inference.

The CRE is very easy to apply to a new domain: the modeler only needs to specify the causes and symptoms, the causes of each symptom, and a small number of numeric parameters. For each cause, the modeler must specify the parameters of its prior distribution. For each symptom, the modeler must specify the probability that it occurs if it is explained and the probability that it occurs if it is unexplained. Symptoms with multiple values (those that are not just present/absent) are supported as a natural extension. Once the model has been specified, the system is ready to go.

The CRE supports both offline and online processing of data. For example, it can be used to monitor cases as they arrive in real time. Or, it can be used as a data-mining tool to discover predictive relationships in data. The CRE also supports both user-driven and data-driven processes.

The CRE is a fully decoupled component: it can communicate with other processes, but is not tied to any other process or user interface. For example, it is not tied to a GUI front-end, although it can support one. This makes it easy to integrate with other components as part of a larger system.

#### **6.1.9. Threat Assessment Module (TAM) Demonstrator**

QLI used the CRE to develop and demonstrate a method enabling the syndromic surveillance of disease outbreaks, called the Threat Assessment Module (TAM). The TAM demonstrated the use of the CRE to assess level of threat based on monitoring individual cases of possible threats (for example, patient records from a hospital). A threat assessment is a representation of a state of belief regarding the probability distribution over a number of possible threats.

The TAM receives incoming hospital patient data and monitors it for various threats (e.g. anthrax). It uses a probabilistic model of diseases and their symptoms to diagnose individual cases and compute the probabilities of each disease in the overall population. These individual diagnoses (case assessments) and population summaries (summary assessments) are inputs to a warning generator, which alerts the users of the system when a threat's probability goes beyond a threshold. The TAM also does online data-mining of the patient records, looking for demographic features that are predictive of threats.

The primary TAM window displays the monitored threats and their current state (based on the most recent summary assessment). The user can access additional information, including background information about the threat, a graph of its assessments over time, predictors of a given threat, and those cases with a high probability of being instances of a given threat. The user can also examine individual cases (patient records) and their assessments.

The TAM is divided into two subsystems: the front end, or GUI (Graphical User Interface), which manages user interactions (cf. Figure 19); and the back end, provided by the CRE, which performs the computations and maintains the system's model of the population. These two subsystems are only coupled through message-passing; they run in separate process-spaces, may be run on separate machines, and multiple front ends may be connected to a single back end.

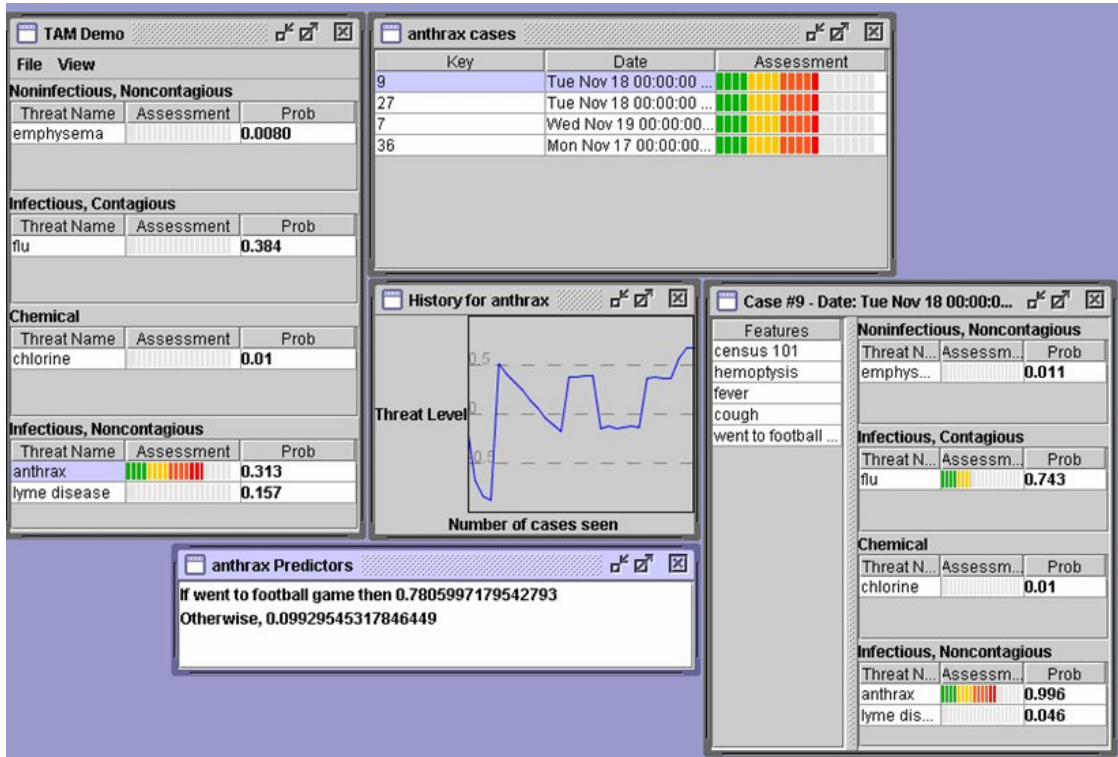


Figure 19: Syndromic Surveillance with the Threat Assessment Module

#### 6.1.10. Dynamic Distributed Data Mining (D3M)

Currently, data mining algorithms work only on a centralized set of data. As the number of data sources increase, especially those providing dynamic data, it becomes increasingly hard to collect data into a single location for analysis. Within IBWTP, QLI outlined a concept for a framework enabling dynamic discovery across a streaming, distributed data environment. The initial implementation concept was to dynamically generate one or more Bayesian networks from data gathered from heterogeneous data sources over a training horizon, and to use the Bayesian network models for predictive modeling over a forecasting horizon. After the forecasting phase, new training data is gathered to augment or possibly replace the previously generated networks. This provides a natural framework for adaptive learning in a dynamic environment. In addition, the Bayesian networks can be used to discover optimal strategies and transmit them to (possibly) remote discovery agents. The discovery agents can be classified as either strategy agents or model agents. A strategy agent acts as a filter on incoming data to dynamically identify observations that satisfy sets of objectives and constraints. Instances of such observations can then be transmitted to interested/relevant parties. A model agent will be used to predict the value of the target feature of interest from the incoming data

stream if the data stream contains a complete set of input attributes. In applications where the data attributes are distributed across several data sources, it may be necessary to first consolidate a subset of potentially interesting observations from different sources into a central site and filter the consolidated set through the discovery agents. In other applications, local models can be generated directly at the remote data sources. For some applications, such as ship maintenance, the observations or predictions of interest can be used to identify optimal response actions using the QLI Adaptive Optimization Engine (AOE) and Real-time Adaptive Planning Framework. Discovering informative network structures and strategies and using them to dynamically process data streams provides a foundation for the QLI vision of linking awareness to action via intelligent computing.

For example, in a shipboard environment, it would be valuable to identify combinations of process conditions of onboard machinery (as measured by physical sensors) that might result in unsatisfactory equipment properties. Identifying these instances to maintenance crew in a timely fashion and identifying the appropriate and optimal corrective actions can reduce the possibility of failure and improve operating characteristics. In an intelligence application, information on individuals or other entities that are flagged as suspicious can be immediately sent to the appropriate authorities. A similar process can be put in place to identify potential disease outbreaks across a global setting.

This technology will be further developed in future projects.

#### **6.1.11. Other Awareness Capabilities**

Other IBWTP ‘Awareness’ Integration technologies noted in Figures 9 and 10 that were pursued, conceptualized, and developed, but not elaborated in further detail in this final report, include:

- **Targeted Information Dissemination (TID):** Technology for “intelligent” push of data and information, getting the right information to the right people at the right time, based on understanding of user’s needs, interests, and situational context. The underlying concepts were originally developed within IBWTP but later moved to a separate project under contract to the Air Force Research Lab.
- **Dynamic Query Processing (DP):** A framework for generating and executing complex query plans over IDM.
- **Structure Learning Adjacency Matrix for Genetic Algorithms (SLAM-GA):** A methodology to learn the structure of a Bayesian Network from a set of given data.
- **Knowledge Extraction Engine (KEE):** A conceptual framework for automatically integrating and deploying a number of different data mining algorithms.
- **Random Data Generator:** A mechanism for generating random data that has distribution according to a given probability curve.
- **Constellation:** A conceptual data mining framework for automatically decomposing large sets of data into more manageable subsets on which to perform data mining and coalesce the results into a distributed virtual data model.

- **Pathfinder:** Java-based interfaces to a tool for text analysis. This was undertaken by QLI's subcontractor, Presearch Incorporated.
- **Sensor Survey:** A comprehensive survey of sensors for biological threats.

## 6.2. Action Capabilities

Work in Task 3, 'Development of Action Capabilities' concentrated on developing, demonstrating, and enhancing technologies supporting rapid response to identified situations of a potential or given biochemical attack.

The effort was primarily composed of two parts:

- Optimization
  - Adaptive Optimization® Engine (AOE)
  - AOE Benchmark
  - Resource Placement Framework (RPF)
  - Sensor Location Demonstrator
- Plan Generation and Execution
  - Real-Time Adaptive Planning (RAP)
  - Truck Routing Demonstrator
  - Distributed Plan Execution
  - AgentCalc Demonstrator

The rest of this section outlines the primary work accomplished in the scope of this task.

### 6.2.1. Adaptive Optimization® Engine (AOE)

The core version of the Adaptive Optimization Engine (AOE) was previously developed and deployed by QLI to solve a number of complex dynamic problems involving scheduling and allocation in both commercial and military domains. The AOE can be used to solve a wide variety of optimization and constraint satisfaction problems. It can solve problems with linear or non-linear constraints and objectives, integer, continuous, incremental, or set-valued variables, and any mixture thereof. Users can solve their problems without regard to the particular type of problem represented, or knowledge of specific problem solving techniques. AOE models typically have tens or hundreds, rather than thousands of variables and constraints. Within IBWTP, the Java version of the AOE was used to develop models in the IBWTP application domain of responding to biochemical threats. It was also maintained according to latest software practices and brought to QLI's SW Maturity Level 4.

### 6.2.2. AOE Utilities and Subpackages

QLI developed the following utilities and subpackages as part of improving the AOE for use in the project.

The AOE Benchmark subpackage is a collection of models on which the AOE can be tested for robustness and performance. QLI created the benchmark suite in order to rapidly test changes to the AOE.

The AOE Test Utils subpackage contains mock AOE objects to aid anyone who needs to make junit tests of their models for ensuring SW quality.

The Combinatorics subpackage is a math utility library for permutations and combinations that the AOE needs. It is based on a variety of known algorithms, some or all of which are referenced in the source code documentation. It was originally used by the AOE, but was also extracted out into a separate library package to allow for use by other programs.

The Interpolation subpackage contains classes to build and use linear and cubic splines in several forms. It was originally used by the AOE, but was also extracted out into a separate library package to allow for use by other programs, such as the random data generator.

The Random Number Generator subpackage extends current random number generation algorithms by allowing explicit control over exactly which algorithm is being used to generate the (pseudo-)random numbers. This package generates very long unique sequences and was originally used by the AOE, but has been extracted into a separate library package to allow for use by other programs.

### **6.2.3. Resource Placement Framework (RPF)**

QLI developed the Resource Placement Framework (RPF) as a model using the AOE to optimize placement of resources in 2-dimensional or 3-dimensional geographical environments. Resources may include sensors, personnel, equipment, and facilities. The purpose of this package was to make the task of modeling the general resource placement problem easier.

Creation of a RPF model involves defining the area that needs coverage by resources, the amount and extent of coverage created by locating a resource at a particular location, and the interaction of nearby resources (i.e. does locating two resources at the same place double the coverage or does it accumulate in other ways). The AOE then produces coordinates for the different resources in a way that optimizes their placement over the given area.

### **6.2.4. Sensor Location Model Demonstrator**

QLI developed the Sensor Location Model as a demonstrator application of the RPF to the specific domain of optimally placing biological sensors (either mechanical devices, or humans collecting samples over a region) to cover a given geographical area (based on census tracts) to maximize the amount of population covered by the sensor's detection capabilities. It is integrated with the Plume Tool (providing input of area to cover) and the Weather Tool (providing input determining sensor range) developed as part of the Awareness capabilities, as well as the Map Tool (providing both input of area to cover as well as display output of determined sensor positions and their ranges) developed as part of the Control capabilities.

Figure 20 shows the input to the Sensor Location Model, where the user selects number and type of sensors, as well as any other criteria to account for, such as weather conditions or plume dispersion models. The user also has control over how long the sensor model should run.

