Defeating Terrorism

A Solution for Integrating our
National Intelligence Infrastructure

February 2002

MitoSystems, Inc.
3205 Ocean Park Blvd., Suite 180
Santa Monica, CA 90405
(310) 581-3600

Executive Summary

It has been clearly demonstrated that the intelligence community cannot reliably identify individuals or groups determined to inflict mass casualties or acts of extreme terrorism on Americans. We face a highly distributed threat that falls below the resolution of detection by the current isolated intelligence organizations each operating under outdated cold war methodologies. The joint development of an "intelligence supercomputer," that will analyze the same limited data sets from the same unreliable sources using the same analysis techniques, does little more than serve the comfort level of each agency's territorial domain.

This proposal summarizes an immediately deployable solution that provides the intelligence community with a *unified* infrastructure that can process all intelligence data of any format in real time. It has been matured over the past 11 years through more than 100 man-years of continuous engineering. This system level software architecture was engineered to manage and process any number and type of sources of intelligence data as well as improve the depth, accuracy and timeliness of the analysis process. There will be continued acts of terrorism like Oklahoma City and New York City unless there is an urgent resolve to make extensive fundamental changes to the methods and support systems within the entire intelligence community.

1. Introduction. The intelligence community is currently organized to deal with a limited set of adversaries that are known and stable. It does not have the information gathering, analysis and decision support tools necessary to deal with an enemy that is amorphous, distributed, and constantly changing. Worse yet, given their ready access to information on weapons technology, terrorists can have an impact vastly disproportionate to their size. A patient and determined individual or group can turn our means of transportation, energy production, water distribution, food processing, mail delivery, and other essential services and facilities into weapons of mass destruction. This document presents a unified intelligence infrastructure that enables the intelligence community to mitigate, nullify, or prevent such acts of terrorism through dramatically improved real-time information analysis and sharing, including the ability to rapidly adjust to changing threat scenarios.

2. Statement of the Problem. The following summarizes the critical barriers within the intelligence community that prevent it from reliably identifying and containing a highly distributed threat:

2.1. Lack of System Interoperability. The intelligence "community" is comprised of numerous independent agencies each possessing information systems that are not compatible. These agencies manage multiple, isolated "pools" of intelligence that cannot be shared, in many cases, even within the same agency. Independent intelligence data from each of these repositories could, if aggregated and analyzed, surface the intent to commit terrorism. Under the current pressure to find “quick” solutions to the numerous and obvious intelligence shortcomings, there is a risk of failing to address the underlying architectural issues necessary to truly solve the problem. There are thousands of existing “stovepipe” analytical tools that must be integrated into a common unified intelligence infrastructure. These tools would be far more effective if they were performing in concert.

Unfortunately, the various agencies have been independently developing, at great expense, their own information management and analysis tools for decades - each tailored to the respective agency's mission statement. The massive investment of time, money, manpower, and management that has gone into each independent system decreases the desire and ability for these agencies to efficiently share intelligence. The inability to identify or respond to a threat in a timely fashion is in great part due to this lack of interoperability, and terrorists can readily exploit this obvious weakness. The need for a community-wide unified intelligence infrastructure for real-time information analysis and sharing is now painfully obvious.

2.2. Constrained System Architectures. The vast majority of today’s computer applications - commercial and government - utilize a problem solving approach that has remained virtually unchanged since the beginning of computer science. We refer to these as “constrained” systems. To understand the concept of a constrained system and why it is a problem, consider the game of chess. The behavior and number of chess pieces is finite and well defined. Chess is perceived as a complex game due to the enormous combination of possible moves, counter strategies and outcomes. If you model the game's problem domain with a computer, the classic approach is to write a large, complex, rules-based application to explore the problem space. The result of a lengthy development process is an immensely complex application to solve what is a reasonably well-behaved and well-defined problem. Adding new chess pieces and new behaviors to the game represents a completely *new* problem domain and the *entire* application development process must be painstakingly repeated.

