TBASSCO Capabilities and Collaboration Plan (Year 1 and Preliminary Year 2)

Bob Neches, Ke-Thia Yao, In-Young Ko
Information Sciences Institute
University of Southern California
TBASSCO Coordination Plan
Summary and Roadmap

- **Products/Capabilities Produced for Others**
  - FY2001: Data flow scripting, component metadata schemas, scripting gauges, component insertion gauges *(see Foils 5-6, 8-11, 15)*
  - FY2002: Data flow to ACME translators, architectural selection gauges *(see Foils 7-9, 11)*

- **Demos**
  - FY2001: Four design and runtime scenarios centered around USC ISI’s GeoWorlds System at PACOM and JFCOM *(see Foils 14-17)*
  - FY2002: Further scenarios including runtime configuration (e.g. for installation in classified environment)

- **Capabilities needed from others** *(see Foils 9, 11)*
  - Semantic distance metric (Georgia); runtime performance gauges (BBN, Columbia/WPI, OBJS); runtime quality gauges (Columbia/WPI)

- **Requirements for other tools to use products** *(see Foil 10)*
  - Components described semantically using metadata, runtime performance and quality gauges expressed in group accepted XML standard

- **Evaluation**
  - Five criteria to be evaluated collaboratively by the organizations at PACOM and JFCOM using GeoWorlds *(see Foil 18)*
TBASSCO: Component-based Software Development

How to correctly build applications using readily available components

Component Database
- by function, by I/O data

Architectural Styles
- synchronous, asynchronous

How to add new components that are interoperable

How to configure and tune the component system to enhance runtime performance

System Architect/Administrator
TBASSCO Technical Approach

Template-based, multi-view abstraction ADL extensions that models and reasons about the semantics of component application systems.
Design-time Semantic Gauges for Application Scripting

Helps developers of new/improved systems combine component services appropriately

To select an alternative component

To select a component to be combined

To check the interoperability level of a connection

TBASSCO FY2001 Product:

Helps developers of new/improved systems combine component services appropriately

To select an alternative component

To select a component to be combined

To check the interoperability level of a connection

Design-time Semantic Gauges for Application Scripting

Helps developers of new/improved systems combine component services appropriately

To select an alternative component

To select a component to be combined

To check the interoperability level of a connection
TBASSCO FY2001 Product:
Design-time Semantic Gauges for Component Service Insertion

Helps developers of new component services evaluate whether to improve the interoperability of their offerings.

New Components

- Location to insert
- Semantic differences

Insertion Gauge

Language: same (Java)
Data: organization differ
Service: interface differ
TBASSCO FY2002 Product:

Design-time Semantic Gauges for Architecture Selection

Helps system administrators best configure applications for optimal performance in their local environment.

To select an architecture based on performance gauge feedback.

Design-time Semantic Gauges for Architecture Selection.
# TBASSCO Capabilities and Impact

## FY2001

- **Data flow scripting tools:**
  - Enable developers to specify how components interoperate
    - Aid combining alternative components and support gauge development
  - **Scripting:** Functional and data gauges
    - Guide in the composition of semantically correct customized applications
    - Help to avoid garbage-in-garbage-out errors
  - **Insertion:** Component insertion gauges
    - Indicate how new components are incompatible
    - Reduce developer time and effort required to add new components

## FY2002

- Translators to transform data flow descriptions to ACME
  - Interface TBASSCO’s contributions to other DASADA projects.
- **Selection:** Architectural level selection and deployment gauges
  - Help developers improve performance by adapting proposed system architectures to computing environment and loads.
Producer/Consumer Relationships

TBASSCO produces

- Semantic service and data flow description capability
  - BBN, Veridian and Object Services use them to describe semantic interoperability of their services
- Service event protocol specification based on semantic service description
  - Columbia/WPI verifies services are *conforming* during runtime

TBASSCO consumes

- Semantic distance metric to measure interoperability
  - Georgia provides metrics, i.e., clustering and factor analysis
- Runtime performance to tune architecture
  - Columbia/WPI, and BBN provide performance gauges
- Runtime service quality to select alternative services
  - Columbia/WPI provides quality gauges, i.e., size of search result
Requirements for Use