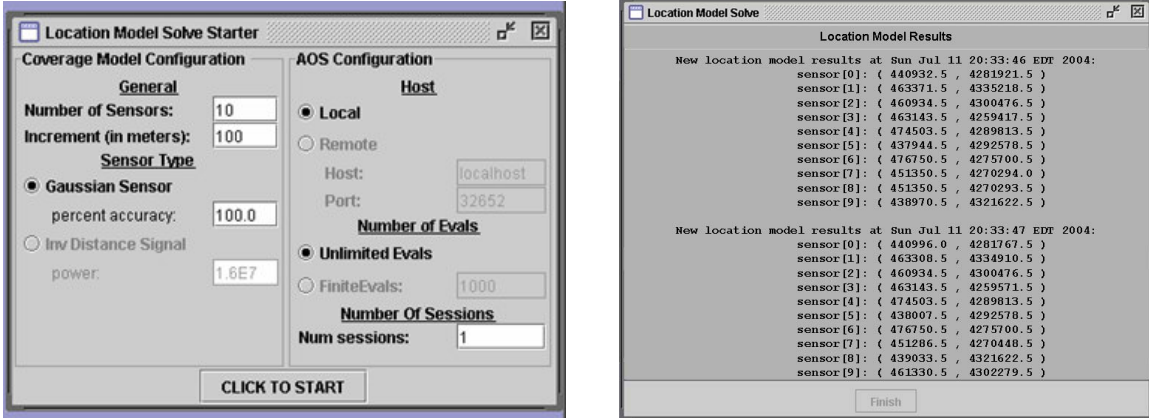


Figure 20: User Input to Sensor Location Model

Figure 21 depicts the output of the Sensor Placement tool. The left hand side shows two possible solutions of locating 10 sensors, the right hand side shows one of these solutions as it appears on the iKE map.

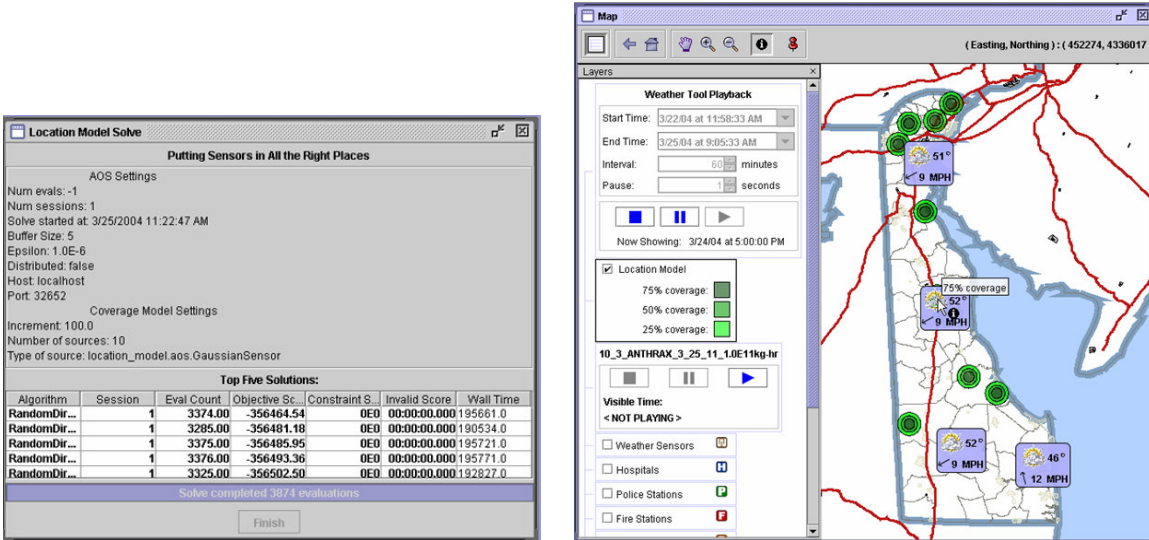


Figure 21: Output of Sensor Location Model

### 6.2.5. Real-Time Adaptive Planning Framework (RAP)

A fundamental problem with planning is that even the best plans can be quickly invalidated by a dynamic world state. Current approaches to this problem rely either on dynamically re-planning when the plan is invalidated (which is often too slow) or on planning for all possible contingencies in advance and integrating them into one unified plan. This approach, however, proves inflexible and incapable of handling dynamic, real-world scenarios. Within IBWTP, QLI developed the “Real-Time Environment” (RTE) that was later renamed to the “Real-Time Adaptive Planning” (RAP) Framework that

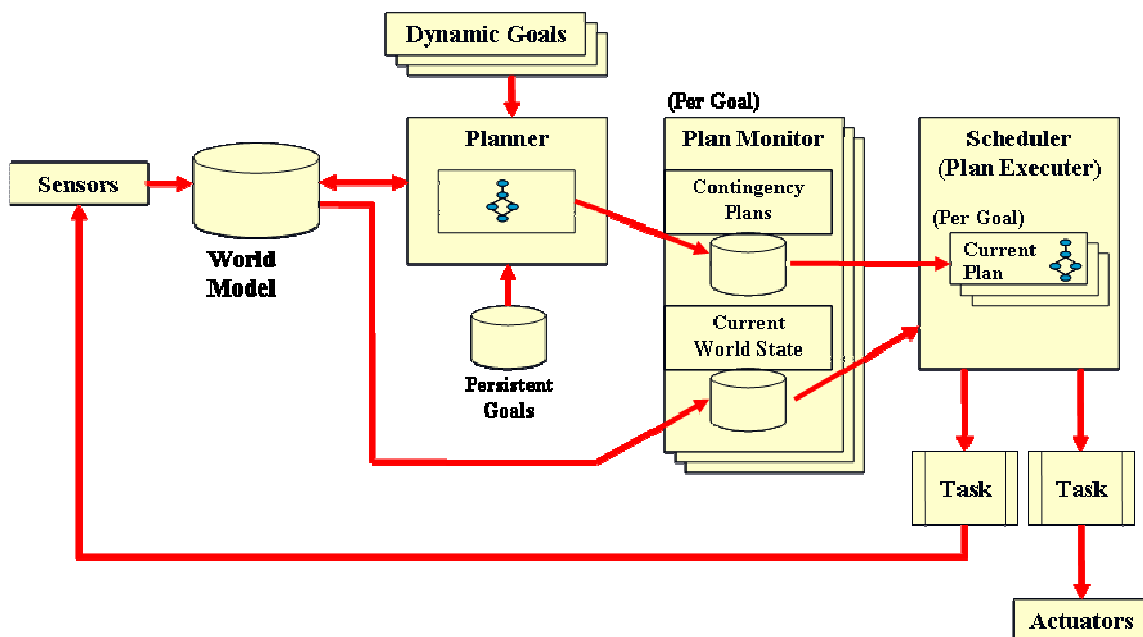


plans continually, generating contingency plans for specific probable world states, works with any plan generation engine, and uses computational resources effectively.

The RAP planner continually generates plans to achieve the current goal based not only on the current world state but also on selected probable future world states. While the RAP works simultaneously with other planning and speculative engines to come up with these future world states, this framework also selects future states to plan for using user-defined criteria. Contingency plans are retrieved from a cache based upon corresponding changes in world state. QLI implemented the first version of the RAP as a single process system.

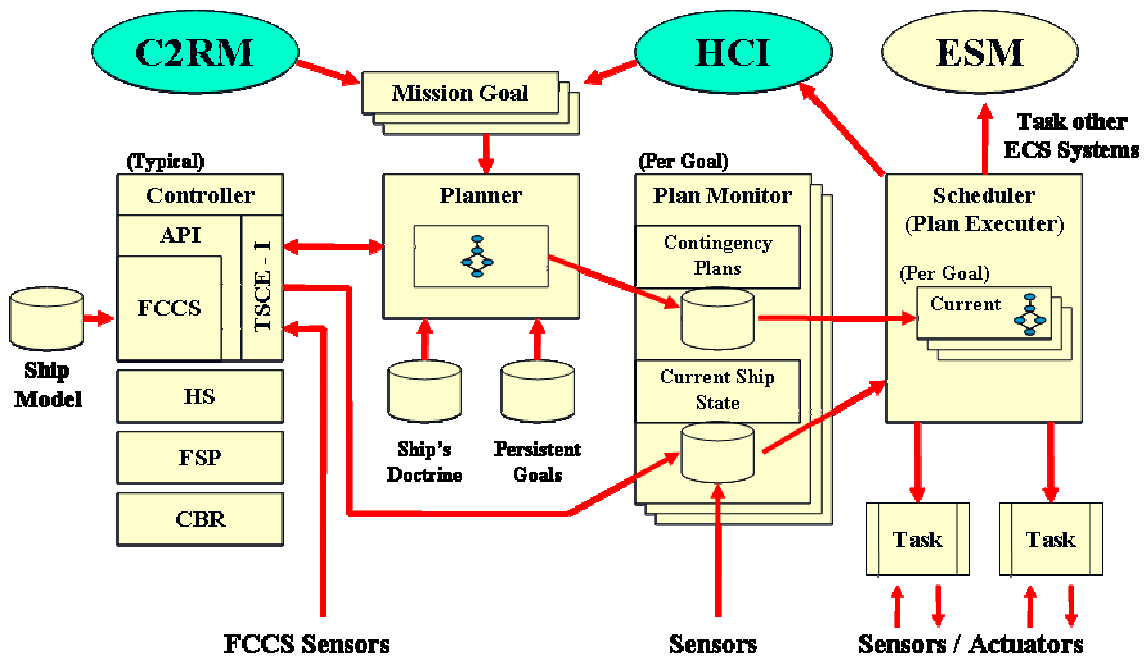
QLI implemented a second version of the RAP using a totally distributed agent-based architecture as exemplified in Figure 22. This allows for deployment of the various components on different machines and seamless on-the-fly integration of different mechanisms for situational awareness, state representation, plan generation, event triggering, and plan execution.

The domain of the system is a model made up of a weighted graph and a number of actuators which have positions at vertices in the graph. The system is given one or more goals which are a single actuator paired with a vertex in the graph. The system determines possible paths between the current vertex of the actuator and the other vertex contained in the goal. These paths are stored in an agent that orders the plans by cost. Another agent will execute the best plan for a particular goal by changing the vertex associated with an actuator. The idea is to simulate the actuator following the steps of the plan to achieve a goal. The weights of the graph can be changed at any time which will affect the ordering of the plans. The system will take account of the new ordering and the agents executing plans will always follow the current best plan.



**Figure 22: General Distributed Realtime Adaptive Planning Architecture**

Figure 23 shows the integration of this distributed architecture with damage control systems, sensors, and other systems aboard a Navy ship.



**Figure 23: Distributed RAP Architecture applied to a Damage Control System**

### 6.2.6. Truck Routing Demonstrator

QLI demonstrated the contingency planning capabilities of the RAP using a truck routing problem. The truck travels over a known network of roads. The demonstrator uses an A-star search heuristic as its planner. The plans are the routes the truck may take from its current position to its destination (cf. Figure 24). Multiple plans are generated taking into account the probability of a particular road becoming inaccessible, for example due to a traffic jam, road construction, flooding, etc. The user can manually set via a graphical user interface which roads are open or closed as the virtual truck is traveling. The module will compare these road closures and openings to the routes that have already been generated by the embedded engine. If a particular road that is on the truck's route becomes inaccessible, the module searches for a route in its plan cache that avoids the blocked road (cf. Figure 25). If one is not present, the module will calculate a new route using the present network graph of accessible roads and begin the contingency planning process again with this new network graph.

The demonstrator was developed on both the centralized single process RAP as well as the decentralized multi-machine version of the RAP.

The demonstrator showed potential users the applicability of RAP to other domains, such as fluid routing (e.g. chilled water supply aboard DD(X)), ballast balancing aboard LPD class ships, damage control (fire suppression and response team deployment) aboard DD(X) and LPD, as well as troop movement and resource deployment in unknown territories.

