Data quality attributes in network-centric systems

Jorgen Hansson
hansson@sei.cmu.edu

Peter H. Feiler
Aaron Greenhouse

April 2006
Outline

Net-centric systems

Data quality attributes

Validating security/confidentiality of a software architecture
  • Bell-LaPadula
  • Representation in AADL
  • Analysis in OSATE

Conclusions
Net-centric systems

Systems-of-systems
- Distributed per say (tightly and loosely coupled)
- Heterogeneous

Complexity and assumption of the systems vary
- High-level information systems
- Computational power
- Network bandwidth
- Life-time of systems (battery)
- Real-time performance requirement vary
- Notion of consistency and correctness vary
  - Sensor networks as information providers introduce “imperfection”
Global Network & System View

Freshness? Confidence? Security?

Freshness: 10s Confidence: 70% Security: secret
Data Flow: Sensor Network to Application

**PHASE**
- Detect anomaly via sensors
- Push data
- Process data #1 (data fusion, filtering, aggregation)
- Distribute data to clients (clients pull data)
- Process data #2 and decide on action

**EXAMPLE**
- Intrusion detection
- Location data, time of reading
- Determine object’s speed, direction etc.
- Distribute data together with:
  - level of security
  - data accuracy
  - data freshness
- Decide on a counter action

© 2005 by Carnegie Mellon University
Data Quality Attributes

Temporal correctness
  • Use temporally coherent and valid input data.
  • Deliver data at the right time!

Logical correctness, i.e., accuracy/precision
  • Deliver correct data, where logical correctness is given by minimum requirements on accuracy/precision.

Confidence
  • Deliver/use data with adequate confidence.

Security/confidentiality levels
  • Ensure that data consumers are classified for the data.
Example

Sensor readings at time t0 → Sampling

C1 → C2 → C3 → A1

C1 → C4 → C5 → A2

C6 → C7 → C8 → A3

Feedback
Example cont’d: component view
Observations

• Ensuring end-to-end latency of a flow does not guarantee that fresh data was used, or that data constraints were enforced.

• Simple scenarios become complex
  - There are normally more data elements than there are components/tasks)
  - Modes add to additional complexity for analysis.

• Need for automated analysis
Specification of Data

Data element $D$ parameter

- Type of data: [base, derived]

- AVI (absolute data validity interval $\equiv$ external consistency)
  - $[x$ time units]; ($x=\infty$ data is valid forever or until next reading; $0 =$ inherits from input data)

- RVI (relative data validity interval $\equiv$ logical consistency)
  - [list of data elements with which $D$ should be consistent]

- Confidence parameter: [0-1]

- Accuracy parameter: [0-1]

- Security level: [0-$x$] (0=no security; $x=$ highest security)
Temporal Correctness

- AVI – Absolute validity interval: expresses the temporal interval during which a data object is fresh.
  - Static
  - Dynamic, e.g., 
- RVI – Relative validity interval: defined as the intersection of AVIs.
  - Ensures data objects are temporally coherent when

\[
RVI (t) \geq \max_{a,b} \{ RVI(a), RVI(b) \}
\]

Temporal coherence

RVI (relative data validity interval) \equiv logical consistency

\[ c = f(a, b) \]

AVI(c) = \[
\begin{cases} 
RVI(a, b) \\
\alpha, \alpha \geq RVI(a, b)
\end{cases}
\]

© 2005 by Carnegie Mellon University
Logical Correctness

Expressed as precision/accuracy/error

- Imprecise computation models
  - Mandatory/optional
  - Primary/alternative

- Epsilon-consistency
  - notion of introduced error, and must never exceed a defined threshold

- Data similarity
  - Two values x and x’ are similar iff x-x’=<threshold.

- Logical correctness can affect temporal correctness (AVI)
Confidence: WPSM and PAN Example

PAN – Personal Area Network
WPSM – The Wearable Sensor System
- Medical hub (embedded system worn on the belt)
- GPS system
- Fluid intake monitor
- Sleep performance watch
- Core temperate pill ingested
- Sensors, worn on the chest, for
  - Ballistic impact detection
  - Skin temperature
  - Heart rate
  - Respiration rate
  - Actigraphy
  - Body orientation

Physiological models

<table>
<thead>
<tr>
<th>Area</th>
<th>State</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal</td>
<td>Im. Action</td>
<td>90%</td>
</tr>
<tr>
<td>Hydration</td>
<td>Attention</td>
<td>60%</td>
</tr>
<tr>
<td>Cognitive</td>
<td>Normal</td>
<td>100%</td>
</tr>
<tr>
<td>Life signs</td>
<td>Normal</td>
<td>90%</td>
</tr>
<tr>
<td>Wound det.</td>
<td>Attention</td>
<td>80%</td>
</tr>
</tbody>
</table>
Confidence: WPSM and PAN Example

Confidence is a function of
- Model
- Latency (confidence decreases over time)
- State (some states can be determined with higher confidence)
- Environment (conditions may affect sensors, e.g., heat and humidity)

Important events require high confidence
- Normal requires 50%
- Attention requires 70%
- Immediate action 80%
Confidence Paradox

Trade-off/balance:
• High confidence requires (i) more sampling, (ii) more sensors etc, which generates more network traffic.
• More network traffic may (i) shorten battery lifetime, (ii) increase collisions and drops, (iii) increase latency => decreases confidence!!

