Static versus Dynamic Data Information Fusion analysis using DDDAS for Cyber Security Trust

Erik Blasch
AFRL/RIEA
Rome, NY
erik.blasch@gmail.com

Youssif Al-Nashif, Salim Hariri
NSF Center for Cloud and Autonomic Computing, The University of Arizona

Sponsor
AFRL/AFOSR

ICCS 2014
June 2014
PRESENTATION OUTLINE

- **Motivation** –
  - Information Fusion for Decision Making (Trust)
  - Understand the flow of information in Man and Machines
  - Information Fusion and DDDAS

- **Information Management** – Flow
  - Managers and QoS Flow
  - SOA-based DDDAS - DaaS

- **End to End** – Trust Metrics
  - Summary of metrics
  - Calculations

- **Potential Use Case Scenarios for DaaS** – DDDAS-based Crisis Management
High Level Information Fusion

Decompose problem into elements of LLIF and HLIF

Determine the user (situation awareness) and machine (computation)

Discussion on evaluation/visualization and projection
Blasch’s book: HLIF

High-Level Information Fusion
- Scenario
- Situation
- Object
- Observable
- Assessment

Low-Level Information Fusion
- Perception
- Sensation
- Awareness

TRUST
- Comprehension
- Projection
- Comprehension

Information/Resource Management

Situation Analysis Evaluation

Systems Design

Machine

Human
Information Fusion and DDDAS

Diagram showing the relationship between information fusion levels, scenarios, theory, measurements, software, analytics, visualizations, user, management, and interaction.
IF-DDDAS Design Challenges

• How do you handle, manage, and analyze “Big” amount of data, results, and users that are heterogeneous, dynamic, and continuously changing?

• How do you trust, quantify trust of users, data, results, etc.?

• How do you adopt, design, QoS, Quality of Protection (QoP) of such DDDAS environments?

• How can you integrate design stage and runtime stage to achieve continuous integration of design and runtime?
OUTLINE

• Motivation –
  • Information Fusion for Decision Making (Trust)
  • Understand the flow of information in Man and Machines
  • Information Fusion and DDDAS

• Information Management – Flow
  • Managers and QoS Flow
  • SOA-based DDDAS - DaaS

• End to End – Trust Metrics
  • Summary of metrics
  • Calculations

• Potential Use Case Scenarios for DaaS – DDDAS-based Crisis Management
Information Management Model

Manager Processing

Control Access and Audit Logs
Optimize Performance
Allocate Resources
Manager Processing
Process Requests
Life Cycle
Instantiate and Maintain Workflow

Security
Workflow
Quality of Service
Transformation
Brokerage
Maintenance
Data

Catalogue
Repository

L1
L2
L3
L4
L5
L6

Log Transactions

Producers

Formats and Standards

Consumers

Erik Blasch, Salim Hariri, Youssif Alnashfi – ICCS14
Adapt DDDAS QoS, QoP at runtime
- DDDAS Analytics
- Resilient DDDAS
- Trustworthy DDDAS

Managed Information Object

*MIO* comprises a payload and metadata that characterize the object such as topic, time, and location.

---

**Actors**

**Managers**
- Manage Types
- Security Policy
- Resource Allocation
- Data Mediation
- Monitor & Audit
- Maintain Federations
- Maintain Currency

**Federates**
- Seamless Access
- Restrictions
- Mediation
- Integrity
- Information SLAs

**Producers**
- Publish
- Advertise
- Get Feedback
- Receive RFIs
- Retract

**Consumers**
- Search/Browse
- Subscribe
- Query
- Transform
- Assess
Distributed Information Management

Coalition Issues

- Acquisition cannot be standardized or synchronized.
- Universal data standards are unrealistic.
- Data standards version skew is the norm not the exception.
- Coalition information spaces must adapt quickly to new partners and processes.

Information Management Best Practices

1. Information Sharing
   - Package information for sharing.
   - Adopt common syntax and semantics for common information attributes such as location, time, and subject.