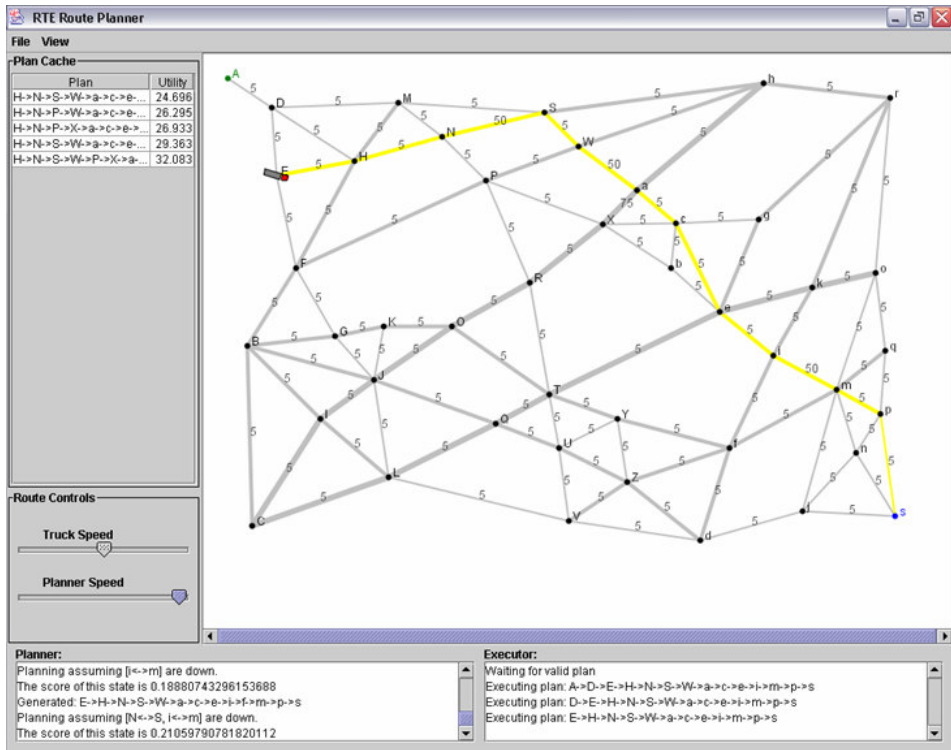


Figure 24: Planned Truck Route with Contingencies

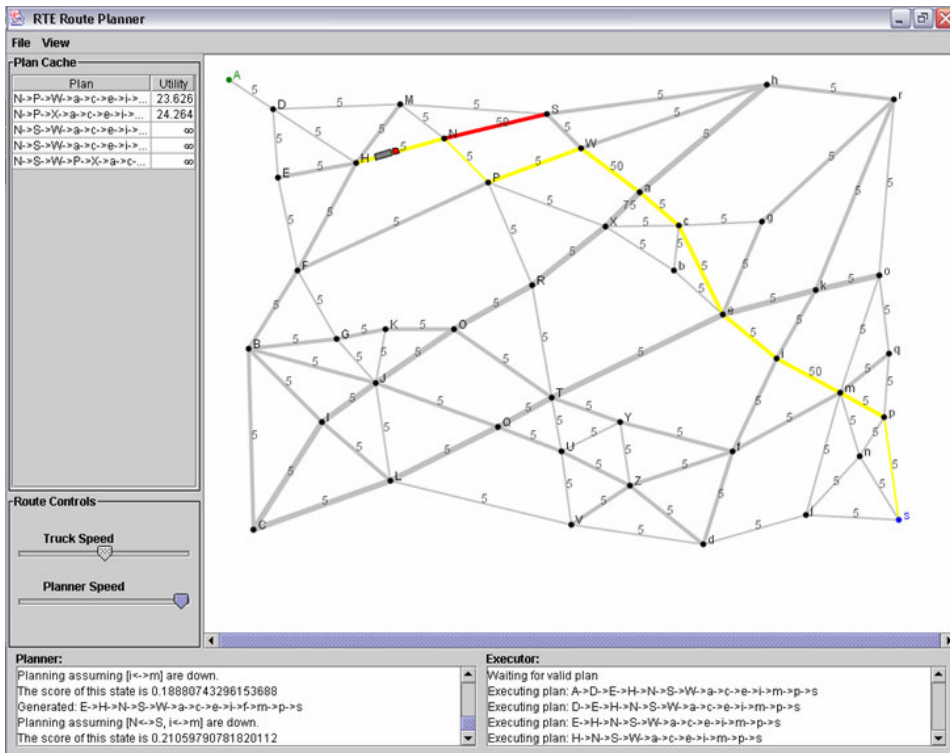


Figure 25: New Truck Route

### 6.2.7. Distributed Process Execution (DPE)

Most process and workplan execution systems rely on the use of a centralized server to coordinate or orchestrate the execution of the tasks by the corresponding actors. However, often there may be cases where a centralized server poses a bottleneck, a vulnerability, or is simply not available. QLI has developed and implemented a system for totally decentralized execution of business processes by autonomous systems on its Multi-Agent Development Environment. This package allows developers to develop (Java Process Definition Language (jPdl) files that describe business processes that will be executed by a community of agents. Coordination and synchronization of process tasks is automatically distributed and managed by the participant agents in collaboration with each other.

The Process Library Agent (cf. Figure 26) is the central repository for processes that agents may execute. When an agent is requested to store a jPdl process, it first runs it through a process analyzer. The process analyzer fragments the process based on predefined roles in the process.

The Process Execution Behaviour (this spelling is intentionally used, as “behaviour” is a named software construct in JADE, on which DPE is implemented) of an agent (cf. Figure 27) handles the setup and management of the Java Business Process Manager (jBpm) within the agent. jBpm stores processes in a process library by their name. The agent can request the Process Execution Behaviour to initiate a process by name. Once initiated, jBpm tracks the execution of the process. If a state within the process is implemented by a behaviour, then the Process Execution Behaviour will notify jBpm when that behaviour is done running.

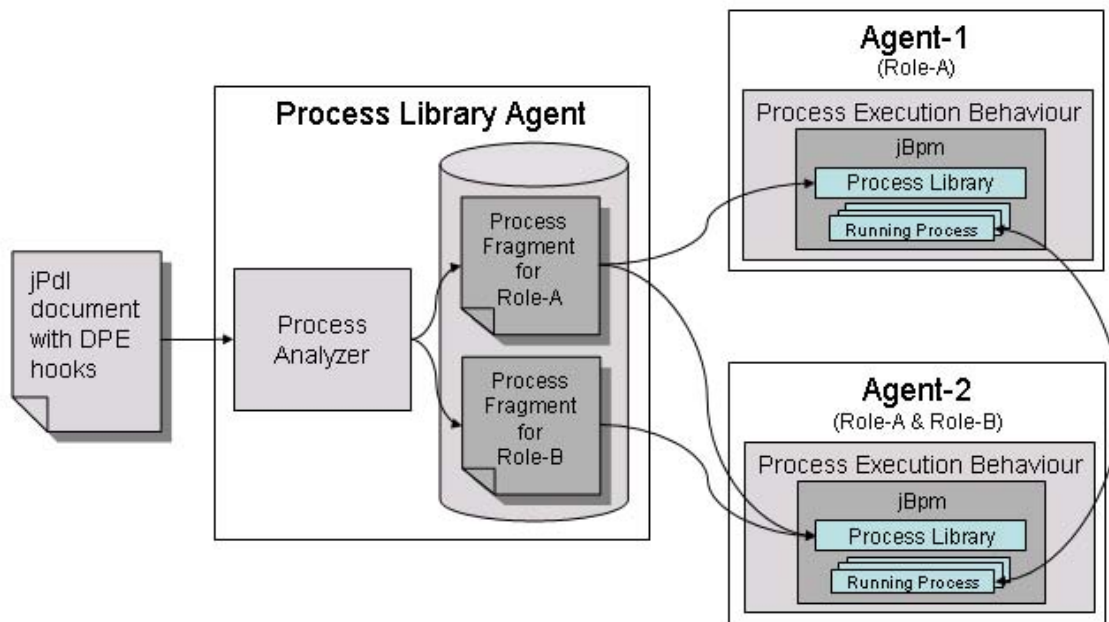
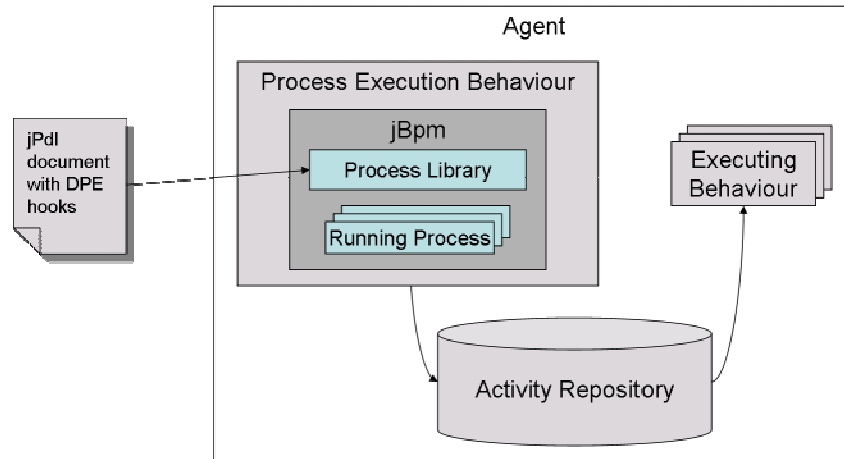


Figure 26: DPE System Architecture Diagram

The Activity Repository holds information about behaviours that can be created and executed within the agent. These behaviours have associated metadata that allows the Process Execution Behaviour to search for a behaviour to perform the current activity in the process.

The jPdl document that contains the process description needs to contain hooks designed for the Distributed Process Execution. The hooks needed to get most processes running are provided within the DPE package; however it is still possible to provide customized extensions to steps within the process.

This technology will be further developed in future projects.



**Figure 27: DPE Agent Architecture**

### 6.2.8. Agentcalc Demonstrator

QLI developed Agentcalc as a demonstrator of the DPE technology. The program takes a lisp-like mathematical expression and converts it into a jPDL file for distributed process execution to calculate using agents providing different arithmetical calculation services. It was not designed to be an efficient way to perform computations, but rather to be a test bed for the implementation and demonstration of distributed process execution. The technology that supports coordination of many small grain-sized tasks such as numeric computation can provably “scale-up” to large grain-sized tasks such as scenario evaluation.

There are two types of agents: The Calculator agents solve numerical problems that they are given according to the distributed process. The Agentcalc client agent takes an arbitrary arithmetic expression from the user and decomposes it into a jPdl process that can then be distributed among the available calculator agents.

## 6.3. Control Capabilities

Work in Task 4, ‘Development of Control Capabilities’ concentrated on developing, demonstrating, and enhancing technologies supporting an integrated command and control environment enabling same-time, different-place support of information visualization, application linking, and decision support to decision makers assessing and responding to potential or given biochemical attacks.

The effort was primarily composed of three parts:

- Integrated Collaboration Framework
  - Integrated Knowledge Environment (iKE)
  - Collaboration Manager (Collman)
  - Distributed Application Framework (DAF)
- Collaboration Tools
  - Clearboard
  - A/V Conferencing
  - Multi-mouse
  - User Modeling
  - Menu Manager
- Interactive Knowledge Wall (IKW)
  - Fixed IKW
  - Portable Control Units

The rest of this section outlines the primary work accomplished in the scope of this task.

### **6.3.1. Integrated Knowledge Environment (iKE)**

QLI developed the Integrated Knowledge Environment (iKE) technology to support group collaboration and access to a large, continually changing, complex set of information. The three key aspects of iKE are task-specific graphical interfaces to information, multiple user collaborative use of those interfaces, and integration of real-time process control with these interfaces.

Information visualization and custom interface generation is the creation of custom, user-defined graphical visualizations of data that the user (a decision maker or technical specialist) can directly manipulate. By creating an on-screen representation of information, such as a chart demonstrating the distribution of patients over hospitals in Delaware, a user can easily observe interesting features that may not be apparent in a simple table, or even a default visualization. The way in which the information is displayed is customized: to fit the data, to fit the task, and to fit the preferences of the user.

CSCW, or computer supported collaborative work, is the study of computer technology to facilitate multiple users working together. iKE allows multiple users at different terminals to view the 'same' visualizations at the same time. Distributed sessions update the graphic display to show changes in the underlying data – changes that may be performed by teammates, caused by a real-time system process, or required by an alteration in an underlying data resource.

Access to data is one aspect of the iKE system. In addition to powerful visualization, exploration, and collaboration tools, the iKE environment supports the integration of process control GUIs into the work environment. A custom interface can be created to control the operation of a real-time process (such as an atmospheric particle release plume modeling tool) and tie that process to on-screen visualizations of resources

required or generated by that process (such as atmospheric condition data for the duration of the release, and calculated particle concentrations over time). These visualizations can be used to support other tasks, allowing a visual association between processes and displays (the weather data used in THAT plume model is being displayed in THIS map, along with traffic density and population demographics).

