



Dynamic Languages and Architectures

GSAW-98

Vince Kovarik, Ph.D.

vincek@xgnt.com

A close-up photograph of satellite hardware, showing various connectors, cables, and metallic surfaces.

Corporate Background

- Building Satellite Command and Control Systems for 20 years.
- Long partnership with Navy Space program.
- Commercial command and control product, OS/COMET.

A close-up photograph of satellite hardware, showing various circular components, wires, and metallic surfaces.

Constellations Supported

- Currently managing the IRIDIUM™ constellation.
 - 51 satellites to date.
- Performing Command and Control for GPS constellation.
 - 27 satellites.

The title "IRIDIUM™ Architecture" is displayed in a large, bold, red, serif font. The "IRIDIUM" part is in all caps, while "Architecture" is in title case. A trademark symbol (TM) is positioned to the upper right of the "I" in "IRIDIUM".A close-up photograph of a satellite antenna or sensor, showing various circular and rectangular components and wiring, set against a blue background.

- Large multi-node distributed system.
- Object-oriented design and implementation using C++.
- Large-scale integration of COTS and previously developed software.
- Uses CORBA as the primary mechanism for distributed application interfaces.

A close-up photograph of a satellite component, showing a circular lens or sensor and various mechanical parts.

IRIDIUM™ Architecture

- OS/COMET used for Command and Control.
- Operational performance has been excellent.
 - All satellites have been caught on first orbit.
 - Three different launch vehicles.
- Experience in designing, building, and operating the system has surfaced several issues regarding distributed applications.

An abstract graphic in the top-left corner consisting of several overlapping, curved lines in shades of blue and white, resembling a stylized globe or a network diagram.

Lessons Learned

- The butterfly effect:
 - Small changes can have big consequences in distributed systems.
 - Immediate effects may be predictable.
 - Long-term effects, while deterministic, are unpredictable.
- Intractability of distributed debugging.
- CORBA is only part of the distributed solution.

A decorative image in the top-left corner showing a close-up, stylized profile of a person's head, possibly a woman, with a focus on the hair and facial features. The image is in shades of blue and white.

More Lessons ...

- Explicit memory management required by C++ is a double-edged sword:
 - Allows for greater control and greater efficiency (maybe)
 - Requires significant level of expertise to understand and use optimally
- Architectural impact analysis is not addressed in current tools.

An abstract graphic in the top-left corner consisting of overlapping, curved lines and shapes in shades of blue and white, resembling a stylized architectural or mechanical design.

Architectures Must Evolve

- Compensate for life-cycle brittleness and obsolescence.
- Incrementally enhance the capabilities of the system:
 - declarative representation
 - functional representation
- Design abstractions must be pushed up to the architecture level

An abstract graphic in the top-left corner consisting of overlapping, curved lines in shades of blue and white, resembling a stylized architectural or mechanical structure.

Dynamic Architecture Characteristics

- Interoperability
 - Common Object Request Broker Architecture (CORBA)
 - High Level Architecture (HLA)
 - Distributed Component Object Model (DCOM)
- Extensibility
 - Shared Libraries
 - Dynamic properties
 - Dynamic languages

An abstract graphic in the top-left corner consisting of blue and white curved lines and shapes, resembling a stylized globe or a network diagram.

Dynamic Languages

- Deployment Forms
 - Interpretive
 - Incremental Compilation
 - Threaded
- More than simply providing a dynamic language, the essential capabilities of programming languages must evolve.

Languages will model the relations, goals, interfaces, events, and be self-referential.

Languages model the problem and encapsulate data and behavior

Languages provide abstraction and encapsulation using data structures and control flow constructs

Languages model or directly access architecture of the computing platform.

Reflective

Semantic

Object-Oriented

Structured

Functional

Assembly

Machine

Hardware

Higher Level Abstractions

Today



A close-up photograph of a satellite component, showing a circular lens or sensor and various mechanical parts.

Conclusions

- While very successful, our experience in controlling the IRIDIUM and GPS constellations has brought several important issues to light.
- Distributed systems must be analyzed at an architectural level.

An abstract graphic in the top-left corner consisting of several overlapping, curved lines in shades of blue and white, resembling a stylized architectural or mechanical design.

Conclusions

- Dynamic languages and architectures are interrelated.
- Incremental changes will have ripple effects throughout the architecture (sometimes greater than the change itself).
- Key technologies needed:
 - Next generation languages
 - Architecture modeling languages/tools