

# TBASSCO – Template-based Assurance of Semantic Interoperability in Software Composition

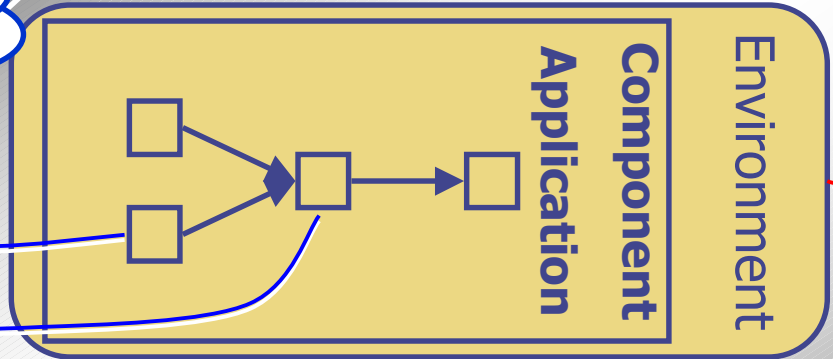
Robert Neches, Ke-Thia Yao, In-Young Ko  
USC Information Sciences Institute

DASADA Kickoff Meeting  
Santa Fe, New Mexico  
Sept 11-14, 2000



# Component-based Software Development

How can I build an correct application with these components?



**Application Developer**



**Component Database**

- by function, by I/O data

**Architectural Styles**

- synchronous, asynchronous

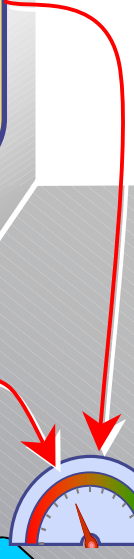
**Component Framework**

**Component Developer**

How do I make a new component interoperable?

How do I best configure this application for my local environment?

**System Architect/  
Administrator**





# Challenges

- ◆ Current tools help up to a point
  - Assume exact matching on data types and behaviors
    - ◆ Plug-compatible substitutions: yes
    - ◆ *Adaptation* of closely related components: no
  - Capture interface and method specifications
    - ◆ Syntax level composition: yes
    - ◆ Qualitative considerations in composition: no
      - implementation effort
      - performance
      - semantics
- ◆ Need better component / system metadata
  - Help developers understand what they're working with
  - Support advanced component selection and adaptation
  - Make gauge implementation feasible



# TBASSCO Hypothesis

We can:

- ◆ Develop a *maintainable, understandable* metadata framework for describing software components, and
- ◆ Support *useful* comparison of components wrt:
  - **Relative ability to perform desired functions;**
  - **Modifications needed to handle each other's i/o;**
  - **Relative performance of resulting alternative systems.**



# TBASSCO Sub-hypotheses

- ◆ “Simple” metadata representations are sufficient
- ◆ Maintaining component library is not unduly burdensome
- ◆ Effort of maintaining the component library is repaid when systems are assembled



# Experimental Approach

- ◆ Test with real-world component application: GeoWorlds
- ◆ Development tasks requiring assembly of sensible systems
  - Evaluate effort required to build the metadata
  - Build gauges that use objective measures
  - Validate the gauges' measures against expert assessments
  - Track the effort required to set up and use the gauges
  - Assess the expressiveness of metadata representation

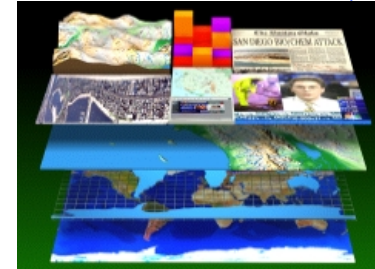


# GeoWorlds: Test-bed Application

- ◆ Large component-based system in use at PACOM
- ◆ Geographic Information Systems plus Web processing
- ◆ Framework for adding components
  
- ◆ Operations and intelligence uses
  - Mapping terrorist bombings
  - Locating recurring natural disasters
  - Investigating drug trafficking and piracy in various locales