The real world equivalent of the chess analogy is the cold war. At that time, there was a well-defined set of adversaries with fairly well understood behaviors and patterns. Tasks were easily definable such as counting missiles. To manage this model, large, expensive, proprietary applications with data stored in a variety of databases seemed justifiable and sufficient. The opponents' behavior changed slowly so adjustments to the model could be accommodated over time. However, imagine trying to extend the constrained model to solve the infinitely more complex problem domain of terrorism that we face today. Terrorism presents the equivalent of a chess game of far greater magnitude where each of the opponent's pieces can continuously acquire new behaviors and strategies in real time. The constrained model would be quickly overwhelmed – it is too cumbersome to implement and cannot adapt. An alternate and far more flexible "unconstrained" approach must be taken.

2.3. Outdated Intelligence Cycle. The current intelligence cycle has undergone very little change since the 1950's. The cycle starts with an intelligence consumer making a formal request for information. The resources are then defined and committed to support the data collection process. Once sufficient data is collected, the analysis process begins. The results of the analysis are then packaged (usually hard copy) for presentation and dissemination to the intelligence consumer(s). Consider the task of determining how many missiles the Soviets have deployed in a particular region. Because it takes a long time to build and deploy missiles, this intelligence cycle does not have to be remarkably fast. Today, however, we are faced with *far* shorter reaction times. A terrorist can go through numerous decision cycles before the current intelligence cycle even *begins*. The amorphous and transient nature of terrorism makes it extremely difficult for the intelligence consumer to even know what questions to ask before the situation has already changed. This fluid and distributed threat requires a dramatically more efficient and responsive intelligence cycle, and the infrastructure to support it.

Fighter pilots, elite tactical military units, successful businesses, and even terrorists use a technique referred to as the Boyd cycle. This cycle consists of Observe, Orient, Decide and Act (OODA). The winner in a conflict is the one who executes this cycle faster. For the intelligence community this means providing the analysts with the tools necessary to "Observe" enough information from enough sources in real time so they can surface some irregularity, anomaly, or pattern of intent. It also means giving them the tools to spot and react, or "Orient," to changes in real time. A unified intelligence infrastructure would enable each analyst to rapidly reach and disseminate localized conclusions. Findings must be rapidly and effectively communicated to the appropriate authorities to "Decide" how to "Act" to thwart an impending event. A centralized resource like the "intelligence supercomputer" will simply not support the same level of responsiveness, performance and adaptability required in this environment. The solution presented herein can immediately support an extremely responsive OODA based intelligence cycle.

3. Elements of the Solution. The engineering focus has been on producing a complete solution at the architectural level. The immediate benefit will be the ability to unify many of the stovepipe applications already in operation. The deliverable is a mature unified infrastructure that provides all of the facilities and services necessary to install a functioning solution for the entire intelligence community, including client computers, server computers, distributed mass storage hardware, and numerous interfaces to support the collection of data in any format from any source. The solution can be characterized as highly scalable, extensible, flexible, media-rich, multilingual, real time, easy to use, easy to maintain and cost effective. The system has been matured over the past 11 years through more than 100 man-years of continuous engineering and is immediately available for deployment. The following summarizes the software architecture of the solution:

3.1. Ontology and Normalization of Data. The foundation of the architecture is the use of an extensible approach for achieving data organization and extraction of knowledge from highly unstructured information. Called ontology, it is an explicit formalization of a data model representing actors, events, actions, and numerous other elements and the relationships between them. The process of converting heterogeneous, unstructured information into formalized, interrelated, structured data is called normalization. From the simplest perspective, the ontology process starts by taking in new raw unstructured data – usually some kind of observation such as a news story, e-mail, image, etc. – and extracting from it a set of data elements as defined by the ontology. These elements might include people, organizations, weapons, etc. The raw observation itself is normalized and stored for potential future access and re-mining of attributes. Once the data is normalized, it becomes accessible to the very rich set of software tools comprised as part of the architecture. Through this unifying characteristic of the solution all incorporated software tools can act *interchangeably* on the data. This also enables an unprecedented level of *automation* of data collection, entry, storage, retrieval, query, mining, visualization, analysis and dissemination. A mature ontology has already been developed for the intelligence community.