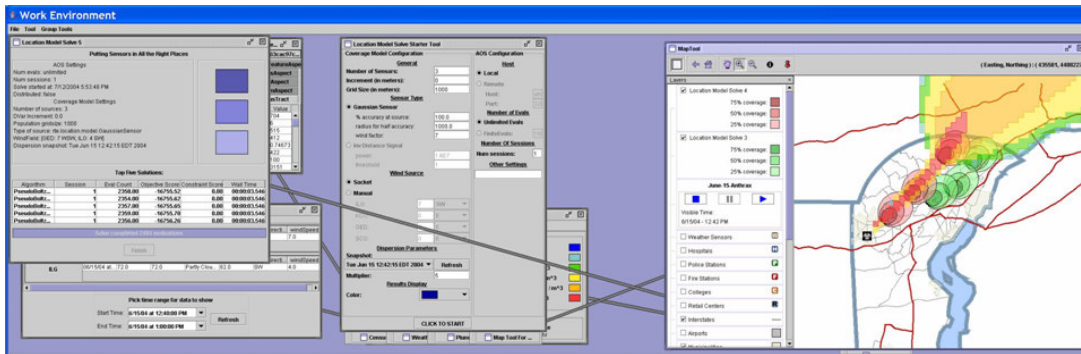
iKE capabilities include:

- Visualization of large amounts of live data
- Custom interfaces for specialized tasks
- Extension & reuse of portions of existing UI elements
- Direct manipulation of data visualizations
- Sharing & coordination of distributed, multiple user access
- Coordination of visualizations of static data, ‘live’ data, system applications, and user interfaces

Decision makers can use the iKE to manipulate various components of the IBWTP system developed in the Awareness, Action, and Control tasks as the components execute, to visualize the progress and results generated by the components, as well as to provide “on-the-fly” composition of combinations of those components. The iKE supports mixed initiative interaction, and support collaborative exploration of evidence, reasoning, scenarios, plans, and consequences.

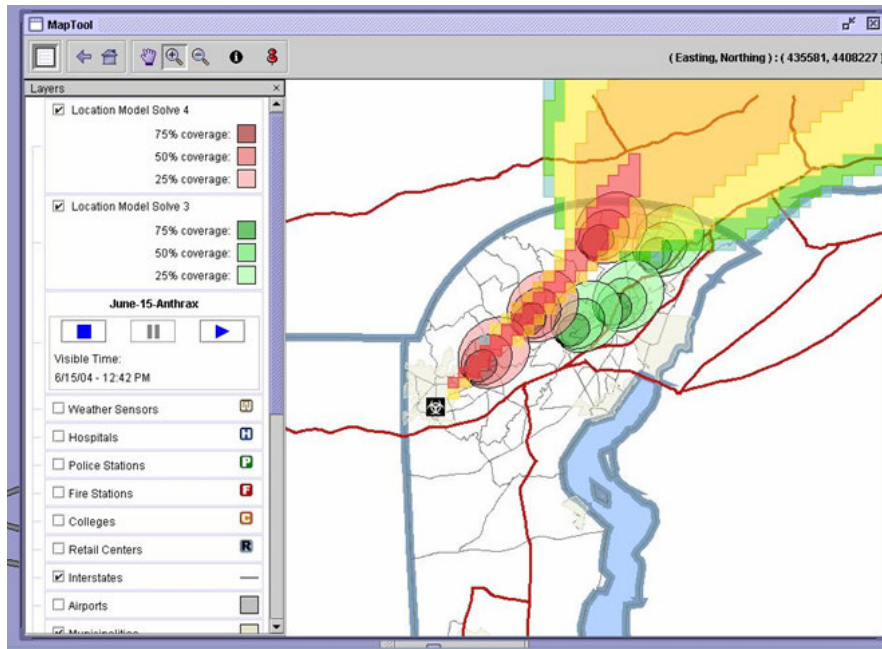
Composeability of applications is enabled simply by the user dragging ‘Tool A’ onto the data input target of ‘Tool B,’ thereby indicating that the output of ‘Tool A’ should be provided as input to ‘Tool B. To enable the various iKE functionalities, QLI used a variety of technologies in data representation and manipulation using entity proxies, persistent data management, and aspect oriented programming.

Figure 28 shows the composition of the Plume Tool, Weather Tool, and Sensor Location Model to optimize and display the placement of sensors over geographic regions taking weather and plume dispersion into account. Figure 29 shows a close up of the resultant map.



**Figure 28: Composable Applications within iKE**





**Figure 29: Output of Dispersion and Sensor Location Models**

### 6.3.2. Collman

Based upon the experience with developing IKE, QLI recognized the need for flexibly managing the team involved in collaboration. QLI developed a concept for a suite of components designed to facilitate teamwork and collaboration called Collman (from **collaboration management**). It is not enough to make a component service available to the team; the **context** in which it is used is also significant. For example, having the capability to simulate an atmospheric dispersion is very different than including a simulation instance in an ongoing analysis. Composition of component service instances facilitates collaboration by reusing the expertise of team members familiar with certain components. Using Collman, clients can quickly view, navigate, manipulate, and extend the component configuration of the team. The Collman interface includes tools for visualizing the current deployment of components by the team. Some components for facilitating collaboration in Collman are:

- **Team monitor:** As clients log into the system, a registry of who is using the system is maintained. Additional information such as what machine they are using, what components are installed on that machine, and what component instances the user is accessing, is also maintained.
- **Team model:** As team members use component tools to achieve tasks and collaborate, Collman records who has used which components how often. Over time, Collman maintains a model of how familiar team members are most with particular component services, and which team members are most often relied upon to operate them.
- **System monitor:** A registry of what component services are available in the current Collman environment. As machines and network resources go up and



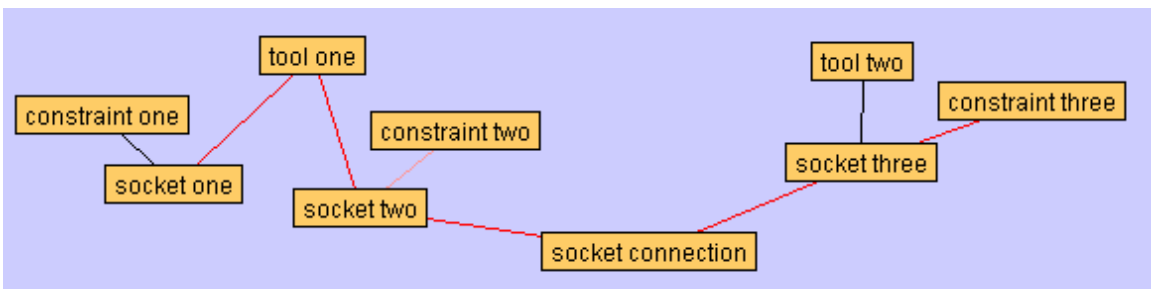
down, the system monitor updates. Using the monitor, the system can detect when a required service is not available.

- **Task analysis:** Collman can ‘remember’ commonly used component configurations and user preferences when performing tasks. Comparing the configurations used previously and the current deployment configuration, Collman can analyze the existing component configuration to help discover opportunities for collaboration, or to prevent duplication of work.
- **DPOL monitor:** records which components are configured to share a component instance of the Distributed Persistent Object Layer (DPOL). Using the DPOL monitor, teams can identify components related by the data model they share, and help understand the impact performing operations on that model may have on the system.

QLI implemented this architecture in a prototype including features of all the above areas, including distributed shared data, distributed process management, and distributed visualizations.

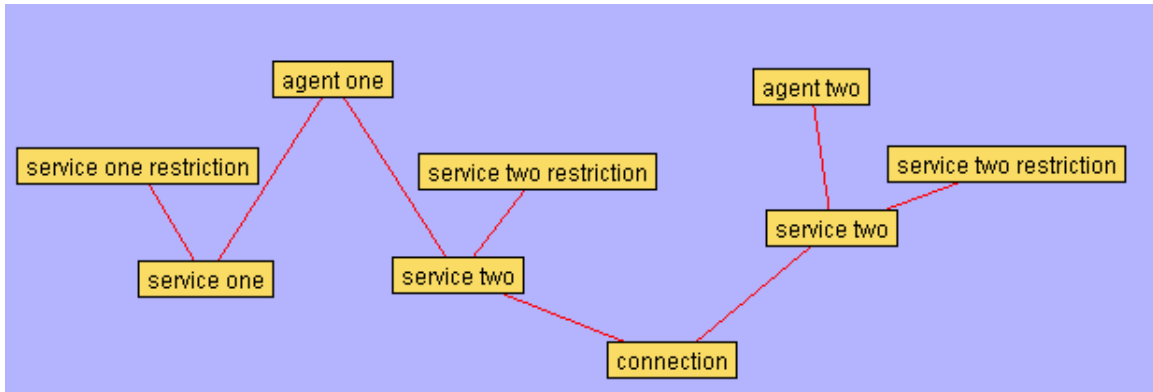
The Collman Workbench is a data structure for component-based groupware that provides interfaces for components and for communication between components. The Workbench is a graph structure (cf. Figure 30), made up of a few special node types:

- **Tools:** These correspond to “components” as generally used in component-based software.
- **Sockets:** Tools have Sockets, which are communication starting and end points. Tools do not directly communicate with each other, but go through their Sockets to communicate with other Tools.
- **Socket Connections:** These are links between Sockets; data going between Sockets goes through Socket Connections. Before Tools may communicate, a Socket Connection must be established between their Sockets.
- **Socket Constraints:** Sockets have Socket Constraints that dictate whether or not a Socket Connection may be established between a Socket and other sockets.



**Figure 30: Socket Connections in the Collman Workbench**

Figure 31 shows an example agent network in Collman with a connection between agents one and two.



**Figure 31: Agent Network Example in the Collman Workbench**

This agent network can be represented as the Workbench from Figure 29. The following benefits arise from using a Workbench representation of an agent network:

- Agent networks can be attached to a Workbench Graph Desktop to create a graphical representation of the network. The Workbench Graph Desktop allows users to explore relationship between agents in the Workbench and create new connections between agents.
- The workbench/tool metaphor is very effective in describing the relationships between application components, and hides unnecessary complexity involved in typical descriptions of an agent network. For end users, it is easier to learn to use tools and sockets than it is to learn to use agents.
- Because the Workbench is a graph structure, it allows for very useful graph-traversal searching based on any number of predicates. Applications based on this easily-searchable structure can be used effectively because end users are aware of what components are available to them and the properties of those components.

Each Collman client application is centered on a CollmanClientAgent instance. Each CollmanClientAgent maintains a local Workbench, which keeps track of its local agent resources. CollmanClientAgents can locate other CollmanClientAgents on a network, and can share the contents of their Workbenches with each other. Also, CollmanClientAgents can register themselves as listeners for changes in others' Workbenches. A CollmanClientAgent that receives information about the contents of another CollmanClientAgent's Workbench can fold that information into its own local Workbench. A Workbench containing local and remote agent information can be searched and used as a basis for visualizations just as effectively as an entirely local Workbench.

When a CollmanClientAgent creates a new agent to perform some task, it will create a new Tool node instance that is representative of the agent and place this node in its local Workbench. It will also create Socket node instances for each of the services that the agent can perform, and add them to its Workbench. For each restriction on the registration of other agents to this agent's services, the CollmanClientAgent will create a new Socket Constraint node instance and add it to its Workbench. When this new agent registers other agents for its services, the CollmanClientAgent will create Socket Connection node instances representing the connections between agents and add them to

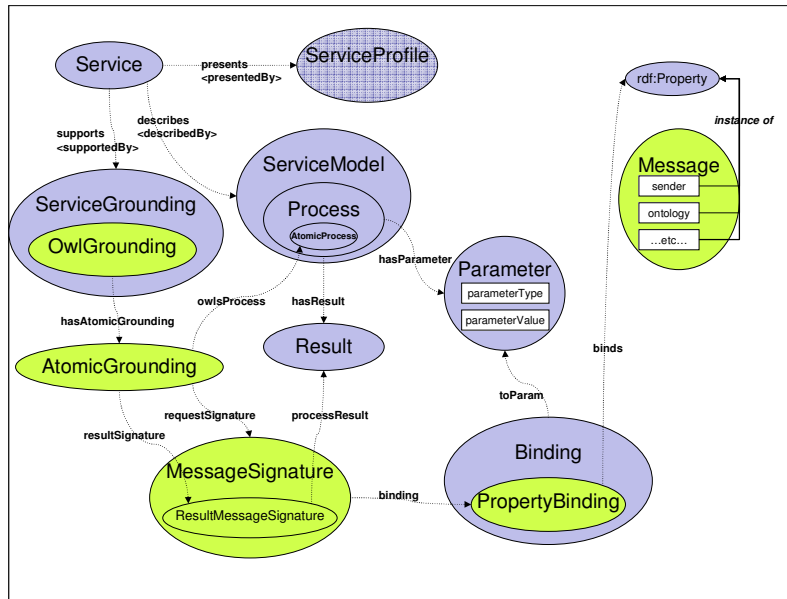
its Workbench. Similarly, if the new agent removes connections to other agents, services, or restrictions on services, the CollmanClientAgent will remove the corresponding nodes from its Workbench.