# GeoWorlds: Test-bed Application



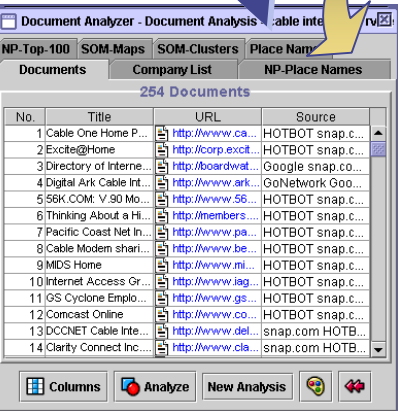
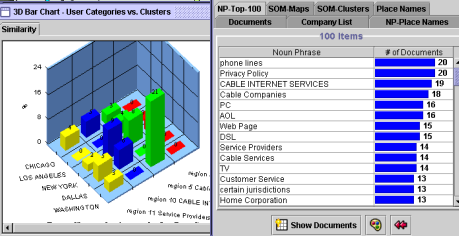
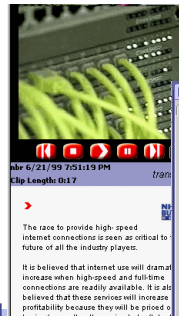
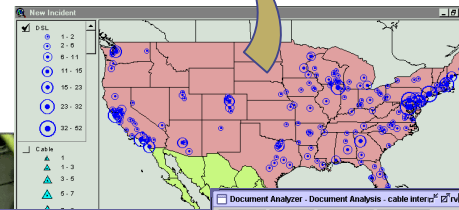
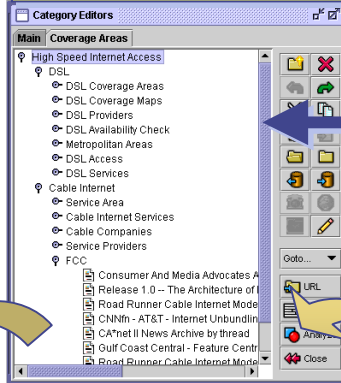
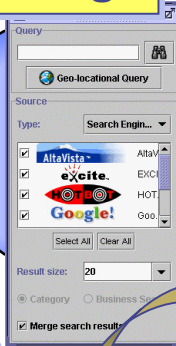
## Information Gathering