Only consume more resources when you need to increase confidence:
• **Normal operating mode**: Low confidence = low power & low bandwidth
• **“Interesting events” mode**: High confidence = high power & high bandwidth
Dependencies

Logical correctness (accuracy)

Temporal coherence (correctness and latency)

Confidence
Security: Objective

Confidentiality concerns that sensitive data should only be disclosed to or accessed/modified by authorized users, i.e., enforcing prevention of unauthorized disclosure of information.

Objective: Model security attributes for an architecture to verify that data is properly accessed and handled by users and applications.

Means to achieve confidentiality include enforcing access control, perform encryption, partitioning of system

Confidentiality frameworks
- Bell-LaPadula framework: military applications
- Chinese wall framework: commercial applications

Representation of these frameworks is similar, but the analysis differs.
Bell-LaPadula: Subjects and Objects

In Bell–La Padula, subjects operate on objects.
- Subjects need permission, expressed as security level, to use objects.

The security level of a subject or object is a pair:
- Classification
  - Drawn from a partial order of classifications (e.g., unclassified < confidential < secret < top secret).
- Set of categories
  - Drawn from a set of labels (e.g., NATO, Nuclear, Crypto).

(Class1, Set1) dominates (Class2, Set2) if and only if
- Class1 >= Class2
- Set1 ⊇ Set2
Example 1

Minimum security level required is (secret, {a,b})

Uncontrolled sanitization as (conf, {a}) is dominated by max(class_i,cat_i), i=1..3

Error: O_4 is read-only
Warning: O_1 are O_2 shared more freely than necessary.
Example

Legend

- $o_1$ = Object (circle)
- Security level = $<\text{Level}; \text{Category}>$
- $s_1$ = Subject (box)
- 4 = Top secret
- a = project 'a'
- 3 = Secret
- b = project 'b'
- 2 = Confidential
- 1 = Classified
- Sanitized data flow = 0 = Unclassified

Example

Confidentiality

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>O1</td>
<td>read</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2</td>
<td>append</td>
<td>append</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O3</td>
<td>append</td>
<td>append</td>
<td>read</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O4</td>
<td>append</td>
<td></td>
<td>append</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>read</td>
</tr>
<tr>
<td>O6</td>
<td></td>
<td>read</td>
<td></td>
<td></td>
<td>append</td>
<td></td>
</tr>
<tr>
<td>O7</td>
<td>append</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>read</td>
</tr>
<tr>
<td>O8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>append</td>
<td></td>
</tr>
</tbody>
</table>

Error: $s_5$ does not dominate $o_1$

Warning: $s_3$ has unnecessarily high security clearance
Modeling and Validation

- Basic confidentiality principle
  - Security clearance

- Need-to-know policy
  - Category membership (e.g., project, division)

- Principle of confidentiality constancy
  - The security level of derived data dominates the security level of the input data

- Controlled sanitization of data
  - Reducing security levels is only performed by trusted components/ports
Modeling and Validation: Software Level

Determine the viability of a software architecture given confidentiality requirements of data objects and security clearance by users

- Processes, threads, ports

An architecture is viable if **subjects dominate objects**

- confidentiality is not breached
- subjects have enough security clearance to perform their function
1. Derive minimum security clearance of processes/threads based on confidentiality requirements of ports

2. Enforce that confidentiality requirements of data elements (ports) are satisfied by the security clearance of processes/threads

3. Derive access control matrix based on the security levels of processes/threads and data elements

4. Strong validation by detailed analysis of security levels of processes/threads and data elements and the access control rules represented as a matrix
Modeling and Validation: Hardware/Software Level

Determine the viability of a system (software architecture mapped to hardware) given confidentiality requirements of data objects and security clearance by users

- Processors, memory, bus, processes, threads

Analysis:

- Ensure processes and threads are mapped to appropriate hardware, communicate over secured channels, and reside/store data in protected memory
- Derive minimum security requirements on hardware components given a software architecture
Bell-LaPadula and AADL Mapping

In AADL, components pass data through **data ports and over connections**.
- Data is not represented explicitly.

Mapping Bell–La Padula to AADL:
- **Subject** → AADL Component
- **Object** → Port Feature
  - (Port is a proxy for the data that passes through it.)
Bell-LaPadula Access Modes

Access mode: How an object can observe/alter a subject.