A Collman client application can give the user a visual representation of the local Workbench of the application's CollmanClientAgent. It can also list for them other CollmanClientAgents that exist on the network, and can give them the choice of importing the Workbenches of remote CollmanClientAgents into their local Workbench. Visualizations based on their local Workbench can be expanded to show information about the remote agent resources that they have imported. A Collman client application can have a Workbench Graph Desktop that is based on the application's CollmanClientAgent's Workbench. The Workbench Graph Desktop will update to reflect changes in the underlying Workbench. Also, user input to the Workbench Graph Desktop may instigate a change to the underlying Workbench, and, subject to approval from the CollmanClientAgent, to the underlying agent network. Changes that the user may instigate include the creation and destruction of agent instances, modifications of agent state, and the creation and destruction of connections between agents.

### **6.3.3. Distributed Application Framework (DAF)**

QLI developed the concept for a Distributed Application Framework (DAF) that addresses the problem of dynamic web service discovery and invocation via semantic web service technology. In standard agent systems, the services that Agent A will want to invoke on Agent B need to be known at the time Agent A is implemented. If later additional services are added to Agent B, Agent A would not be able to invoke these services unless the implementation of Agent A was then changed. Attempts to handle this problem in the web services domain include WSDL and UDDI.

The approach QLI took to solve this problem was to describe the services offered by an agent with Ontologies. DAF agents use ontologies written in OWL and OWL-S to describe the services they offer. OWL is a markup language based on XML used to describe ontologies that was also designed to allow software to reason about ontologies written in OWL. OWL-S is a specialized ontology using OWL used to describe services (cf. Figure 32). The ontology has information such as the name of the service, input and output parameters, and the error conditions of the service.



**Figure 32: AclOwlGrounding Diagram**

### 6.3.4. Clearboard

QLI developed the ClearBoard component, allowing users across distributed groups to layer graphics and text over the display of existing applications within the IKE framework. Users can use common annotations such as drawing shapes, scribbling, changing colors, entering text, and inserting images. ClearBoard component extends the more commonly available Whiteboard components in several ways.

First, the ClearBoard is capable of being used as a transparent overlay. That is, the ClearBoard can be placed over other components with the underlying component showing through. This provides the ability to annotate visual representations within applications. In this way distributed teams can more easily communicate, share knowledge, and come to conclusions about the presented data. Because the ClearBoard overlays other components, the clearboard needs the ability to replicate some of the common transforms that components can undergo. These transforms include:

- a) Scrolling
- b) Scaling

Thus when the ClearBoard is placed over a component that scrolls, it must have the capability to scroll its annotations to match the underlying component. Likewise, when placed over a component that scales, the ClearBoard must have the ability to scale its annotations to match the underlying component. One example type of component that can easily demonstrate the need for such capabilities is a zoomable map component.

As an overlay component, the ClearBoard must also be able to be placed in a view-only mode, where the user can interact with the underlying component and still see the annotations presented by the ClearBoard. The user may also want to temporarily hide the ClearBoard annotations, so the ability to hide the ClearBoard is necessary as well.

Secondly, because the ClearBoard is distributed, it must present the same set of annotations to all distributed participants. This can be a difficult task given the latency introduced by the network. Strategies must be incorporated to insure that each participating distributed ClearBoard synchronizes to the same state when all ClearBoard messages have been delivered.

The ClearBoard component must also provide capabilities for users to acquire peer awareness. Awareness will be facilitated in the following ways:

- a) Ability to query the system for what a given peer is looking at
- b) Identifiers of an author associated with modifications by that author

The two major technical hurdles QLI handled during development of ClearBoard were the visual presentation and the communications between distributed ClearBoards. For the visual presentation, QLI used the open source Piccolo toolkit. Communication was handled by a hybrid synchronization system; presenting most of the data in an immutable fashion.

### **6.3.5. A/V Conference**

The benefits of video conferencing are well known and accepted in today's collaborative work environment. But the costs and inconvenience of pure video conferencing solutions using hardware remains a barrier to making this a common collaborative solution. Most software audio/video conferencing applications available in the market are mostly platform dependent and can not run across majority of the platforms, and use some sort of registration service, not direct point-to-point connection. In addition, the majority of the audio/video conferencing applications do not provide user-control of the audio/video so that it is hard to adjust based on different network bandwidth situations.

QLI developed a Java-based platform-independent audio/video conferencing component (A/V Conference) that can be integrated into the IKE environment. A/V Conference allows face-to-face meeting using unicast, multiple unicast, multicast or broadcast approaches.

A/V Conference is designed as a Java component. It is wrapped in a Java internal frame so that it can be placed in any Java container. It is easy to be integrated into any Java-based application. It can be simply placed into a Java container like any other JInternalFrame component.

The application uses direct point-to-point connection to eliminate performance issues caused by a central server relaying data streams to provide fast data delivery and data security. The biggest issue and key problem for software video conferencing applications is performance. A/V Conference uses the industry-standard Real Time Protocol (RTP) running on top of UDP to guarantee the fastest data delivery possible to achieve real time communications among users.

### **6.3.6. Multi-Mouse**

QLI designed and implemented a java-based, software solution for supporting the use of different mice and keyboards by several users with one shared collaborative workspace (iKE). The system transmits mouse and keyboard I/O from one Multi-Mouse client to

another. The work environment uses mouse interactions with this shared display to allow users to move a tool between the wall and their client, duplicate a tool on multiple machines, or to link to a tool that exists on another machine.

### **6.3.7. User Modeling**

- Control user access to system capabilities

User id and password authentication facilitates control over user privileges

- Added input to automatic tool network constructor

Modeling user tasks, responsibilities, and preferences adds many interesting possibilities to the automatic tool network construction defined above. For example, the system may elect to show a simplified version (or not show at all!) some of the tools in a tool network, based on the model of the user(s) intended to actually view the result.

- Added information to display in shared work environments

When the system has added information to allow it to distinguish between users, it can add information to the display to indicate what users are viewing, and what their privileges are. For example, suppose a particular tool instance (a map) is being viewed by multiple users on multiple clients. Shading on the map can be used to indicate to all users who is looking at what areas of the map. A key identifying all the users, and a color assigned to each user, is shown to the side. Bold text in the key indicates the users with 'write' or 'modify' privileges for this tool.

### **6.3.8. Menu Manager**

QLI designed and implemented an API for model-based dynamic menu management. Different application components can build a tree model describing what they want their menus to look like; a MenuBarManager instance takes tree models from different components, merges them when there are conflicts, and adds the corresponding menu structures to the menu bar it's managing. This allows application components to only need to worry about the content of their menus; they don't need to include any of the code that does the adding of the menus to a menu bar.

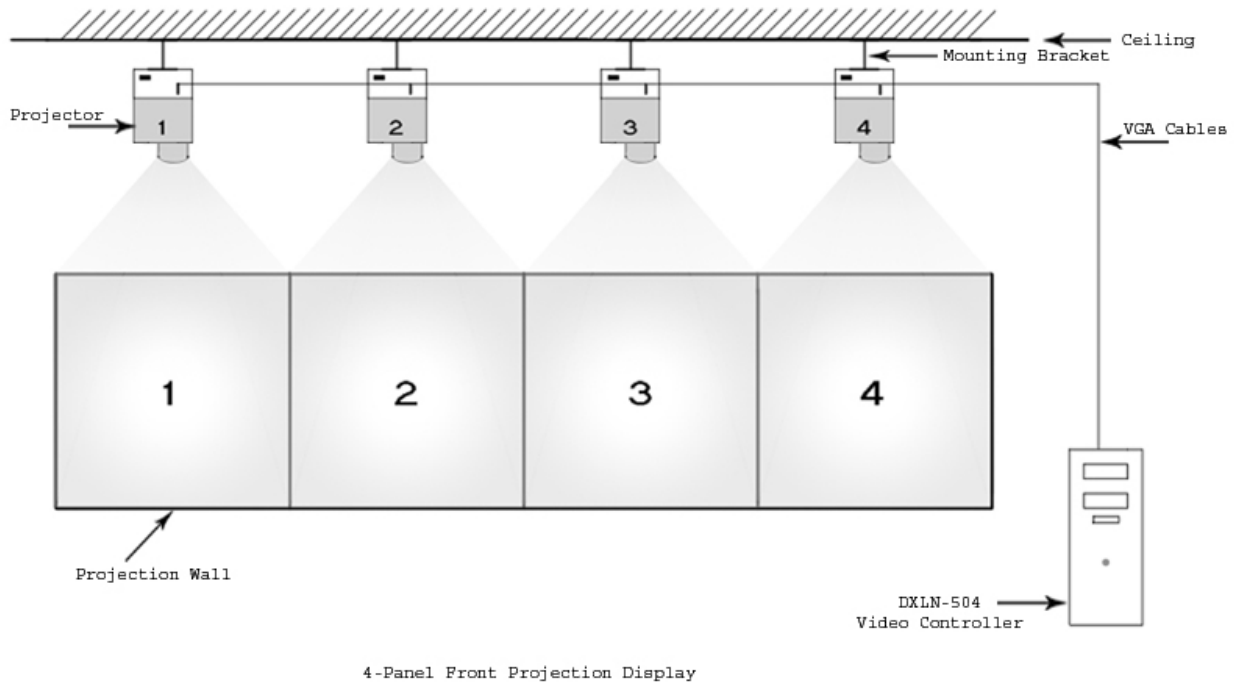
### **6.3.9. Interactive Knowledge Wall (IKW)**

QLI specified the Interactive Knowledge Wall (IKW) to improve on similar display technologies such as the Data Wall developed by Air Force Research Lab; InfoWall developed by InfoValley; Knowledge Board developed by SAIC; and Knowledge Wall developed by SPAWAR. Our goal was to implement a relatively simple, portable, collaborative display technology that supports the distributive nature of the information/knowledge being developed under IBWTP.

Initial operational features include front projection and mouse and keyboard control of the IKW.

The Interactive Knowledge Wall serves as a portal to access information from the system or external sources (such as the internet), and also to aid with collaboration. The high-resolution display is approximately 29 ft by 4ft. The Knowledge Wall module consists of a Dexon DXLN-504 Windows2000&Linux controller and four Epson 811 projectors. This machine controls all access to the Interactive Knowledge Wall display.

The Interactive Knowledge Wall at Quantum Leap Innovations is a complete system built from COTS components. Figure 33 shows the layout of the fixed IKW. The room is 21' (6.4m) long along the display wall by 20' (6.1m) deep. The projectors are attached to the ceiling 9' (2.7m) away from the display wall. The tables and chairs in the room are configured in a “U” shape. The tips of the “U” are 6.5' (2.0m) from the display wall. The projected image on the display wall is 20' (6.1m) long by 3.75' (1.1m) high.

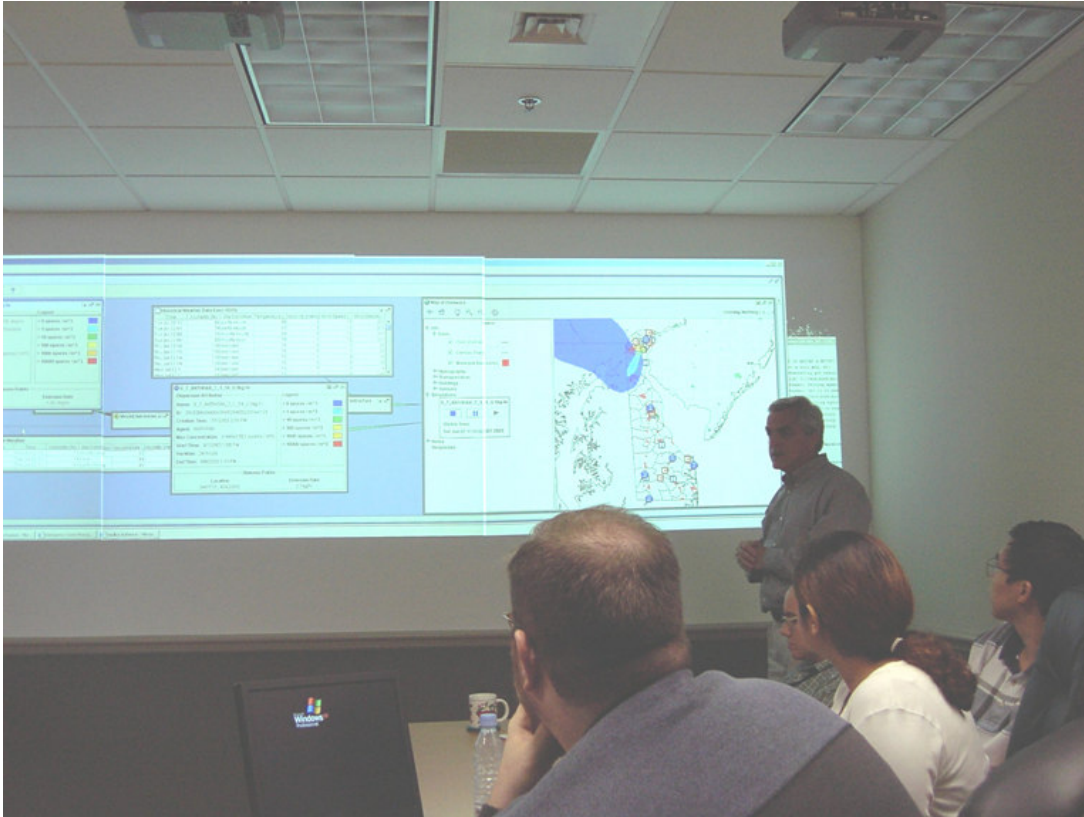


**Figure 33: Overview of the Interactive Knowledge Wall Setup**

*Collaborative/Interactive Benefits of the IKW*

The DXLN-504 system supports a collaborative working environment for enhanced teamwork. The extended display area behaves as a single desktop and allows many applications to be opened and presented simultaneously for sharing of ideas and information (cf. Figure 34). Using a hub connected to the local area network and Virtual Networking Computing (VNC) software, various computer desktops can be linked and presented on the IKW. If space on the display is scarce, windows can be overlapped as well as minimized to make space for other windows to be displayed. Users can interact with their respective desktops using their own keyboard and mouse. While collaborative sessions are taking place, the group leader can use the system remote mouse and keyboard of the DXLN-504 to control and manipulate windows and applications on the IKW. Computers from the local area network can also have their desktops displayed on the wall.