## Information Organization

## Asynchronous Service Invocation Architecture

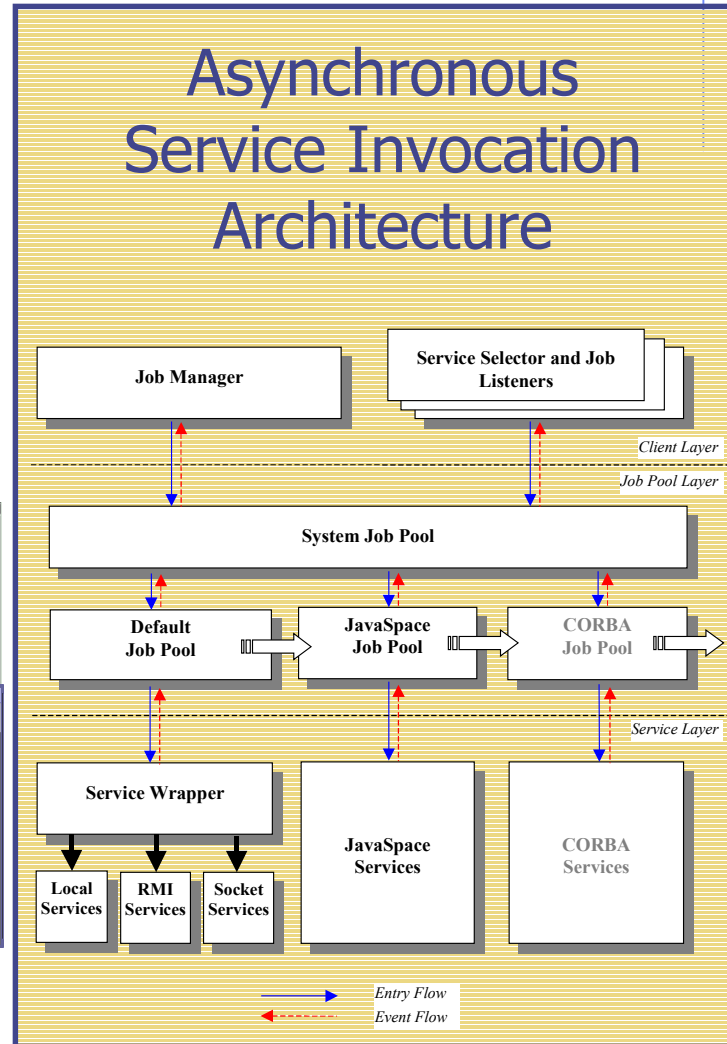


Document Collection



## Document Analysis

## Information Visualization





# Applications at PACOM

## Tested by Four Analysts

## Results

Natural disasters in China and India  
(for liaison to Pacific Disaster Center)

- ◆ *Handled 352 reports in 8 man-hours*
- ◆ *W/o GeoWorlds: "3-4 weeks of mundane, time consuming labor-intensive work."*

Six different analyses on Thailand/Burma:

- Thai Political Parties / Key Political Figures
- Thai Royal Family
- Thai Military and Politics
- Khmer Rouge Officials
- "Press Coverage on Narcotics" \*
- "Involvement of X with Drugs in Burma" \*

(\* The last two topics have been sanitized.)

- ◆ *Analyses were performed in about an hour each*
  - *Five of six found and organized collections of well over a thousand documents apiece*
  - *Sixth found 112 documents; w/o GeoWorlds, the analyst had found only seven*
- ◆ *Estimated effort w/o GeoWorlds: 3-5 days per analysis*

Patterns of terrorist bombings in the Phillipines

- ◆ *W/o GeoWorlds: "much more time consuming."*

Nationalist movements  
and nuclear testing incidents in Tahiti

- ◆ *Results forwarded to PACOM J2, recommended for inclusion in briefing to the CINC in preparation for his pending visit to that area*



# GeoWorlds Test-bed Application

- ◆ Illustrates common challenges
  - Restructuring new version releases
  - Integrating new functionality
  - Adapting for local environments