Bell–La Padula has four access modes:
  • Execute — neither observe nor alter
  • Read — observation with no alteration
  • Append — alteration with no observation
  • Write — both observation and alteration

Mapping to AADL:
  • In port → Read access
    - Incoming data object is consumed
  • Out port → Append access
    - Outgoing data object is created anew
Security Levels in AADL

The security level of a component or port is represented by a pair of property values:

- **Security_Attributes::Class**
  - Drawn from a total order defined by an AADL enumeration type.

- **Security_Attributes::Category**
  - Drawn from a set defined by an AADL enumeration type.
Security Levels Example

Inputs in1 and in2 are consumed to create a higher-security output object.

ExampleSystem

Security_Attributes::Class => confidential;
Security_Attributes::Category => (A, B, C);

Security_Attributes::Class => unclassified;
Security_Attributes::Category => (A, B);

Security_Attributes::Class => unclassified;
Security_Attributes::Category => (B, C);

Security_Attributes::Class => confidential;
Security_Attributes::Category => (A, B, C);
Security Level Example

system ExampleSystem
features
  in1: in data port {
    Security_Attributes::Class => unclassified;
    Security_Attributes::Category => (A, B);
  };
  in2: in data port {
    Security_Attributes::Class => unclassified;
    Security_Attributes::Category => (B, C);
  };
output: out data port {
  Security_Attributes::Class => confidential;
  Security_Attributes::Category => (A, B, C);
};
properties
  Security_Attributes::Class => confidential;
  Security_Attributes::Category => (A, B, C);
end ExampleSystem;
Additional Mapping Details

The security level of a port is the exact security level of the data that passes through the port.
- In particular, it is not an upper bound on the level.
- Makes analysis more precise.

The security level of a component is its maximum security level.
- Right now: Also the current security level.
  - Problem: (simplification) Makes rules too conservative.
- Future: Use AADL flows to model current security level.
  - Pro: Enables more flexibility in modelling
  - Con: Makes modeling more expensive
    (must declare flows)
Checking Ports

The simple security property
  • The maximum security level of a component must dominate the security level of each of its ports.
    - Covers basic confidentiality and enforces need-to-know

The star property
  • The security level of an out port must dominate the current security* level of its component.
  • The security level of an in port must be dominated by the current security level of its component.

* Source of conservativism when maximum security level == current security level.
Checking Component Nesting

The security level of a component must dominate the security levels of its subcomponents.
  • (No direct analog in Bell–La Padula.)

Subcomponent is part of the internals of component.
  • Subcomponent computes on behalf of component.
  • Thus, component is accessing data too.
  • So, component must be authorized to use the data accessed by its subcomponents.
Checking Connections

Connections must connect ports with equal security levels.
- (No direct analog in Bell–La Padula.)
- Derives from
  - AADL connection semantics
  - Our use of ports as Bell–La Padula objects.

Data object that leaves an **out port**...

...is the **same** data object that enters an **in port**.
Security Analysis Plug-in (current status)

We are implementing the security analysis as a plug-in to OSATE:
- Checks simple security property
- Checks star property
- Checks nested components
- Checks connections
Analyzed System
Analyzed System: Errors

<table>
<thead>
<tr>
<th>Problems</th>
<th>Properties</th>
<th>AADL Property Values</th>
<th>Error Log</th>
<th>Progress</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 errors, 0 warnings, 0 infos</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Connection c3's source security level (confidential, {B, A}) does not equal the level of its destination (confidential, {C, B})
- The security level of out port result, (confidential, {B, A}), does not equal the level, (secret, {B, A}), of its containing component
- The security level of subcomponent dest, (confidential, {C, B}), is not dominated by the security level, (secret, {B, A}), of its containing component
Button to run security analysis

Results of analysis

- Connection c3's source security level (confidential, C, B) does not equal the level of its destination (confidential, C, B).
- The security level of output result (confidential, C, B), does not equal the level (secret, B, A), of its containing component.
- The security level of subcomponent dest, (confidential, C, B), is not dominated by the security level, (secret, B, A), of its containing component.
Impact on Latency

Key management
- Key transport (centralized)
- Key agreement (peer-to-peer)
- Preloading of network-wide keys (sensor networks)
- AADL: Model the key management as part of the communication.

Encryption protocols
- Computational cost increases
- Bandwidth usage increases
- AADL: Apply cost factor(s) with the security level to be applied to execution times and message size/bandwidth usage.
Security in Sensor Networks: Goals

- Preventing data disclosure to non-owners
  - Privacy, confidentiality
- Preserving data authenticity/message authentication
  - Data integrity and source authentication
- Preserving data availability
  - Protection from subversion in multi-hop networks with ad hoc routing

Constraints and limitations

- Battery energy
- Computational and communications capability
- Sensor networks are unattended
  - vulnerable to physical compromise by adversaries
Conclusions

• Confidentiality work is on-going (flows, sanitization, access matrix, generation of access matrix, minimum security levels of subjects, Chinese wall)

• Modeling temporal correctness of data (AVI, RVI) is paramount to ensure freshness (coming next)

• Accuracy, precision, and confidence are likely to be modeled and validated in a similar way.
  - Based on flows and modes