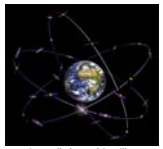
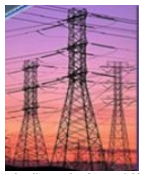
Enhancing Security in Ultra-Large Scale (ULS) Systems using Domain-specific Modeling

Joe Hoffert, Akshay Dabholkar, Aniruddha Gokhale, and Douglas Schmidt – Vanderbilt University

Trustworthiness Challenges for Ultra-Large Scale (ULS) Systems


Constellations of Satellites


Air Traffic Management


National/International Power Grid


Homeland Defense

Traits:

a) Services spanning systems of systems
b) Predominantly Pub-sub architectures
c) Require multiple Trustworthiness qualities, such as security, resiliency, and predictably

Challenges:

a) Trustworthiness issues tangled across different layers of middleware hosting platforms and multiple tiers
b) Multiple different middleware technologies with large number of configuration options
- Provide "what" (i.e., provide the mechanisms to deal with individual dimensions of trustworthiness)
- Don't provide "how" (i.e., don't provide the reasoning tools and techniques to address these)

Summary of Solution Approach: Use Model Driven Engineering (MDE) to
1. Modularize trustworthiness concerns,
2. Reason about the system, and
3. Synthesize "correct-by-construction" system provisioning metadata

Addressing Security via the Security Quality of Service (QoS) Modeling Language (SQML)

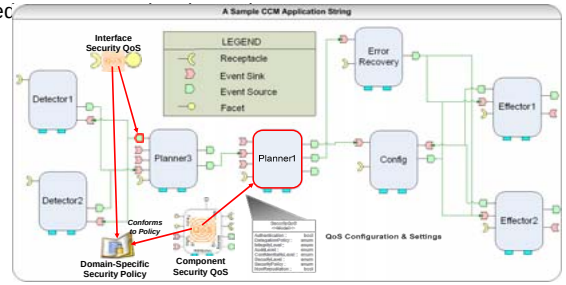
Model CORBA Component Model (CCM) role-based access control (RBAC) rules and rights at design time
- Eliminates tedious and error-prone role-based checking at run-time
- Allows definition of platform-specific rights families like a platform independent model (PIM)

Modularizes multilevel Security QoS provisioning through a configurable security policy framework
- Eliminates time consuming and inefficient runtime checks for consistency, conflicts, redundancy.
- Tailored to meet domain & application specific QoS requirements

Provide fine-grained as well as coarse-grained
- Facilitates flexibility and customization

Define annotations for configuring security in component middleware
- Allows middleware configurations specific to the needs of different parts of a system
- Enables secure application deployment through middleware configuration

Capabilities for defining Workflow/Business Process/Critical Path security



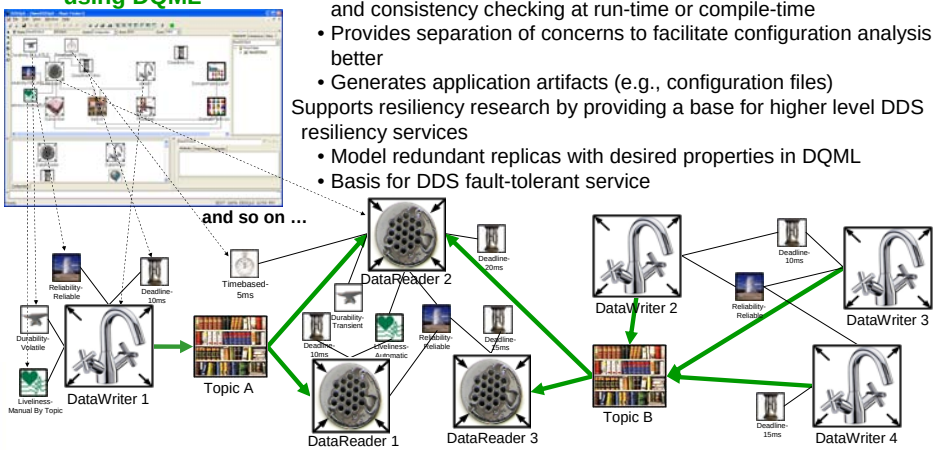
SQML uses aspect-oriented design to modularize security at various layers of the system

Addressing Trust & Resiliency via the Data Distribution Service (DDS) Quality of Service (QoS) Modeling Language (DQML)

Enhances trust by supporting intended QoS configurations at design time
- Eliminates complex, tedious, and error-prone QoS compatibility and consistency checking at run-time or compile-time
- Provides separation of concerns to facilitate configuration analysis better
- Generates application artifacts (e.g., configuration files)

Supports resiliency research by providing a base for higher level DDS resiliency services
- Model redundant replicas with desired properties in DQML
- Basis for DDS fault-tolerant service

Data flows as intended using DQML



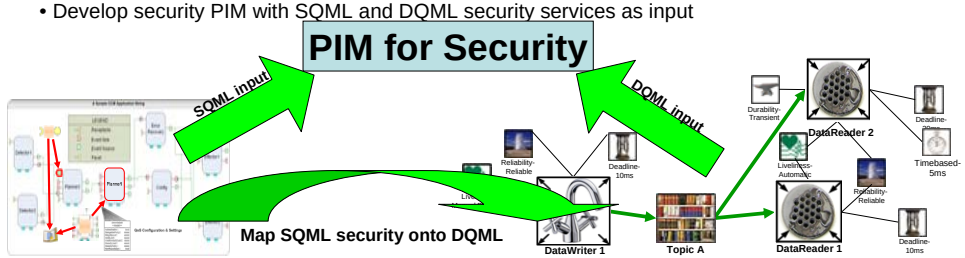
DQML uses constraint-checking for analysis and generates intended QoS metadata

Ongoing Research

http://www.dre.vanderbilt.edu/CoSMIC, http://www.cs.wustl.edu/~schmidt/CIAO

Creation of higher level trustworthy DDS services built on DQML work
- Discovery and documentation of DDS patterns
- DDS fault-tolerance service (e.g., using ownership/ownership strength, durability policies, multiple readers and writers, hot-swap and failover DDS pattern)
- DDS real-time data service (e.g., using deadline, transport priority, latency budget policies, continuous data pattern)

Generation of security mapping and security platform independent model (PIM)
- Map SQML's RBAC onto DDS security service
- Develop security PIM with SQML and DQML security services as input



Researching:
1. Integration of SQML and DQML with other analysis/benchmarking tools (e.g., Deployment and Configuration Engine (DAnCE), Component Workload Emulator (CoWorkEr) Utilization Test Suite (CUTS))
2. Higher level DDS service (e.g., fault-tolerance, real-time data, security)
3. Creation of Security Platform Independent Model (PIM) applicable across technologies