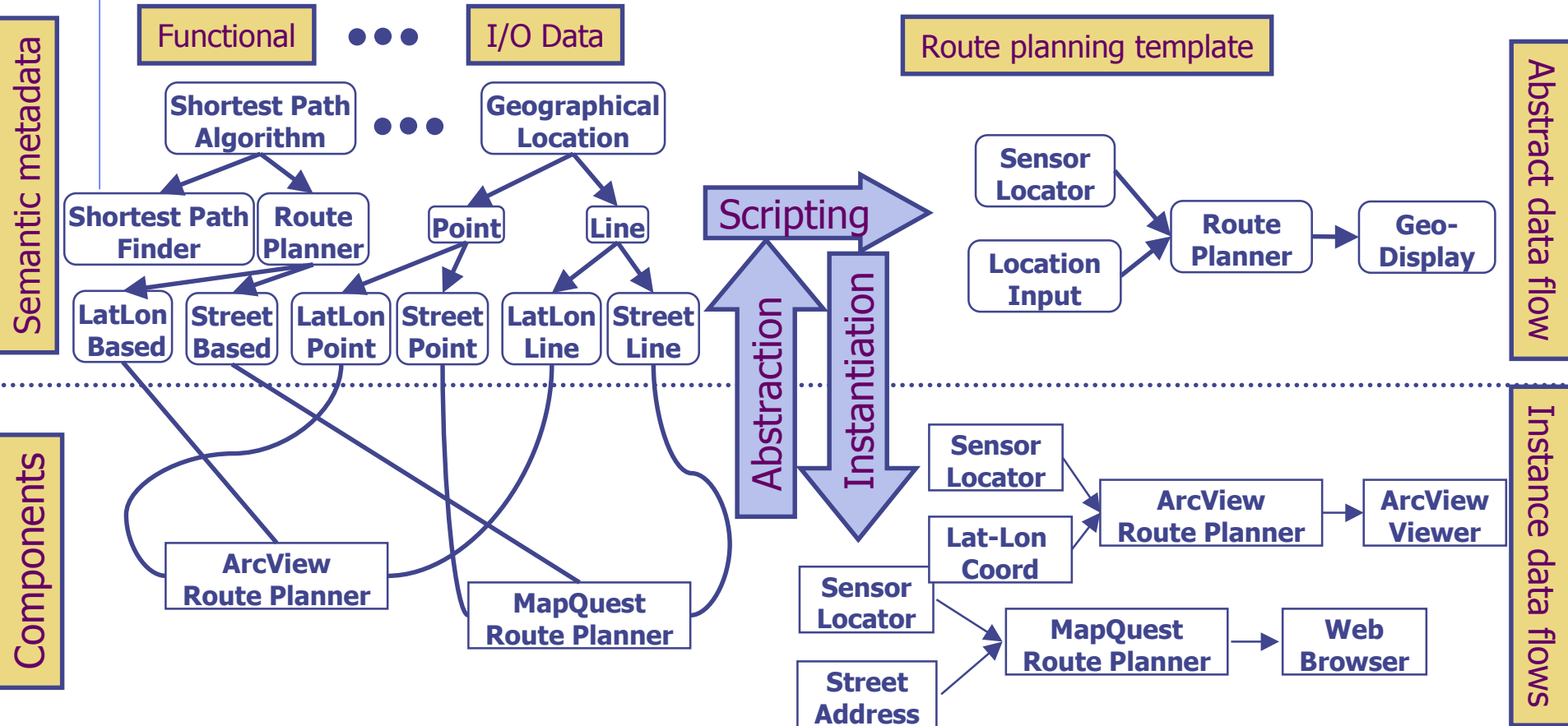
# TBASSCO Technologies and Impact

- ◆ **Data flow scripting language**
  - Lets developers specify how components interoperate
  - Aids combining alternative components, supports gauge development
- ◆ **Functional and data gauges**
  - Help compose semantically correct, customized applications
  - Attack garbage-in-garbage-out errors
- ◆ **Component insertion gauges**
  - Indicate how new components are incompatible
  - Reduce time and effort required to register new components
- ◆ **Architecture-level selection & deployment gauges**
  - Adapt proposed architectures to computing environment, expected loads
- ◆ **Translators: data flow descriptions to ACME**
  - Interface our work to other DASADA projects