2. Reduce Complexity
   - Adopt dedicated IM infrastructure
   - Create simple ubiquitous common services

3. Control and Flexibility
   - Common Information Space
Proposed Solution: DDDAS as a Service (DaaS)

Diagram showing the integration of services such as Information Brokering Service, Submission Service, Dissemination Service, Repository Service, Query Service, and Online Storage.
OUTLINE

• Motivation –
  • Information Fusion for Decision Making (Trust)
  • Understand the flow of information in Man and Machines
  • Information Fusion and DDDAS

• Information Management – Flow
  • Managers and QoS Flow (trust is a QoS service)
  • Policies for Information Flow

• End to End – Trust Metrics
  • Summary of metrics
  • Calculations

• Potential Use Case Scenarios – DDDAS-based Crisis Management
Trust (Multidiscipline)

- Users
- Systems

Rempel’s stages, (Muir & Moray, 1996) postulated that

Trust = Predictability + Dependability + Faith + Competence + Responsibility + Reliability

Trust Dimensions

<table>
<thead>
<tr>
<th>Operator’s trust &amp; allocation of function</th>
<th>Quality of the automation ‘Good’</th>
<th>‘Poor’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusts and uses the automation</td>
<td>Appropriate Trust</td>
<td>False Trust (risk automated disaster)</td>
</tr>
<tr>
<td></td>
<td>(optimize system performance)</td>
<td></td>
</tr>
<tr>
<td>Distrusts and rejects the automation</td>
<td>False Distrust</td>
<td>Appropriate Distrust</td>
</tr>
<tr>
<td></td>
<td>(lose benefits of automation, inc. workload)</td>
<td>(optimize system performance)</td>
</tr>
</tbody>
</table>
Trust in Action

Trust (Communication)

- Networks
- Message Passing

Proposed Approach: Trust Analysis and Quantification

- Trust can be defined as a state of mind consisting of
  - Expectancy of what the trustee will do
  - Believe expected behavior will be performed
  - Willingness to take risk in trustee

- Trust can be based on reputation
  - Collect from collaborating and interacting entities

- Trust a factor of technology and computations

\[
\text{End-to-end trust} = f (\text{Trust in infrastructure}, \text{Trust in entities})
\]

Which can be as simple as:

\[
\text{End-to-end trust} = (\text{Trust in infrastructure}) \times (\text{Trust in entities})
\]
Trust Stack

Brokerage
- Policies Enforcement

Domain Trust Authority
- Collecting Raw Metrics
- Behavior Analysis Analysis (Situation Awareness)

Workflow
- Authentication and Authorization

Security
- Secure Communication
Domainless Trust Evaluation Architecture

Trust Authority

Mutual Authentication

$\text{ATM}_A$

$\text{ATM}_B$

Entity 1

End-to-End Communication

Entity 2
Domain Based Trust Evaluation

Cross-Domain Trust

Domain A

Trust Authority A

ATMP_A

ATM_A1

Entity 21

End-to-End Communication

Domain B

Trust Authority B

ATMP_B

ATM_B3

Entity 68

Mutual Authentication
ATM Functionalities

Trust Metrics between analysis and filtering

- Environment
- Entity
- Activity
- User
- Context

Monitoring → Situation Analysis → Vulnerability Analysis → Anomaly Analysis → Security Policy Analysis → Context-Based Filtering → Quantification → Trust Evaluation

- Trust Database
- Self-Trust Evaluation
- Machine-Trust Evaluation
- Peer-Trust Evaluation

Data → Collected Information → Directly Collected → Trust Metrics → Decision → Dissemination