Quarterly integrated demonstrations in increasing complexity of IBWTP technologies were made on the IKW. The demonstrations were named Jenner, Koch, Lister, Moreau, and Nobel, respectively.



**Figure 34: The Fixed Interactive Knowledge Wall.**

### **6.3.10. Portable Control Units (PCU)**

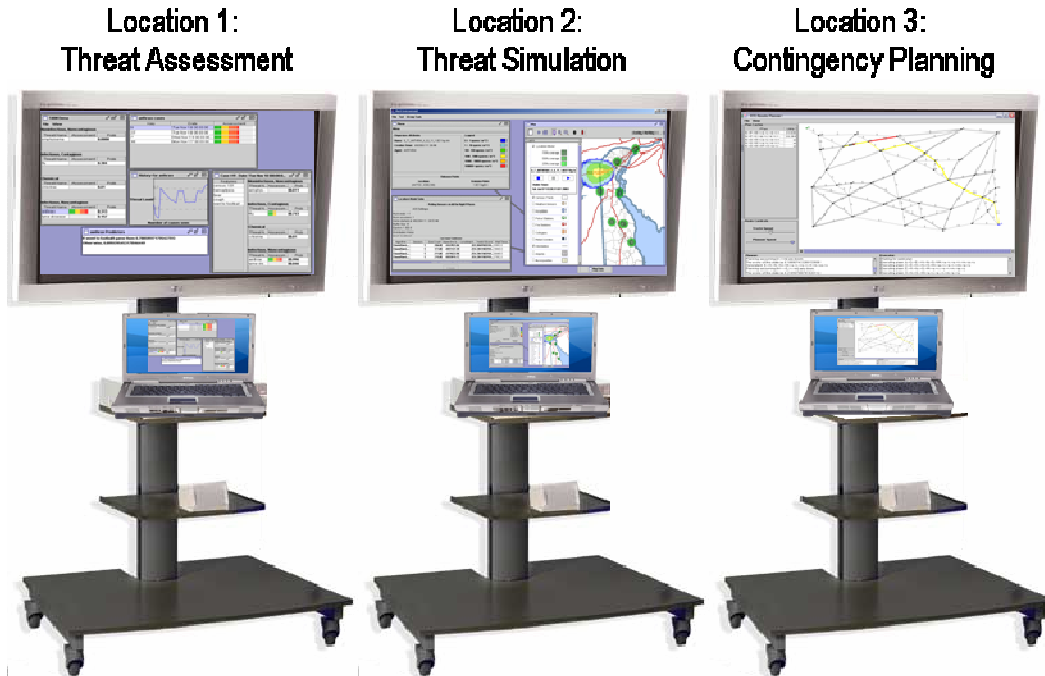
A problem with the wall-based IKW is that such a fixed arrangement in a room is difficult to deploy in a short amount of time or where there is not a blank wall with controlled lighting. It is also not mobile. QLI developed a second-generation approach to the interactive knowledge wall by assembling portable COTS components into a mobile configuration that allows rapid deployment (cf. Figures 34 and 35). The portable knowledge wall is disassembled by two people in a matter of minutes, easily transported in a minivan, with room for several responders or domain experts, and is reassembled again in minutes. The individual control units can be rolled easily by one person from one room to another. In particular, several control units can be placed side by side when interacting in a larger group situation, and then wheeled apart when the group breaks up into subgroups for more interactive sessions.

The components of the PCUs are as follows:

- 52" Plasma TV Screen
- High performance Laptop (DELL XPS)
- Mobile Plasma TV Stand with shelves
- Wireless Router
- Wireless Keyboard and Mouse

Each PCU was assembled at a cost of less than \$5,000.





**Figure 35: Conceptual deployment of Portable Control Units**



**Figure 36: Portable Control Units displaying iKE**

### 6.3.11. Other Control Capabilities

Other IBWTP Control technologies noted in Figures 9 and 10 that were pursued, conceptualized, and developed, but not elaborated in further detail in this final report, include:

- **Ontogenesis:** Automatic generation of Java code corresponding to a given OWL-based ontology.
- **QRocket:** A framework for managing lifecycle of and dependencies among software components within an application.

## 6.4. Integration Capabilities

Work in Task 5, ‘Development of Integration Capabilities’ concentrated on developing, demonstrating, and enhancing technologies supporting dynamic integration of and collaboration among applications and humans in a distributed network.

The effort was primarily composed of three parts:

- Integration Framework
  - Multi-Agent Development Environment (MADE)
  - Distributed Application Framework (DAF)
- Multi-agent Tools & Techniques
  - Policy Management
  - Ontogenesis
  - Group Communication Service
  - Multicast Communication Service
  - Multi-Agent Management System (MMS)
- Glossary

The rest of this section outlines the primary work accomplished in the scope of this task.

### 6.4.1. Multi-Agent Development Environment (MADE)

MADE is the **Multi-Agent Development Environment** that enables other applications to exploit multi-agent technology. An agent can be thought of as an autonomous entity that interacts with other agents in a social manner and interacts with its environment by being aware of changes in the environment through sensors and making changes in the environment through actuators.

QLI designed and developed MADE by extending and enhancing the open source Java Agent DEvelopment framework (JADE) [21]. An Agent is specified by a collection of behaviors. Some of the enhancements QLI developed include:

- Ability to load an entire agent system directly from an XML configuration file, as opposed to manually launching an agent or series of agents and configuring everything at runtime or hard-coding it within a java object. The platform setup, container setup, method of communication, number of agents, location of agents,

composition of an agent, security level of the agents, timing of agents, and etc. are all easily configured within a simple XML file.

- Flexible and powerful subscription service to automatically notify agents upon the occurrence of events, such as the passing of time, the joining of users, the production or consumption of a resource.
- Ontology-based policy definition and enforcement.
- Finite State Machine (FSM) editor to allow GUI-based development of an agent's behavior. Transitions can be monitored at runtime and all behaviors are pluggable within any QLI Agent.
- Database connectivity behaviors for integration of any JDBC-enabled database into an agent system.
- Multicast Message Transport Protocol
- Improved registration and deregistration mechanisms.

#### **6.4.2. Policy Management**

QLI researched the applicability of existing frameworks for semantics-based policy specification and enforcement, including Rei [22], KAoS, and Ponder. QLI developed software modules for utilizing the semantic constructs of Rei to model users, resources, actions, and policies. QLI also implemented interfaces for reasoning over Rei-based representations of entities in the system. This allows for answering queries as to whether a particular domain action is allowed under a set of policies by a specific user, given a specific context. QLI incorporated this into the MADE platform for enforcement of policies at run time, making use of the credentials feature of JADE for user authentication processes in the enforcement architecture. The architecture centers around a Policy Manager that maintains a database of policies and instance of the Rei Engine to reason about registered policies. The policy manager reasons over policies and authorizes a pair of credentials and action as per requests by other agents in the agent system. Within MADE, a policy manager carries out the task of enforcing policies within one container. The policy manager gets policies corresponding to a policy enforcement request from the policy directory service within the agent system. The description of a policy consists of the level at which the policy is applicable (agent, container, or platform) and the set of action-user-context triples that the policy controls.

#### **6.4.3. Group Communication Service**

QLI implemented a Group Communication Service (GCS) package that enables software agents to send and receive messages based on membership to groups. This overcomes the difficulty of knowing in advance the identities and addresses of all recipients. To send messages to a group, the sending agent only needs to know the group name and the protocol name used for this service. To receive group messages, the receiver only needs to register itself with the group using the service. The GCS is an enhancement to the JADE agent framework.

#### **6.4.4. Multi-Agent System Management System (MMS)**

Multi-agent systems are commonly used in large scale, distributed problem domains. Because of the inherent difficulty in monitoring, testing, and debugging these domains, tools are extremely valuable to developers and administrators of such systems. The JADE framework for developing multi-agent systems provides a FIPA-compliant architecture solution used by many researchers in the field. Currently, the tools provided with the standard JADE distribution, notably the Remote Management Agent and the Introspector, do not adequately address the needs of development and administrative users. Specifically, they lack functionality in the areas of deployment, visualization, run-time inspection, and reporting. QLI developed the Multi-agent system Management System (MMS) to address these needs.

The MMS provides a graphical user interface (cf. Figures 37 and 38) to measure properties of a running agent including uptime, number of messages, origin, and assigned functions. The MMS enables the administrator of an agent system to visualize agents, activity levels, and communication flow between agents and to apply grouping and filtering to the visualization to allow easy administrative usage.

#### **6.4.5. Agent Glossary**

Within IBWTP, QLI developed a glossary of pertinent terms related to distributed systems. This was particularly useful to establish a common basis and grounding, as many terms in these areas are ambiguous and used in many different contexts to refer to difference ideas and research thrusts. The glossary drew upon and enhanced other glossaries and standards in the field, especially Web Services [23], Software Engineering [24], Process Engineering [25], and Multi-Agent Systems [26].

#### **6.4.6. Other Integration Capabilities**

Other IBWTP Integration technologies noted in Figures 9 and 10 that were pursued, conceptualized, and developed, but not elaborated in further detail in this final report, include:

- **Multicast Component:** A JADE Message Transport Protocol using standard multi-cast services.
- **Distributed Persistent Object Layer (DPOL):** A solution unifying access and management of stored objects by distributed applications.