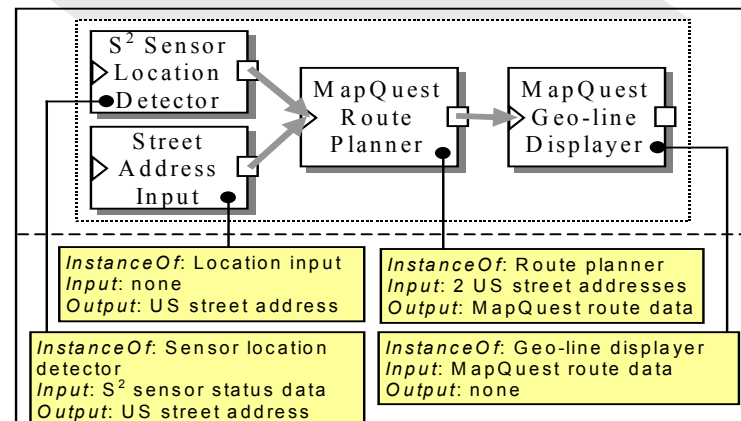
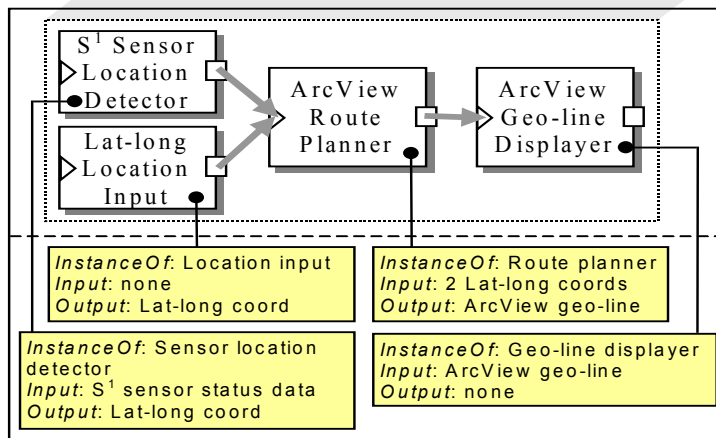
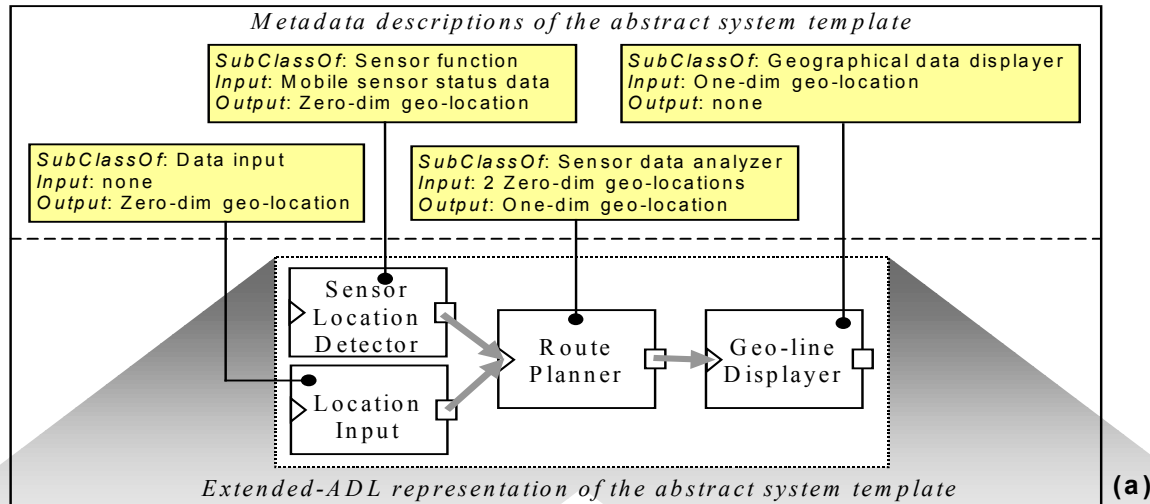


# Technical Approach

- ◆ Template-based, ADL extensions that model and reason about component application systems *semantics*



# Example: Semantically-based Application Scripting

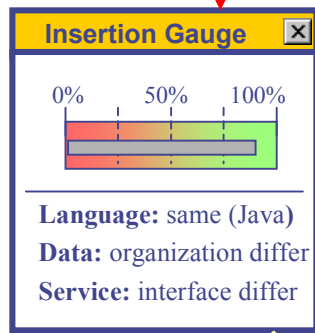
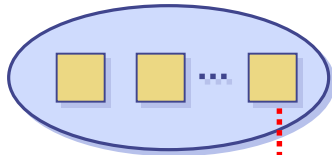




# Design-time Semantic Gauges

## Component Insertion

### New Components



- Location to insert
- Semantic differences

**InfoManager Resource Editor**

**File**

**Analytic** | Visual | User Input | Data Converters

- Analytic Service
  - KeywordExtraction
  - NounPhraseExtraction
  - NounPhraseBasedClassification
  - UAZ\_NounPhraseExtractor
  - CompanyNameExtraction
  - CompanyNameBasedClassification
  - GW\_CompanyNameExtractor
  - DocumentClassification
  - KeywordBasedClassification
  - PlaceNameBasedClassification

**Service Schema**

URI: <http://www.isi.edu/geoworlds> | anyNameBasedClassification

**Description:**  
Services of classifying documents based on company names extracted from the document collection

**Superclass:**  
KeywordBasedClassification  
CompanyNameExtraction

**Input Data:**

Name	Content	Structure
Input\$1	Docume...	

**Output Data:**

Name	Content	Structure
Output\$1	Compan...	FlatCate...
Output\$2	Compan...	ListOfD...