- Semantic level metadata must be supplied for components analyzed by scripting gauges
- Group agreement needed on runtime event definitions in XML Schema
- Group agreement needed on performance metric representations
Functionality to be Illustrated: Gauges and Probes in the Software Lifecycle

**Design Time Aids**
- **ISI**
  - Gauges to select interoperable components
  - Gauges to determine difficulty of adding new components
  - Gauges to adapt architecture to computing environment
- **Georgia State**
  - Semantic distance metrics

**Existing Software**
(From Library of Available Components)
- **ISI – GeoWorlds**
  - Object Services – SDC
  - BBN – Abstract search engine
  - Veridian – GIS map layers

**Run Time Aids**
- **BBN**
  - Network (bandwidth, latency) gauges
  - Uptime gauges
- **Columbia / WPI**
  - Protocol gauges (partial matching on event posets)
  - Run-time gauge plugin and modify

**Prospective Software**
(Same Library: Alternative Extensions, Compositions)
- **ACME ADL XML/FleXML**

**Requirements / Capability Descriptions**
- **ISI**
  - Semantic function and data descriptions
- **BBN**
  - Architecture requirement documents

**Object Services**
- Application profiling gauges and topology gauges on configuration, component usage
  - Component binding
  - Dead libraries
  - Versioning
  - Activity
Test-bed Application: GeoWorlds

- Large component-based system in use at PACOM & JFCOM
  - PACOM will serve as outside evaluator
  - JFCOM Experimental Battle Lab is potential second evaluator

- GeoWorlds architected from beginning as framework for adding components
- Geographic Information Systems plus Web processing
- Ops and intelligence uses, e.g.,
  - Mapping terrorist bombings
  - Locating recurring natural disasters
  - Investigating drug trafficking and piracy in various locales
GeoWorlds: Test-bed Application

Information Gathering

Web

Document Collection

Information Organization

Information Spaces

Asynchronous Service Invocation Architecture

Entry Flow

Event Flow

Job Manager

Service Selector and Job Listeners

Client Layer

Job Pool Layer

System Job Pool

Default Job Pool

JavaSpace Job Pool

CORBA Job Pool

Service Wrapper

Local Services

RMI Services

Socket Services

JavaSpace Services

CORBA Services
Lifecycle Scenarios

**Installation Time**

<table>
<thead>
<tr>
<th>User Observes</th>
<th>Causes</th>
<th>Gauges Do</th>
<th>How</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install script fails</td>
<td>Expected component not found</td>
<td>Config gauge identifies missing component</td>
<td>Compare installed config w/ Acme spec</td>
</tr>
<tr>
<td>Installation completes, but GeoWorlds doesn't work</td>
<td>Version mismatch</td>
<td>Config gauge identifies use of different version</td>
<td>Compare installed config w/ a good installation</td>
</tr>
<tr>
<td></td>
<td>Namespace error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Method invocation fails</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Demo Scenario 1:**

**Installation not fully operational**
- Assume
  - System deployed with specific probes/gauges
- Action
  - System administrator runs self-diagnostic tests
  - Probes extract information
  - Special GeoWorlds analyzer (provided by GeoWorlds) detects anomalies
Lifecycle Scenarios

Information Management

Scripting Time

Probes & Gauges

User Observes

GeoWorld script cannot be completed

Causes

I/O data semantic mismatch between components

Syntactic (interface) non-compliance between components

Dataflow violation (e.g., pipe output, page input)

GeoWorlds can't find appropriate data source

Gauges Do

Semantic gauges identify the mismatch & suggest intermediate components to resolve it

Syntactic gauges determine the cause of non-compliance and suggest adapters

Dataflow gauge detects mismatch & suggests a dataflow adaptor to allow

Semantic gauge subscribes data channels that meet requirement

How

Perform reasoning on I/O semantics and find components that make a semantic connection