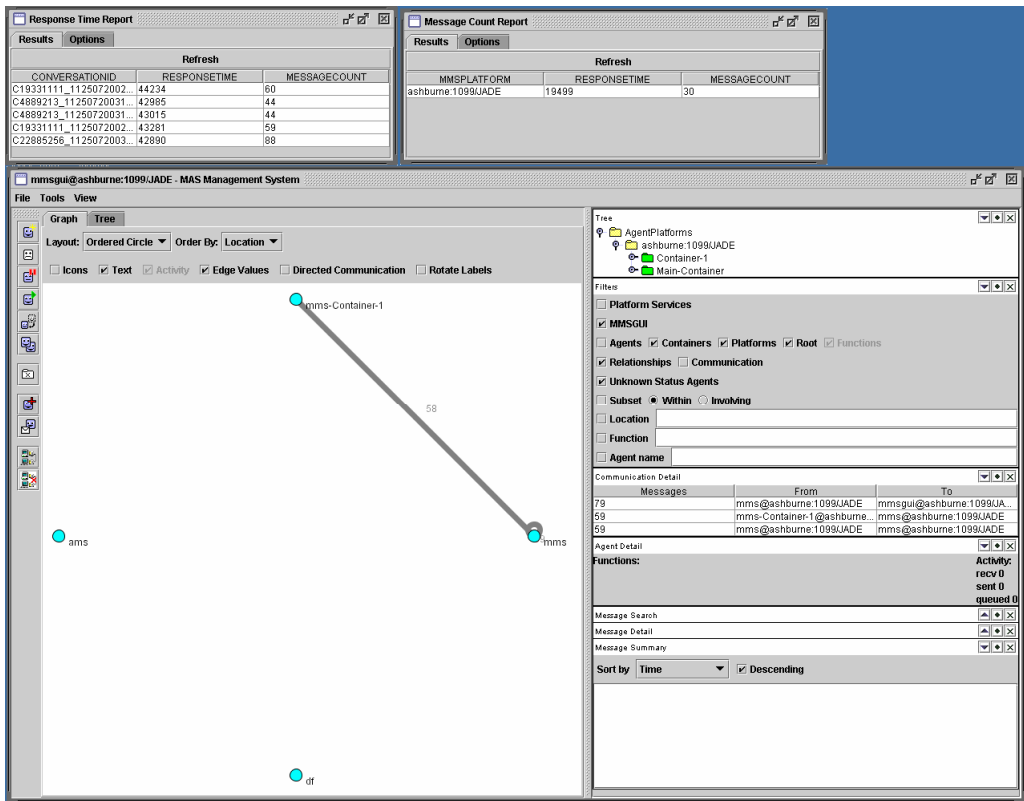


Figure 37: MMS Screen Shot 1

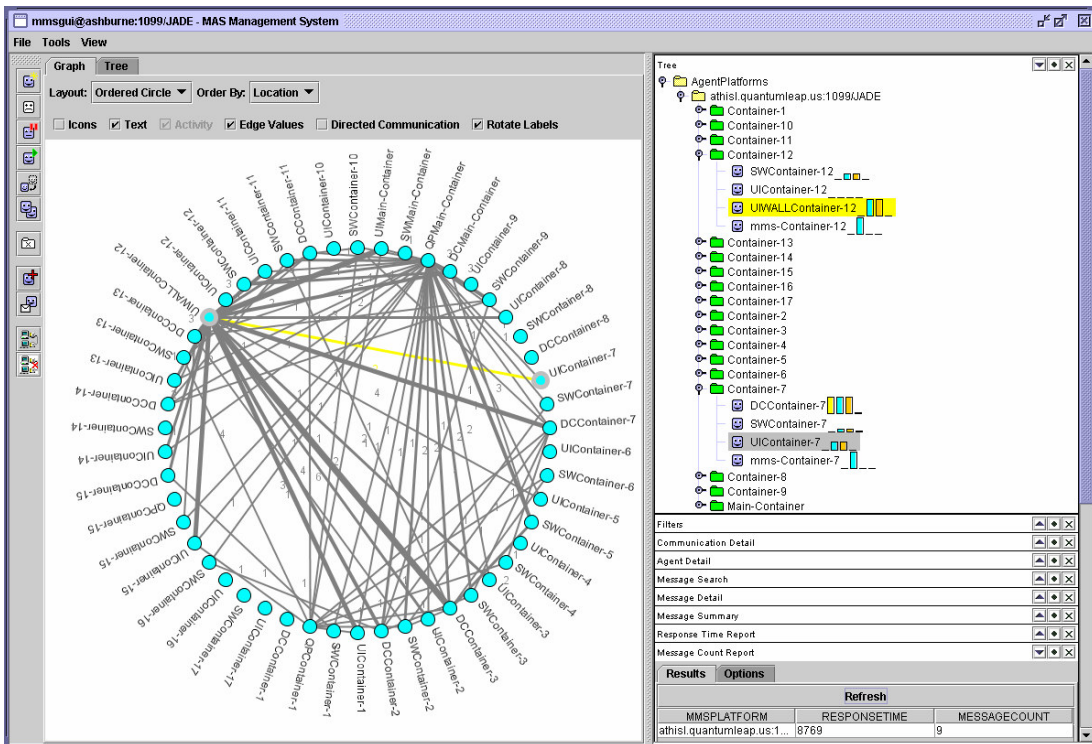


Figure 38: MMS Screen Shot 2

## 7. Conclusions

This section presents the conclusions about the relevance and applicability of the technology developed within IBWTP to support early detection and rapid response to biological and chemical threats, emergency management and response and the transformation of the Navy to meet the emerging threats in the 21st Century.

### 7.1. Applicability

#### 7.1.1. BioDefense and Emergency Response

The technologies developed under IBWTP were designed to support early detection and rapid response to biological and chemical threats, and are applicable to decision making and support in these areas in Homeland Security, Defense, and Agriculture. However, as the underlying technologies were designed with the goal of being as broadly applicable as possible to other areas, the results of the IBWTP project can be used not only in real-time decision making but also in simulations, functional exercises, and field exercises in training and preparation for emergency management and emergency response at the urban, regional, state, and federal levels.

Though development of the IBWTP technologies was aimed primarily at minimizing casualties due to a biological terrorist attack, there are many related uses of the technology, including the monitoring of naturally occurring disease events, both in humans, such as SARS, or in livestock (and subsequently in humans) such as HPAI Influenza A(H5N1). Though there is no adversary (known, to date) intentionally disseminating these pathogens, much of the same data, analysis, planning, and decision-making would be similar in responding to these natural disease threats.

Other applications of the technology apply to threats at a different scale (such as hospital-wide or regional health-care monitoring of nosocomial disease) or to different threats, such as monitoring and planning for extreme weather emergencies, or nuclear or chemical plant emergencies.

Specific areas of deployment of IBWTP technologies planned for future projects include:

##### *Urban Area Evacuation and Emergency Response Planning*

Apply QLI's Resource Allocation Framework and Realtime Adaptive Planning Framework developed under IBWTP to the problem of planning emergency response along with evacuation in major urban areas in the event of forecasted disaster. Specifically, the system will concentrate on the scenario of a tsunami approaching Honolulu with little warning. A future effort will target the development and installation of a prototype system at the Pacific Disaster Center (PDC).

##### *Integrated Collaborative Visualization for Emergency Response Planning*

Apply QLI's Interactive Knowledge Environment to provide an integrated real-time collaborative display of hazardous weather conditions, especially flooding, along with traffic condition monitoring for real-time monitoring and predictive futures analysis. This will be extended with biological/chemical/nuclear incidents, especially with those that

occur at Delaware's major industrial plants and Salem River nuclear power plants. A future effort will target the development, installation and evaluation of a prototype system at the Delaware Emergency Management Agency (DEMA).

Potential other IBWTP system users include Command and Control centers, such as those operated by

- U.S. Northern Command Operations Center, Surgeon General
- Pacific Command, Navy Region Hawaii Operations Centers
- Department of Homeland Security, Transportation Security Administration, Maritime and Land Security
- Delaware Division of Public Health and Delaware Emergency Operations Center – WMD exercise 2004
- State Bioterrorism Exercises, Tabletop and Full Preparedness Exercises

### **7.1.2. FORCEnet**

The core technologies developed in the Awareness/Action/Control/Integration framework are also applicable to real-world US NAVY application environments as they enable:

- Situational awareness within the Naval battlespace using knowledge discovery and data mining techniques across multiple on-line sources,
- Plan definition and generation to support distributed real-time execution of conflict-free and optimized Naval actions, plans, and procedures,
- Distributed collaborative information sharing and process execution across chain of Naval command, and
- Network-centric integration of heterogeneous applications and databases.

These technologies have broad applicability towards enhancing the Future Naval Capability (FNC) Knowledge Superiority and Assurance (KSA) in support of the transformation of the Navy to meet the emerging threats in the 21st Century. Specific areas of deployment of IBWTP technologies planned for future projects include:

#### *Condition-based Maintenance of Naval Equipment on DD(X)*

Apply QLI's Probabilistic Reasoning Toolkit and Causal Reasoning Engine to continuously monitor, assess, and diagnose condition of NAVY heavy machinery and equipment. It is planned to install and evaluate a prototype system at the DEI Group. This will enable early warning and failure diagnoses at equipment and fleet level:

- Required to achieve aggressive 60-80% reduced manpower on DD(X) and LCS vessels
- Real-time knowledge vs. "average lifetime" of asset
- Foundation for mission-readiness analysis
- Scalable from single small pump to entire systems of equipment

### *Realtime Adaptive Ballast Control in LPD-17*

Apply QLI's Real-time Adaptive Planning Framework to automate current manual system for dynamic load and ballast control in the LPD-17 ship and to plan for contingencies in emergency conditions. This requires controlling valve alignment, hydraulic pump units, air compressors, & ballast tanks to maintain dock at required levels for loading and launching. It is planned to develop, install, and evaluate a prototype system at L-3 Marine Systems using the LPD ECS simulation system. Further potential application areas at L-3 include assembly, deployment, and management of Rapid Response Teams for Advanced Damage Control and identification and resolution of problems with Propulsion System. All of these draw upon the reduction in workforce and the resultant need to optimally combine and use the capabilities of the sailors on hand.

The following are further examples of potential opportunities for application of the Awareness/Action/Control/Integration capabilities.

#### *Awareness*

- Situational Awareness
  - New models of adversary actions may emerge rapidly during conflicts: Causal Reasoning Engine supports automatic rapid creation, maintenance, & testing of models
  - Initial warnings of adversary operations through indirect evidence
  - Recent, local predictors of adversary actions can help direct additional intelligence resources, and aid in model adaptation
  - Terrorist/hostile identification, rapid maritime ID & tracking
- Virtual Data Warehousing
  - Enable integration of and access to all available information sources
  - Zero configuration – new sources exploited as soon as they join the network
  - Interpolation and projection provide standardized data for analytic tools

#### *Action*

- Direction of UAVs
  - Continual re-tasking of UAVS in light of new information
  - Dynamic UAV path determination to satisfy objectives, constraints
  - UAVs with different flight constraints (altitude, flight duration, stealth,)
  - Seamless adjustment to new situational context provided by operator or automated processing (damaged sensors, damaged or destroyed UAVs, added sensors or UAVs, changing parameters of coverage goals)
- Automatic distribution of sensors on UAVs
  - Heterogeneous sensors with different capabilities, costs
  - What combination of sensors in what numbers
- Optimize fixed sensor location
  - Permanently or as part of a convoy
- Sensor activation



- Fixed or roaming
- Lower manpower, more capable platforms (e.g. DD(X))
  - Automated routine control and coordination of assets enables humans to focus on critical decisions under context awareness
  - Embedded contingency plans for onboard systems, continual reconciliation of plans with current conditions
- Rapid re-tasking in light of emerging threats
  - High-speed mission changes require integrated and coordinated action with little advanced warning
  - Warm-start speculative execution enables initiation of dynamic plans while plan details are still being finalized

*Control*

- Experimentation and training
  - Rapid configuration of realistic & challenging scenarios, with realistic data, current analytic tools (Marine Corps Combat Development Command – MCCDC)
  - Multi-site, multi-server, multi-client training sessions
  - Adaptable to realistic failure modes – sporadic communication, off-line units

*Integration*

- Spiral development of new network-centric IT capabilities
  - Easy integration of new and legacy models, simulations, tools
  - On-the-fly linking of tool groups to meet emerging needs (Composeable FORCENet)
  - Rapid prototyping of new capabilities (Joint Battle Management Command and Control – JBMC2)
  - Distributed, collaborative analysis of situation, action alternatives

## 7.2. Publications

### 7.2.1. Patent Applications

Development of the technology underlying the following inventions was funded in part by the IBWTP project.

Table 3: Filed Patents		
ID	Title	Date
10/360,051	System, method, process, and article of manufacture for retrieving information, for representing and organizing known relationships, and for learning new relationships.	Feb. 6, 2003
10/773,638	Real time engine – a planning system for changing environments	Feb. 5, 2004

Table 3: Filed Patents		
ID	Title	Date
11/015,951	Automated Method and System for Generating Models from Data	Dec. 16, 2004
11/105,005	The Causal Reasoning Engine (CRE)	Apr. 12, 2005
11/195,963	Method and system for dynamic linking and coordination of distributed software components supporting collaborative manipulation and visualization.	Aug. 3, 2005
11/196,099	Integrated knowledge environment (IKE) - A method and system for dynamic linking and coordination of distributed software components supporting collaborative manipulation and visualization.	Aug. 3, 2005
11/196,162	Integrated knowledge environment (IKE) - A method and system for dynamic linking and coordination of distributed software components supporting collaborative manipulation and visualization	Aug. 3, 2005
11/433,314	Automated method and system for awareness, action, control, and integration	April 20, 2006
60/761,173	Extensible Bayesian network editor with inferencing capabilities	Jan. 23, 2006
60/762,337	System and method for optimization and constraint satisfaction of values for variables in a probabilistic model	Jan. 25, 2006
60/776,604	Constellation - A scalable System for Data Mining, Predication, Analysis, and Decision Support	Feb. 23, 2006.

### 7.2.2. Publications

The following publications in professional conference proceedings and journals were based on work accomplished within IBWTP.