Done    Cancel

**Content Classes** | Organization Structure Classes

- Data Content
  - DocumentCollection
    - CategorizedDocumentCollection
      - KeywordBasedCategories
      - NounPhraseBasedCategories
      - CompanyNameBasedCategories
      - PlaceNameBasedCategories
      - DASHERCategory
      - YahooDocumentCollection
      - SearchEngineResultClassifiedBySc
      - DocumentClusters
      - DASHERCategory

**Data Schema**

URI: <http://www.isi.edu/geoworlds> | ipanyNameBasedCategories

**Description:**  
Document collection categorized by company names

**Superclass:**  
KeywordBasedCategories  
CompanyNameList

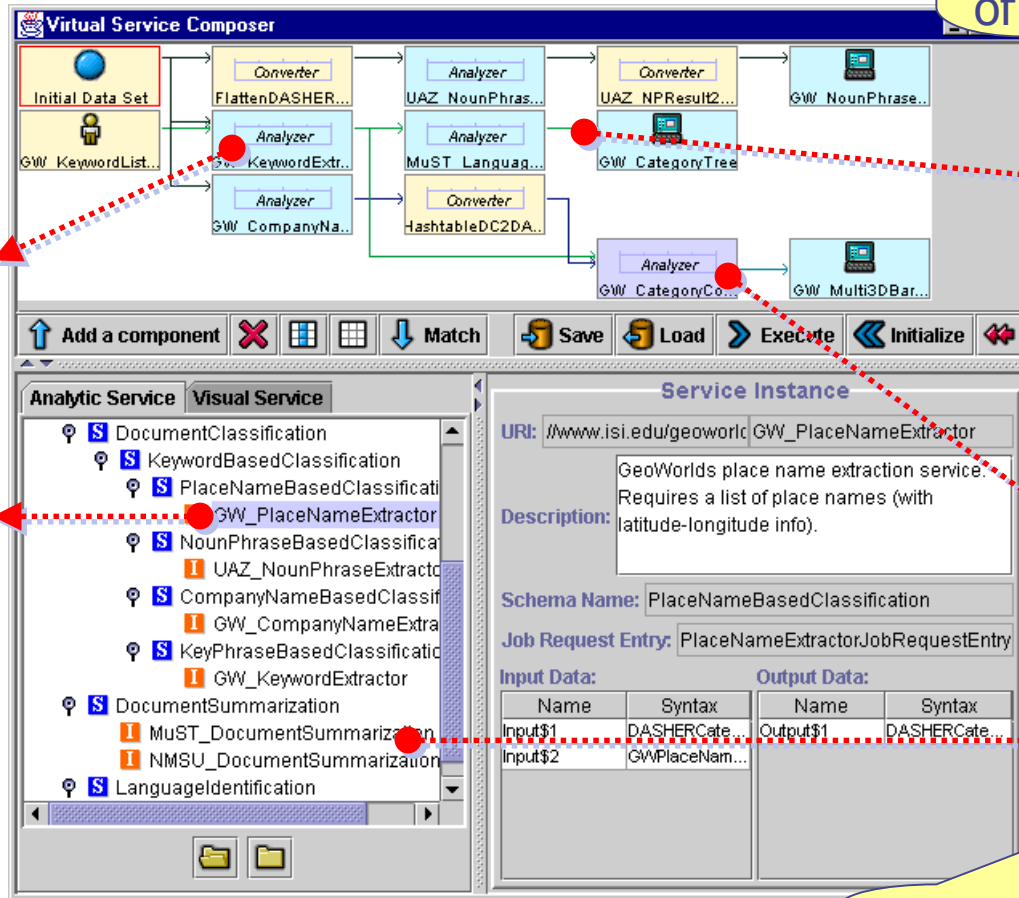
Edit