Erik Blasch, Salim Hariri, Youssif Alnashfi – ICCS14
## Trust Metrics

<table>
<thead>
<tr>
<th>Category</th>
<th>Metric</th>
<th>Quantification</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Password Strength</td>
<td>$Q = \begin{cases} 0, &amp; \text{Password Length} &lt; 8 \ 0.1 + 0.9 \cdot \frac{\text{Password Length}}{\text{Maximum Password Length}}, &amp; \text{Otherwise} \end{cases}$</td>
</tr>
<tr>
<td>User</td>
<td>Days since last password change</td>
<td>$Q = \begin{cases} 0, &amp; #days &gt; \text{Maximum Number Of Days} \ \frac{#days}{\text{Maximum Number of Days}}, &amp; \text{Otherwise} \end{cases}$</td>
</tr>
<tr>
<td>User</td>
<td>Number of authentication failures</td>
<td>$Q = \begin{cases} 0, &amp; #\text{failures} &gt; \text{Maximum Number Of Allowed Failures} \ 1 - \frac{#\text{failures}}{\text{Maximum Number Of Allowed Failures}}, &amp; \text{Otherwise} \end{cases}$</td>
</tr>
<tr>
<td>User</td>
<td>Lock Outs</td>
<td>$Q = \begin{cases} 0, &amp; #\text{Lock Outs} &gt; \text{Maximum Number Of Allowed Lock Outs} \ \frac{#\text{Lock Outs}}{\text{Maximum Number Of Allowed Lock Outs}}, &amp; \text{Otherwise} \end{cases}$</td>
</tr>
<tr>
<td>Application</td>
<td>Developer Reputation</td>
<td>$Q = \frac{\text{Reputation}}{\text{Maximum Reputation}}$</td>
</tr>
<tr>
<td>Application</td>
<td>Who manages the software</td>
<td>$Q = \begin{cases} 1, &amp; \text{Global Administrator} \ 0.5, &amp; \text{Local Administrator} \ 0, &amp; \text{No Administrator} \end{cases}$</td>
</tr>
<tr>
<td>Application</td>
<td>Developer Reputation</td>
<td>( Q = \frac{\text{Reputation}}{\text{Maximum Reputation}} )</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------</td>
</tr>
</tbody>
</table>
| Application | Who manages the software | \( Q = \begin{cases} 
1, \text{Global Administrator} \\
0.5, \text{Local Administrator} \\
0, \text{No Administrator} 
\end{cases} \) |
| Connection | Number of hops | \( Q = \begin{cases} 
0, \#\text{Hops} > \text{Maximum Number Of Hops} \\
1 - \frac{\#\text{Hops}}{\text{Maximum Number of Hops}}, \text{Otherwise} 
\end{cases} \) |
| Connection | Number of discarded Packets | \( Q = \begin{cases} 
0, \#\text{Discarded Packet} > \text{Maximum #Discarded Packet} \\
1 - \frac{\#\text{Discarded Packet}}{\text{Maximum #Discarded Packet}}, \text{Otherwise} 
\end{cases} \) |
| Machine | Firmware version | \( Q = \begin{cases} 
1, \text{Up to date} \\
0.5, 1 \text{ Version Behind} \\
0, \text{Otherwise} 
\end{cases} \) |
| Machine | Shared Folders | \( Q = \begin{cases} 
1, \text{No Shared Folders} \\
0.5, \text{Shared User Folders} \\
0, \text{Shared System Folders} 
\end{cases} \) |
| Analyzer | Integrity Check | \( Q = \begin{cases} 
1, \text{No Problem} \\
0.5, \text{Problem in user data} \\
0, \text{Problem in system integrity} 
\end{cases} \) |
| Analyzer | Virus Alerts | \( Q = \begin{cases} 
1, \text{No Alert} \\
0.5, \text{Virus Found in a document} \\
0.25, \text{Virus Found in an executable} \\
0, \text{worm found} 
\end{cases} \) |
The trust of an entity is a function of its CIA: Confidentiality, Integrity, and Availability:

And since trust metrics are used to determine the values of the CIA components, we can use a function $h$ that will map the trust metrics to the CIA components to get the trust:

$$T(E) = f(Confidentiality, Integrity, Availability)$$

$$T(E) = f(h(Trust Metrics))$$

- Directly collected
- Trust Measured outputs
Determining Initial Trust for a system

Using NIST CVSS can determine the risk in a system. CVSS uses the following metrics to evaluate the score:
- Base Metrics
- Temporal Metrics
- Environmental Metrics