The most powerful and unique aspect of this offering is not just the efficiency and elegance of an ontology, but the fact that an analyst can change it at any time. The solution includes the tools to modify and create new ontologies and have those changes be reflected throughout the system. Take the September 11 scenario where a commercial airliner has been used as a weapon. The analysts now realize they need to look for potential terrorists who have been through flight school training in the past five years. In current systems, a request must be made to modify the existing data model to include fields related to flight school. Sources must be identified and new code written to populate the appropriate fields. Finally, client user interface (UI) modification is needed to access and search this new information. The whole process could take months and would likely be quite expensive. In the new approach, analysts simply change the ontology to reflect these new elements and relationships and start to collect and normalize data from as many relevant sources as available. Moreover, since all captured data is preserved in its original form, all of the existing system data can be mined again in light of the new ontology. All of the tools used for data collection, entry, storage, retrieval, query, mining, visualization, analysis and dissemination automatically understand and reflect this new ontology. Such a change can percolate through a massive system in a matter of hours. This represents the kind of system-wide responsiveness required to identify and adapt to such a rapidly changing threat.

Another dramatic advantage of this approach is that it eliminates the dependency on fixed data structures. The ontology is the mechanism through which rich connections are made to highly time variant, unstructured data. The software tools do not have a direct reliance on or prior knowledge of the details of the structure used to store a specific type of normalized datum. Any software tool can handle any normalized data type. This allows the architecture to support any number of data sources, types and formats. For example, the following are all currently supported by the deliverable because of this flexible, extensible and dynamic approach to organizing data:

- Audio
- Video
- News Feeds
- Photowire Feeds
- Satellite Imagery
- Specialized Imagery
- Internet
- Published Data Sources
- Legacy Information Systems
- Manual Data Entry
- Documents
- Maps
- Covert Digital Intercepts

It is important to ensure programmatic overhead is kept to a minimum. Accordingly, wherever possible, the normalization process assigns all necessary “awareness” to the data. Any type of normalized data automatically "knows" how to read and write itself to and from disk, how to move across a network, how to respond to a query, what user interface to display, how to present itself and how to interact with a user; all database tables and the required UI are generated *automatically* from the data itself. In this model, the data, not the application, drives the entire information flow process.

The normalization process and its relationship to the designed ontology is vastly more complicated than can be put into a summary document. However, there is another powerful characteristic of this approach that is of significant importance to the intelligence community - the ability to rapidly add new data types. Consider the types of data a terrorist would generate as a trail of forensic evidence. Sources include the Internet, phone calls, radio and video transmissions, e-mails, faxes, dedicated data feeds, databases, chat rooms, banking systems, bulletin boards, web pages, etc. The tools and infrastructure readily enable seamless connection to new feeds, extraction of critical information, and integration with the composite knowledge infrastructure. It is vastly simpler, more achievable and more reliable to impart these characteristics to a type of *data* than it is to model an entire *problem domain*. The distinction to be made clear is that a pure, ontology-driven system can respond to internal or external change immediately, whereas constrained approaches cannot. This indicates the beginning of the end of massive, monolithic, complex, error prone, expensive and difficult to maintain applications.

3.2. Unconstrained Architecture. Our intelligence agencies continue to collect greater quantities of information, but are unable to process even a fraction of it. This will only get worse. The reason is that they employ constrained system architectures that are quickly overwhelmed by information overload. To understand this situation, let's consider a socialist model of government or an organization where the structure is largely centralized and top-down, control oriented. The command structure requires decision making to occur increasingly toward the top. Tremendous bottlenecks tend to develop and efficiency, progress, and change are inherently hindered. The same problem occurs with constrained system "control-flow" applications – a central program has to coordinate all the system input and output processes. The larger and more complex the application becomes and the more feeds it needs to process, the more it becomes saturated and cannot effectively process the load. Attempting to apply faster machines may ameliorate this problem but only on a marginally *linear* scale - the real problem is that information is *exponentiating.* The current model is inherently flawed and cannot deal with the information flow problem and distributed threat we are faced with today.