Cowart, J. and Faulkner, E. The Adaptive Optimization Engine, INFORMS Annual Meeting, November 5-8, 2006, Pittsburgh, PA

Perry, B. and Van Allen, T. Causal Reasoning Engine: An Explanation-Based Approach to Syndromic Surveillance, Hawaii International Conference on Systems Sciences (HICSS) 2005

Vick, R. and Johnson, A. Prototyping the Emergence of Collaborative Knowledge, Hawaii International Conference on Systems Sciences (HICSS) 2005

Kalra, G. and Steiner, D. Weather Data Warehouse: An Agent-Based Data Warehousing System, Hawaii International Conference on Systems Sciences (HICSS) 2005

McNamara, Joe, "Cover Story: Log4j vs java.util.logging: Which logging library is better for you?" Java Development Journal. Volume 10, Issue 3, (March 2005): pp. 46-48

### 7.2.3. Technical Reports

The following internal QLI Technical Reports were created within IBWTP and are available upon request from Quantum Leap Innovations.

<b>Table 4: Technical Reports</b>			
<b>ID</b>	<b>Title</b>	<b>Authors</b>	<b>Date</b>
QLI-TR-2005-04	The Quantum Leap Adaptive Optimization Engine	Faulkner, E., Cowart J.	February 9, 2005
QLI-TR-2005-03	Cubic Spline Optimization	Faulkner, E.	September 27, 2005
QLI-TR-2005-02	Building Dynamic Bayesian Networks from Time Series Data Using Optimization	Faulkner, E.	September, 2005
QLI-TR-2005-01	Multi-agent system Management System	Atlas, James	August 26, 2005
QLI-TR-2004-07	Dynamic Population Distribution (DPD)	Zhou, Wei	August 31, 2004
QLI-TR-2004-06	The Implementation of The Environment Quality Monitoring (EQM) System	Zhou, Wei	July 19, 2004
QLI-TR-2004-05	Coupled Constraint Satisfaction and Optimization Problems	Faulkner, E.	September, 2004
QLI-TR-2004-04	The Weather Tool	Kalra, G.	March, 2004
QLI-TR-2004-03	The Scheduling and Planning Framework	Faulkner, E. and D. Cleaver	March 2004
QLI-TR-2004-02	Optimizing Vehicle Routing	Faulkner, E.	March 2004
QLI-TR-2004-01	The Causal Reasoning Engine	van Allen, T.	January 31, 2004

## 8. References

- [1] Joyce, G. et al., "Biodetection Architectures JSR-02-330", February 2003, <http://www.fas.org/irp/agency/dod/jason/biodet.pdf> (2005).
- [2] Centers for Disease Control, "National Immunization Program Guide: Vaccines and How They Work", <http://www2.cdc.gov/nip/isd/immtoolkit/content/products/NPIGuide.pdf>.
- [3] Smith B, Inglesby T, O'Toole T., "Biodefense R&D: Anticipating future threats, establishing a strategic environment", *Biosecurity and Bioterrorism*, 1(3), pp. 193-202 (2003).
- [4] Teich, J., Wagner, M., Mackenzie, C., Schafer, K., "The Informatics Response in Disaster, Terrorism, and War", *Journal of the American Medical Informatics Association* 9(2), pp. 97-104 (2002).
- [5] United States General Accounting Office, "Bioterrorism Public Health Response to Anthrax Incidents of 2001", Report to the Honorable Bill Frist, Majority Leader, U.S. Senate, (2003).
- [6] Garwin, R., "Nuclear and Biological Megaterrorism", 27th Session Of The International Seminars On Planetary Emergencies, August 21, 2002, <http://www.fas.org/RLG/020821-terrorism.htm> (2005).
- [7] Parachini, J., "Putting WMD Terrorism into Perspective", *The Washington Quarterly* 26(4) pp. 37-50. (2003).
- [8] Knobler, S., et. al, "Assessing the Science and Response Capabilities: Workshop Summary, 2002", Institute of Medicine (IOM), <http://fermat.nap.edu/catalog/10290.html> (2005).
- [9] Centers for Disease Control, "Bioterrorism Agents/Diseases", Centers for Disease Control and Prevention <http://www.bt.cdc.gov/agent/agentlist-category.asp> (2005).
- [10] Chalk, P., "Hitting America's Soft Underbelly: The Potential Threat of Deliberate Biological Attacks Against U.S. Agricultural and Food Industry", RAND National Defense Research Institute, <http://www.rand.org/publications/MG/MG135/MG135.pdf> (2004).
- [11] Centers for Disease Control, "Overview of Public Health Surveillance", Epidemiology Program Office Centers for Disease Control and Prevention, 4/12/2002, <http://www.cdc.gov/epo/dphsi/phs/overview.htm> (2005).
- [12] Kortepeter, M. et. al, "USAMRIID's Medical Management Of Biological Casualties Handbook - Fourth Edition", U.S. Army Medical Research Institute Of Infectious Diseases, (2001).
- [13] Tan, D., Stefanucci, J., Proffitt, D., Pausch, R., "The Infocockpit: providing location and place to aid human memory", *Proceedings of the 2001 workshop on Perceptive user interfaces PUI 01*, ACM Press (2001).

- [14] Tsui, F., Espino, J., Dato V., et al. "Technical Description of RODS: A Real-time Public Health Surveillance System", Journal of the American Medical Informatics Association (JAMIA) 10 pp. 399-408 (2003).
- [15] Tsui F., Espino J, Wagner M, et al., Key design elements of a data utility for national biosurveillance: event-driven architecture, caching, and web service model, Proceedings of AMIA Annual Symposium, 739-43, 2005.
- [16] Department of Defense, "Executive Summary: DoD-GEIS Implementation of Electronic Surveillance for Emerging Infections, Including Bioterrorism", US Department of Defense Global Emerging Infections Surveillance and Response System, <http://www.geis.fhp.osd.mil/GEIS/SurveillanceActivities/ESSENCE/ESSENCEexsum.asp> (2005).
- [17] Holzhauer, D. et al., "Building An Experimental Joint Battlespace Infosphere (YJBI-CB)", Proceedings of the Sixth International Command and Control Research and Technology Symposium (ICCRTS), Annapolis (2001).
- [18] Wilkins, D., "Using the SIPE-2 Planning System: A Manual for Version 6.1" SRI International Artificial Intelligence Center, July 1999, <http://www.ai.sri.com/~sipe/manual.pdf> (2005).
- [19] Atkin, M., Westbrook, D., and Cohen, P., "Capture the Flag: Military Simulation Meets Computer Games", Proceedings of AAAI Spring Symposium on AI and Computer Games, (1999).
- [20] Garvey, A., Lesser, V., "Design-to-time Scheduling and Anytime Algorithms", IJCAI Workshop on Anytime Algorithms and Deliberation Scheduling, 1995, <http://mas.cs.umass.edu/paper/169> (2005).
- [21] Telecom Italia Lab, "Java Agent DEvelopment Framework (JADE)", <http://jade.tilab.com/>
- [22] Kagal, L., Finin, T., and Joshi, A., "A Policy Based Approach to Security for the Semantic Web", Proceedings of the 2nd International Semantic Web Conference (ISWC2003), September 2003.
- [23] World Wide Web Consortium, "Web Services Glossary" <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>.
- [24] Lancaster University Computing Department, "Software Engineering Glossary", <http://www.comp.lancs.ac.uk/computing/resources/IanS/SE6/PDF/SEGlossary.pdf>.
- [25] Process Wave, "Process Engineering Glossary", <http://www.processwave.net/Glossary/glossary.htm>.
- [26] Foundation for Intelligent Physical Agents (FIPA), FIPA Presentation <http://www.fipa.org/docs/output/f-out-00111/>.

## 9. Index

- AACI, 7, 8, 9, 25, 26, 30
- AclOwlGrounding, 67
- Action, 1, 2, 3, 7, 8, 10, 11, 24, 25, 26, 28, 30, 39, 53, 62, 78, 79
- Agentacalc, 60
- AOE, 52, 53, 54, 81, 82
  - AOE Benchmark, 53
  - AOE Test Utils, 54
- AutoNet Express
  - BBN, 49
- Awareness, 1, 2, 3, 7, 8, 9, 10, 11, 24, 25, 26, 27, 28, 30, 39, 49, 52, 54, 62, 68, 78, 79
- Business Process, 59
- Clearboard, 61, 67
- Collman, 61, 63, 64, 65, 66
- Combinatorics, 54
- Constellation, 52, 81
- Control, 2, 3, 7, 8, 10, 11, 24, 25, 26, 29, 30, 32, 39, 47, 54, 57, 60, 61, 62, 69, 71, 72, 73, 78, 79, 80, 83, 84
- CRE, 40, 49, 50, 51, 78, 79, 81, 82
- CSCW, 11, 20, 29, 61
- D3M, 40, 51
- DAF, 61, 66, 73
- DEMA, 78
- Dispersion Model, 44
- Dispersion Model, 46
- DPD, 39, 48, 82
- DPE, 59, 60
- EAE, 39, 47
- Eclipse IDE, 32
- Emergency Response, 77
- EQM, 39, 47, 82
- FORCEnet, 2, 7, 11, 25, 39, 78, 80
- Glossary, 73, 75, 84
- Group Communication Service, 73, 74
- IBCWD, 1, 9, 39
- IDM, 39, 40, 41, 42, 43, 44, 52
- iKE, 2, 10, 29, 45, 47, 55, 61, 62, 63, 67, 68, 72, 81
- iKW, 10, 29, 61, 69, 70, 71
- Integration, 2, 3, 6, 7, 8, 11, 21, 23, 25, 26, 30, 39, 52, 73, 75, 78, 79, 80
- Interpolation, 18, 54, 79
- JADE, 73, 74, 75, 84
- KEE, 52
- Knowledge, 2, 3, 10, 11, 18, 29, 45, 52, 61, 69, 70, 71, 77, 78, 81
- MADE, 11, 30, 42, 59, 73, 74
- MenuManager, 69
- MMS, 73, 75, 76
- Multicast Component, 75
- Ontogenesis, 73
- Optimization, 1, 8, 23, 28, 52, 53, 81, 82
- Pathfinder, 53
- PCU, 71
- Planning, 1, 2, 3, 7, 20, 23, 25, 28, 41, 52, 53, 55, 56, 77, 79, 82, 84
- Plume Tool, 54, 62
- Policy Management, 73, 74
- PRT, 40, 49, 78
- QRocket, 73
- Random, 52, 54
- RAPF, 52, 53, 55, 56, 57, 79
- RPF, 53, 54
- Sensor Location Model, 54, 55, 62, 63
- SLAM-GA, 52
- Syndromic Surveillance, 51, 81
- TAM, 40, 50, 51
- TID, 39, 52
- VNC, 70
- Weather Tool, 39, 44, 45, 54, 62, 82

# REPORT DOCUMENTATION PAGE

*Form Approved*  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> 11-01-2007		<b>2. REPORT TYPE</b> Final Report		<b>3. DATES COVERED (From - To)</b> Jul 2002 - Jan 2007	
<b>4. TITLE AND SUBTITLE</b> Integrated Biological Warfare Technology Platform Intelligent Software Supporting Situation Awareness, Response, and Operations				<b>5a. CONTRACT NUMBER</b> N00014-02-C-0320	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b> 0602114N	
<b>6. AUTHOR(S)</b> Abbott, Frank, T. Johnson, Apperson, H. Prior, Stephen, D. Steiner, Donald, D.				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Quantum Leap Innovations, Inc. 3 Innovation Way, Suite 100 Newark, DE 19711				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b> QLI-TR-2007-01	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> Office of Naval Research ONR Code 313 875 North Randolph St., Suite 1425 Arlington, VA 22203-1995				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b> ONR	
				<b>11. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>12. DISTRIBUTION AVAILABILITY STATEMENT</b> Approved for public release; Distribution unlimited.					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b> Within the context of the Integrated Biological Warfare Technology Platform (IBWTP) program, Quantum Leap Innovations, Inc. (QLI) was tasked by the Office of Naval Research to develop, evaluate, and demonstrate novel technology supporting early detection of and rapid response to biological or chemical threats. This report provides an overview of the challenges QLI faced, the approach it took to creating the technologies, and some of the specific technological solutions in the areas of Situational Awareness, Course of Action Planning, Command & Control, and Data & Process Integration. It also presents the applicability of the developed technologies to areas other than biological response, such as Department of Homeland Security applications in emergency management, and Department of Defense applications in force transformation, especially regarding Future Naval Capability (FNC) Knowledge Superiority and Assurance (KSA).					
<b>15. SUBJECT TERMS</b> Biological Defense, Emergency Management, Force Transformation, Situational Awareness, Course of Action Planning, Command and Control, Probabilistic Reasoning, Knowledge Discovery, Collaboration, Multi-Agent Systems					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b> UU	<b>18. NUMBER OF PAGES</b> XX	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified			<b>19b. TELEPHONE NUMBER (Include area code)</b>