Potential vulnerability found (CVE-2013-0419)
CVSS Score is 7.6
Potential vulnerability found (CVE-2013-0423)
CVSS Score is 7.6
Potential vulnerability found (CVE-2013-0424)
CVSS Score is 5.0
Potential vulnerability found (CVE-2013-0425)
CVSS Score is 10.0
Potential vulnerability found (CVE-2013-0426)
CVSS Score is 10.0
Potential vulnerability found (CVE-2013-0427)
CVSS Score is 5.0
Potential vulnerability found (CVE-2013-0428)
CVSS Score is 10.0
Potential vulnerability found (CVE-2013-0429)
CVSS Score is 7.6
Potential vulnerability found (CVE-2013-0430)
CVSS Score is 6.9
Potential vulnerability found (CVE-2013-0432)
CVSS Score is 6.4
Potential vulnerability found (CVE-2013-0433)
CVSS Score is 5.0
Potential vulnerability found (CVE-2013-0434)
CVSS Score is 5.0
Potential vulnerability found (CVE-2013-0435)
CVSS Score is 5.0
Potential vulnerability found (CVE-2013-0438)
CVSS Score is 4.3
Determining Initial Trust for a system

Base Metrics:
- Exploitability Metrics
  - Access Vector
  - Access Complexity
  - Authentication
- Impact Metrics:
  - Confidentiality
  - Integrity
  - Availability

Temporal Metrics:
- Exploitability
- Remediation Level
- Report Confidence

Environmental Metrics:
- Collateral Damage Potential
- Target Distribution
- Security Requirements
The initial user trust is a function based on the user rank and reputation at the time of adding it to the system.

\[ T(E,C) = T(\text{User}) \cdot T(\text{Machine based on CVSS}) \]

Trust measurement takes the value from 0 to 1. 0 represents the distrust and 1 represents the blind or full trust.

We adopt the NIST standard SP 800 four levels of trust:
- Distrust: 0
- Low trust: 0.33
- Moderate Trust: 0.66
- High Trust: 1
Adaptive Trust

- Trust Flow
  - The trust value assigned to each component is not static and will be updated continuously.
  - The Trust Authority module is the one responsible for re-evaluating the trust at runtime.
  - The trust is measured per entity and the trust levels are between 0 and 1.

- The trust of an entity $E$ is given by:

$$T(E) \in [0,1]$$

- Interaction between entities is defined by a context $C$. And the trust for entities will be computed per context:

$$T(E,C) \in [0,1]$$
\( M_i \) denotes the collected Trust Flow metric, where \( i \) is the metric identifier. The function \( m_i() \) is a quantifying function that returns a measurement between 0 and 1 for the metric \( M_i \).

The trust for an entity will be computed using two types of trust:
- Self-measured trust and
- Reputation-measured trust.
Adaptive Trust (continue)

Trust is evaluated by the Trust Authority

Self-Trust Evaluation:

\[ T_s(E,C) = T(\text{ATM}_E,C) \cdot \sum_{i=1}^{L} I_i(C) \cdot m_i(M_i) \]

Peer-Trust Evaluation:

\[ T_p(E,C) = \frac{1}{K} \sum_{j=1}^{K} T(\text{ATM}_j,C) \cdot T_{p,E_j}(E_j,C) \]

C = Context (Mandatory Fields: Federal ID#; Optional Fields)
E = Evaluation Strategies / Enforcement (Rules – e.g.; Space, Time Policies)

- The values of the metric weight \( I_i \) for metric \( i \) is determined based on the feature selection technique, where:

\[ \sum_{i=1}^{L} I_i(C) = 1 \]
Adaptive Trust (continue)

- Determining the peer trust is based on the gain or lose in the measured self-trust.
- Assume that $E_1$ is communicating with $E_2$. We have the following facts after the communication:

$$T_s(E_1, C \mid \text{Before}), T_s(E_1, C \mid \text{After}), T(E_2, C \mid \text{Before})$$

