Dynamic Languages and Architectures

GSAW-98

Vince Kovarik, Ph.D.

vincek@xgnt.com

Corporate Background

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

Constellations Supported

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

IRIDIUM™ Architecture

- 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.

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.

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.

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.

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

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

EXIGENT INTERNATIONAL

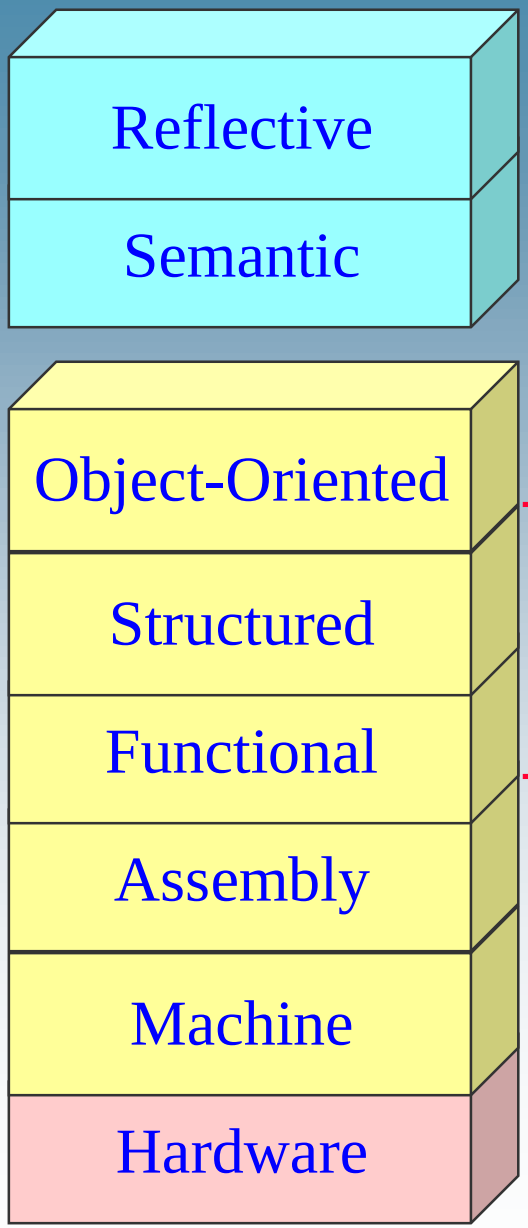
Dynamic Languages

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

EXIGENT INTERNATIONAL

Today

Higher Level Abstractions



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.

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.

EXIGENT INTERNATIONAL

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