Access to library of converters and wrappers

Access to library of available converters

Compare the data requirement and channel descriptions

Demo Scenario 2:

GeoWorlds script not working

Assume

- User Interface has probe palette to monitor component interactions
- User Interface gauges to interpret probe information

Action

- User selects to probe various connections
- Identifies a sequence where dataflow flawed
- User selects alternate component from User Interface
- Re-execute script correctly
Lifecycle Scenarios

GeoWorlds Application Testbed from ISI

Script Execution Time

<table>
<thead>
<tr>
<th>User Observes</th>
<th>Causes</th>
<th>Gauges Do</th>
<th>How</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoWorlds hangs during script execution</td>
<td>External data source has moved, disappeared, or become unresponsive</td>
<td>Connectivity gauge identifies the broken connection &amp; finds new URL or drops the source</td>
<td>Monitoring request &amp; response pairs and comparing timing with previous interactions</td>
</tr>
<tr>
<td></td>
<td>External data source has changed its (XML) interface</td>
<td>Change monitoring gauge determines an XML encoding has changed</td>
<td>Comparing the XML Schema used in a sequence of accesses</td>
</tr>
<tr>
<td></td>
<td>External service failure causes GeoWorlds script failure</td>
<td>Connectivity gauge identifies broken connection &amp; suggests alternate service</td>
<td>Monitoring request &amp; response pairs and comparing timing with previous interactions + knowledge of service alternates with similar interfaces</td>
</tr>
<tr>
<td>Script returns suspicious results</td>
<td>Spam site introduces flood of dubious responses</td>
<td>QoS gauge identifies growth in result size</td>
<td>Dataflow gauge identifies the path through the query that caused the increase</td>
</tr>
</tbody>
</table>

Demo Scenario 3: GeoWorlds hangs or returns unexpected result

- **Assume**
  - Probes already in place storing relevant info
  - Data source connectivity gauges
  - Query difference detection gauges are available

- **Action**
  - Gauge alerts user to abnormal situation
  - The user (sys admin) tracks down the wayward site and realizes that the URL has changed; reconfigures and continues
  - Observe growth in queries returned by GeoWorlds
  - Detect aberrant spam site that introduced errant info
  - Create new filter to eliminate spam sites and introduce into GW
  - Re-execute query to produce more accurate results
### Lifecycle Scenarios

#### Scenario 4: User service causes failure in GeoWorlds

- **Assume**
  - Probes already in place at service layer
  - Gauges exist to verify conformance to expected service behaviors

- **Action**
  - User notified through gauge that their service is non-conformant
  - The gauge delivers specific information for how to be compliant
  - User modifies the service as directed by the gauge
  - System functions normally

#### User Observes

<table>
<thead>
<tr>
<th>Causes</th>
<th>Gauges Do</th>
<th>How</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attempt to install new remote service fails</td>
<td>Should not happen - GeoWorlds will accept any service</td>
<td>Conformity gauges measure how well do I/O data and functionality conform to GeoWorlds data and function specifications</td>
</tr>
<tr>
<td>Users don’t know how difficult it is to add a new service</td>
<td>Mismatch with GeoWorlds data and service specifications</td>
<td>Semantic matching of components in reconfiguring the service</td>
</tr>
<tr>
<td>Installation Completes, but GeoWorlds doesn’t work with new service</td>
<td>GIS Data mismatch with request</td>
<td></td>
</tr>
</tbody>
</table>

---

GeoWorlds Application Testbed from ISI

![Diagram of GeoWorlds Application Testbed](image-url)
Evaluation Criteria

Evaluations to be performed collaboratively with GeoWorlds users at PACOM (and, tentatively, JFCOM):

- How *efficiently* GeoWorlds can be installed in different environments and its services deployed.
- How *easily* complex information management tasks can be scripted with assured semantic and syntactic interoperability.
- How *reliably* the scripts can be executed while maintaining desired quality level.
- How *quickly* the scripts can be evolved based on resource availability and requirement changes.
- How *efficiently* can new services be added to GeoWorlds while maintaining compatibility.