Then:

$$T_{p,E_1}(E_2, C \mid \text{After}) = \begin{cases} \frac{T_s(E_1, C \mid \text{After})}{T_s(E_1, C \mid \text{Before})} > 1, \text{Increase Trust}(T(E_2, C \mid \text{Before})) \\ \frac{T_s(E_1, C \mid \text{After})}{T_s(E_1, C \mid \text{Before})} = 1, T(E_2, C \mid \text{Before}) \\ \frac{T_s(E_1, C \mid \text{After})}{T_s(E_1, C \mid \text{Before})} < 1, \text{Decrease Trust}(T(E_2, C \mid \text{Before})) \end{cases}$$
Based on the context and the type of operations, the end-to-end trust can be evaluated using three trust evaluation strategies: Optimistic, Pessimistic, and Average. The end-to-end trust for each strategy can be evaluated as follows:

- **Optimistic Trust Evaluation Strategy:**
  \[
  T(E,C) = \max(T_s(E,C), T_p(E,C))
  \]

- **Pessimistic Trust Evaluation Strategy:**
  \[
  T(E,C) = \min(T_s(E,C), T_p(E,C))
  \]

- **Average Trust Evaluation Strategy:**
  \[
  T(E,C) = \text{average}(T_s(E,C), T_p(E,C))
  \]
**Context**

- **Context ID**: The context ID consists of a mandatory part which is the application protocol that is used in establishing the connection and an optional part which consists of additional information such as location and objectives. Note: Optional Context ID doesn’t mean it is not required for connections, it only means that not all connections require this field.

  - **EX**: A fighter is trying to download Targets map.
    - **Context ID**:
      - Optional: Location, Operation

- **Context Enforcement**: The context enforcement is the process of enforcing security policies based on the context of the communication. It is handled by the ATM.

  - The context enforcement could have the following as inputs:
    - **Context ID**
    - **Entity ID**: in cased of the domain based trust, this is a combination of the entity ID, group, domain.
    - **Location**
    - **Trust Level**
Example of Context Enforcement

Assume $E_1$ is communicating with $E_2$, the following is a possible policy for the context enforcement on ATM$_1$ which is responsible for handling $E_1$ communications:

- if ($E_2 \not\in$ Rome Labs) then Deny
- if (Context $\not\in$ Acceptable Air Force Contexts) then Deny
- if ($T(E_2,C) <$ Moderate Trust) then Deny
- if (Location($E_2$) not in US) then Deny

Accept
Trust Metrics

- Trust (T), Context (C), Metric (M) – CYBER Example, ICCS2014
  (ATM specific to Communications)

\[ T_s(E, C) = T(ATM_E, C) \cdot \sum_{i=1}^{L} I_i(C) \cdot m_i(M_i) \]

\[ T_p(E, C) = \frac{1}{K} \sum_{j=1}^{K} T(ATM_j, C) \cdot \sum_{i=1}^{L} I_i(C) \cdot m_i(M_i) \]

<table>
<thead>
<tr>
<th>Trust Confidence</th>
<th>Trust Evaluation Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimistic</td>
<td>( T(E, C) = \max { T_s(E, C), T_p(E, C) } )</td>
</tr>
<tr>
<td>Average</td>
<td>( T(E, C) = \text{ave}{ T_s(E, C), T_p(E, C) } )</td>
</tr>
<tr>
<td>Pessimistic</td>
<td>( T(E, C) = \min { T_s(E, C), T_p(E, C) } )</td>
</tr>
</tbody>
</table>
• Motivation –
  • Information Fusion for Decision Making (Trust)
  • Understand the flow of information in Man and Machines
  • Information Fusion and DDDAS

• Information Management – Flow
  • Managers and QoS Flow (trust is a QoS service)
  • SOA-based DDDAS - DaaS

• End to End – Trust Metrics
  • Summary of metrics
  • Calculations

• Potential Use Case Scenarios for DaaS – DDDAS-based Crisis Management
DDDAS-based Crisis Management

- Battle Management
- Nuclear Disaster Management
- Terrorist/Accident Management