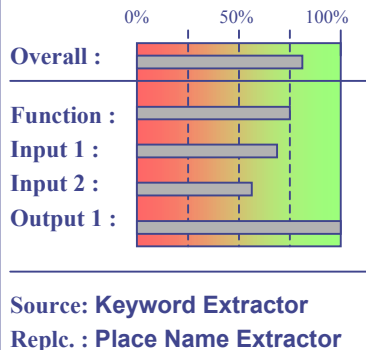
# Design-time Semantic Gauges

## Application Scripting

To check the interoperability level of a connection

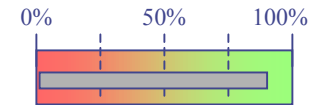


Compatibility Gauge



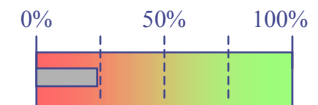
To select an alternative component

Interoperability Gauge



From :   
 Language Identifier   
 To :   
 Category Tree Viewer

Interoperability Gauge



From :   
 Category Comparison   
 To :   
 Document Summary

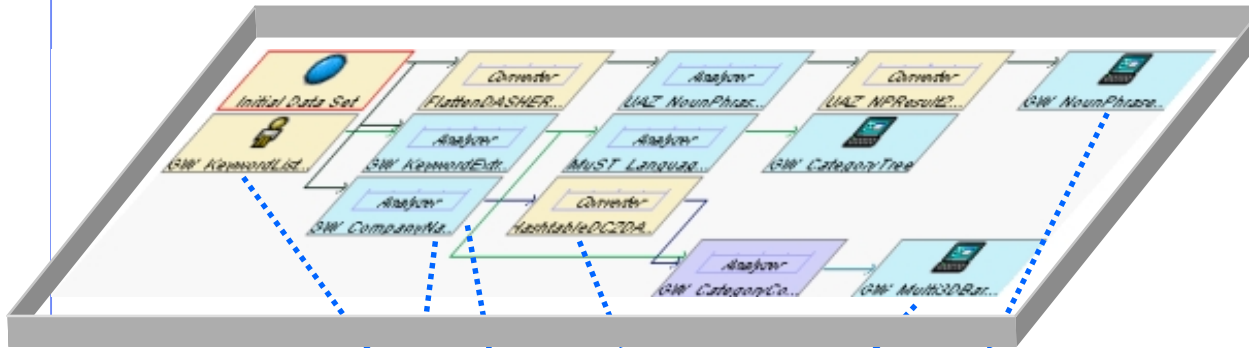
To select a component to be combined



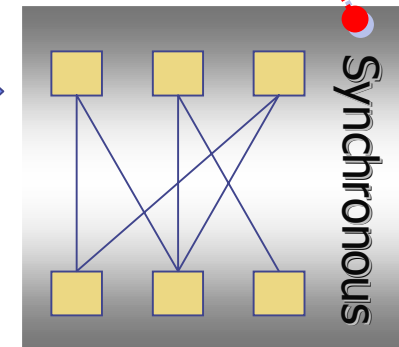
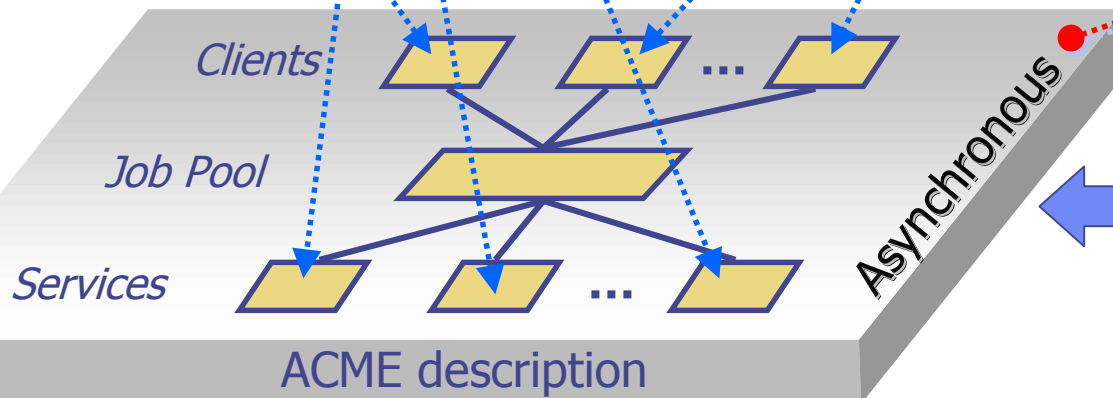
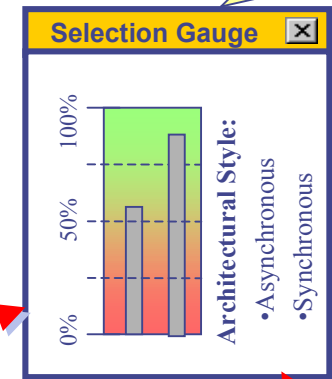
# Design-time Semantic Gauges