The problem can only be solved by an unconstrained system architecture. An unconstrained architecture must employ a "data-flow," rather than "control-flow," approach in order to be effective. This means that there is *no* central controlling program. It behaves as the equivalent of a capitalist model of computing – with a completely *decentralized* structure. In this model, parallel processing of information can be readily harnessed since there is no overhead of central control – each unit runs virtually autonomously from every other. Programmatic control is issued only at the very local level, enabling rapid change and adaptability to new feeds, increased quantities of feed volume, change in source language or format, etc. In fact, program control is devolved all the way out to the actual data level: CPU clock cycles are assigned according to the presence of data rather than according to programmed allocation. Programs no longer *seek* their inputs: they are *fed* them and only run when all the appropriate inputs have been supplied. This makes programming simpler, modular, and optimizes the ability to process information. By creating data-centric, rather than application-centric, processing models we have, in essence, inverted the computing paradigm.

The same inversion needs to occur at the user level. Stovepipe applications are intended to provide some predetermined set of functionality. Since these require custom UI, custom data input and output modules, and custom functionality, by the time these are developed and deployed, the needs of the user have likely changed. Accordingly, analysts are left with an inappropriate set of tools to deal with a changing problem. In essence, the programmatic control has been left to the system designers while the analyst – the real subject matter expert – has no influence. In an unconstrained system the analyst is in command. There is no central control "program" but there *are* local modules of control functionality - focused purely on the algorithm, not on the UI or input/output. To the analyst, these can be considered as functional building blocks – representing functions, algorithms, etc. The analyst can literally "wire" these building blocks, via a visual programming interface, to create the functional data process flow required. At this simplest level, the analyst can set up and modify any number of intelligent agents, or interest profiles, against incoming data feeds. He or she could also go as far as creating complete automated process flows where numerous feeds go through various function transforms and are generated into a series of visualizers for recognition of patterns – in *real time*. Hundreds of these more complex agents can be formed, launched into the system, shared collaboratively with others, and run automatically all the time. A group of analysts working as such can provide far greater synergistic learning than relying on the limited functionality imposed by control-oriented application designs. The act of empowering functionality out to the user on a distributed system has the exact equivalent effect as empowering decision making down in an organization: dramatically increased productivity, expanded bandwidth/scalability, reduced response time, and better decision making at the local level. The creative capacity of humans working in free collaboration *far* outperforms any model where the individual has been relegated to autocratic servitude.

3.3. Security. This solution is extremely modular. It is possible for an intelligence customer to integrate the certified security measures of choice into the architecture. This flexibility is intentional due to the number of different security policies and standards currently utilized across the intelligence community.

3.4. Customization and Implementation. MitoSystems has proven experience in this domain, having developed and installed large systems operating for many years. Extensive capabilities exist far beyond the scope of this paper (available upon request). In cases where cooperation with large systems integrators is required, MitoSystems can provide the necessary support to facilitate successful conversion of applications to this architecture.

4. Conclusion. The intelligence community has been developing technology solutions that rely on a fundamentally flawed approach. They were sufficient for relatively finite sets of adversaries and where slow response times were acceptable. The proponents of these constrained systems presume that even more complex software applications and faster machines are the answer to the information overload problem and distributed threats that we now face. The truth is that these systems are non-interoperable, tremendously difficult and expensive to maintain, and cannot adapt to changes in the environment. A different approach has to be taken. It must occur at the architectural level and these systems must be exceptionally flexible. The only approach is to develop unconstrained systems where compatibility is inherent, adaptability is the norm, and intelligence cycle times are faster than those of the enemy we wish to defeat. Such an architectural approach requires a long-term view, and would take many years to develop. Fortunately, one company has already done this. With over ten years of development and deployment in exactly this domain, MitoSystems has the architectural infrastructure already in place to transform the intelligence systems of the United States. This is likely to become the new strategic information weapon of the 21st century.

For further information, please direct all inquiries to Ted Whetstone, Business Development, at (310) 581-3600 ext. 228 or tedw@mitosystems.com