Management Domain

- Decision Makers
- Domain Experts

Operations Domain

- Air Force
- Police
- Firemen
- First Medical Responders

Actions

Sensors

Measurements

• Battle Management
• Nuclear Disaster Management
• Terrorist/Accident Management
• DDDAS Analytics for Cybersecurity
DDDAS Components as a Service (DaaS)
SOA-based DDDAS Environment

We are collaborating with:

- Youakim Badr, LIRIS Lab, INSA-Lyon, France
- Ilkay Altintas, UCSD

Erik Blasch, Salim Hariri, Youssif Alnashfi – ICCS14
We are collaborating, with Ilkay Altintas, UCSD to use Kepler to implement DDDAS workflows that can meet any QoS, QoP or trust requirements for mission critical DDDAS environments.
Resilient DDAS-based Models

- Resilient models are based on (DDDAS) – Info-Symbiotic Systems
- Unify computing and measurements

\[
\begin{align*}
S_n \otimes E_n &\vdash D_n \\
S_n \otimes f(D_n) \otimes E_n &\sqsubseteq S_{n+1}
\end{align*}
\]
Generalized DaaS

Business Requirements Model

Ad-hoc Composition Approach

Running processes

Infrastructure

Endogenous Changes
- QoS Model
- Tolerance Model
- Security Model

Exogenous Changes
- Contextual Information
- Business Logic
- Security Model

Design time

runtime
Generalized Resilient DaaS

Endogenous Changes
- Security Policy
- Context Model

Service Model

Risk Model

Annotation Model

Web Service Composition

Exogenous Changes
- Business Domain
- Security Objective
- Context Model

Business Requirement

Feedback

Generate

Affect
USE CASE SCENARIO – Crisis Management

## Access Control Policies

<table>
<thead>
<tr>
<th>S1: non-functional contract</th>
<th>S2: non-functional contract</th>
</tr>
</thead>
<tbody>
<tr>
<td>60°C $\leq$ Fire resistance $\leq$ 90°C</td>
<td>60°C $\leq$ Fire resistance $\leq$ 90°C</td>
</tr>
<tr>
<td>Physical strain $\leq$ 70 PPM</td>
<td>20 $\leq$ physical strain $\leq$ 70 PPM</td>
</tr>
<tr>
<td>2Ghz $&lt;$ Radio frequency $\leq$ 2.5 Ghz</td>
<td>2Ghz $&lt;$ Radio frequency $\leq$ 3Ghz</td>
</tr>
<tr>
<td><strong>Authentication: Kerberos</strong></td>
<td><strong>Authentication: Login</strong></td>
</tr>
<tr>
<td><strong>Transport: SSL</strong></td>
<td><strong>Transport: HTTP non-secured</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S3: non-functional contract</th>
<th>S5: non-functional contract</th>
</tr>
</thead>
<tbody>
<tr>
<td>60°C $\leq$ Fire resistance $\leq$ 90°C</td>
<td>60°C $\leq$ Fire resistance $\leq$ 90°C</td>
</tr>
<tr>
<td>Physical strain $\leq$ 100 PPM</td>
<td>physical strain $\leq$ 70 PPM</td>
</tr>
<tr>
<td>2Ghz $&lt;$ Radio frequency $\leq$ 2.5 Ghz</td>
<td>2Ghz $&lt;$ Radio frequency $\leq$ 2.5 Ghz</td>
</tr>
<tr>
<td><strong>Authentication: Kerberos</strong></td>
<td><strong>Authenticate via SAML with an</strong></td>
</tr>
<tr>
<td><strong>Transport: HTTPS</strong></td>
<td><strong>X.509 Certificate</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Transport: SSL</strong></td>
</tr>
</tbody>
</table>

Concluding Remarks

How to continuously adapt DDDAS environments to changes, occurring within/outside the DDDAS composition process and satisfying a wide range of security/trust constraints in dynamic environments

Solution: Resilient SOA- DDDAS Architecture (DaaS)

(Continuous integration of runtime and design time)
Back Up Slides
Continuous Behavior Analysis
User Cyber DNA

Erik Blasch, Salim Hariri, Youssif Alnashfi – ICCS14
User Cyber DNA