## Architecture Selection

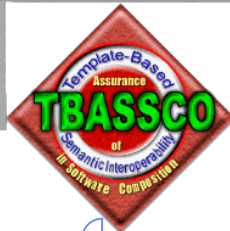
### Data Flow Description



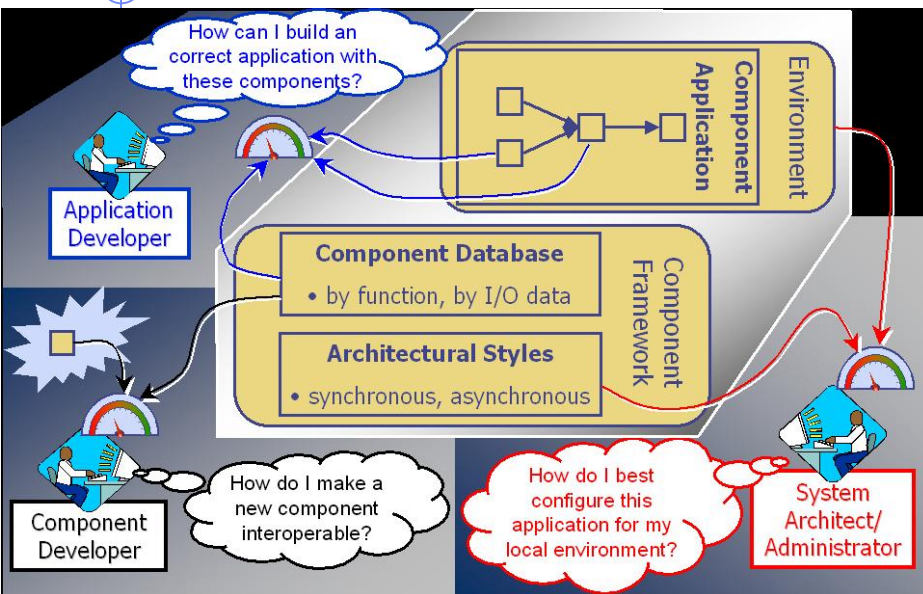
To select an architecture based on performance gauge feedback



### Architectural Description



# TBASSCO – Template-based Assurance of Semantic Interoperability in Software Composition



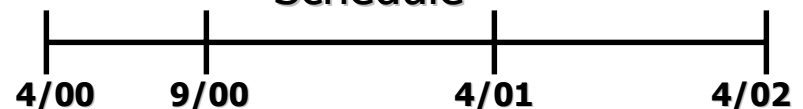
## Impact

- ◆ Lower cost by enabling the rapid assembly of semantically correct component systems
- ◆ Lower cost by facilitating integration of new components into component frameworks
- ◆ Increase performance by allowing system architects to tune system architecture

## New Ideas

- ◆ Lightweight semantically-based metadata to describe components
- ◆ Granularity and organization schemas to identify new component mismatch
- ◆ Meta-level data flow to facilitate switch among architectural styles
- ◆ Semantic gauges to guide design

## Schedule



- ◆ 9/00 – Joint collaboration and demonstration scenario for assured system assembly
- ◆ 4/01 – Component-level gauges to guide application and component development
  - ◆ Component compatibility, insertion gauges
- ◆ 4/02 – Architectural-level gauges to guide system level refinements
  - ◆ Architecture selection, deployment gauges



# Collaboration Plans

## ISI

- 2001
- 2002

Other Participants

## Design Time Aids

### ISI

- Gauges to select interoperable components
- Gauges to determine difficulty of adding new components
- **Gauges to adapt architecture to computing environment**

## Requirements / Capability Descriptions

### ISI

- Semantic function and data descriptions

### BBN

- Architecture requirement documents

## Existing Software

(From Library of Available GeoWorlds / SDC Components)

**ACME ADL  
XML/FleXML**

## Prospective Software

(Same Library:  
Alternative Extensions,  
Compositions)

## 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

### Object Services

- Application profiling gauges and topology gauges on configuration, component usage
  - Component binding
  - Dead libraries
  - Versioning
  - Activity