Infosphere: A Midterm Update on Infopipes

Calton Pu
Professor and John P. Imlay, Jr. Chair in Software
Georgia Institute of Technology
© 2003 Calton Pu and Georgia Institute of Technology

Ubiquitous Computing

♦ Plenty of computers (Moore’s Law)
  • Are everywhere, all connected, almost free
  • Know Everything
  • 5 IT Expeditions in 1999 (e.g., MIT’s Oxygen)
♦ Information is where value is
  • Paying distributed applications (DL, EC)
  • Cheap disks connected by cheap bandwidth
Infosphere

Problem: too many sources, too much information

Internet: Information Jungle

Clean, Reliable, Timely Information, Anywhere

Resource Adaptation, Infopipes, Property Mgmt, specialization

Personalized Filtering & Info. Delivery

Digital Earth

Sensors

Infopipes

 continuum of queries

Continual Queries

Microfeedback

Navigator

- Freeway Surveillance
  - 80+ PTZ Cameras
  - 500+ VDS Cams
  - Many info sources
  - Potential direct input
    - Instrumented cars
    - Trucks and others
RT Traffic Optimization

♦ Central traffic control scenario
  • Car navigational system registers route
  • Control sends back detailed traffic info/estimate
  • When rerouting: register the new route

♦ Peer-to-peer scenarios
  • Taxi/delivery trucks exchange information
  • Opposite direction cars know what’s ahead

Emergency Response

♦ Increasing importance after 9/11, Iraque
♦ Proactive planning, handling, response
  • Hurricanes, tornados, presidents, etc
  • Real-time observation and forecasting of exact route for “big events”
  • Get out of harm’s way, get back to help
  • A few minutes can make a big difference
Infopipes: Backbone of Infosphere

- Syntax, semantics, QoS
- Property preserving composition of Infopipes
- Timely delivery of high quality fresh information

Infopipe Abstraction

- Syntax, semantics, QoS requirements
- Component Infopipes
  - Ends: Typespec, property specifications
  - Middle: processing, buffering, active
- Composition of Infopipes
  - End-to-end QoS property preservation
Goals for ISL/ISG

♦ ISL – Infopipe Specification Language
  • Simple description for Infopipes
  • Support datatypes
  • Support serial composition
  • QoS requirements
♦ ISG – Infopipe Stub Generator
  • Generate datatypes, communication stubs
  • Support multiple communication layers
  • Support multiple implementation languages

Infopipe Internals

♦ Information flow driven
  • Middle method by hand
  • ISG generates the rest
Infopipe Software Architecture

- ISL Spec
- GUI
- XIP Spec
- Multi-stage translators
- ECho
- Java RMI
- Sockets
- CORBA A/V Stream
- Fill Repository
- XIP + build
- C Sockets
- Java Sockets
- C ECho
- XIP assembly
- Apply templates

Multi-Stage Translation

- XIP Spec
- Custom Translators
- XSLT Translators
- Execut. Source
- XSLT Translators
Demo Scenario

Big-picture image stream

UAV video

UAV video

UAV video

Team video

Mobile Command
Receiver controls signals through remote data reduction

Demo Application – Sensor Streams in Wired and Wireless Environments

Multiple Wireless Media

Command Post

Wired Media
Ptolemy Design Tool

UAV: datatype, filter, pipe
XIP => Gen code

Multiple Language/API Bindings
Generating Multiple Languages

Code Generation & Compilation
Encryption (Symmetric)

JPEG (factor of 1/25)
Infopipe and Software Eng.

♦ QoS/QoI support implemented with AOP
  • QoS dimensions: performance, security, etc
  • Performance monitoring, recording, adjusting
♦ Feedback-based QoS maintenance
  • Monitoring at each stage of information flow
  • Make adjustments when QoS exception raised

XIP Benefits

♦ Extensible XML intermediate representation
  • Several abstract machines, custom code
  • Datatypes, Filters, Pipes, Serial composition
  • Easy addition of AXpect
♦ Good XML software tools
  • XSLT Modularity (xsl:include)
  • Meta programming through templates
  • Leveraging future XML tool development
An Aspect

Pointcut: selects joinpoint in XIP from generated code
Joinpoint: Added in aspect, or from generated code

AXpect Weaver

- XIP specification has `apply-aspect` commands
- Aspects are XSLT, operate on XIP + generated code
- Aspects can refer to outside XML (e.g. WSLA) for additional data

Source code: C, C++, Java
An AOP Experiment

- Infopipe: Image stream sender to receiver
- Receiver has CPU usage limits
- Aspects build on top of other aspects (e.g. CPU usage computation requires timing info)
- Rate control aspect governs receiver CPU usage through sender rate adjustment
- Aspect code accounts for 348 of 1135 NCLOC (31%)
- Aspects affect 9 of 14 generated files

Specialization of Infopipes

- Specialization of Infopipes
  - Tools: Tempo-C and Java partial evaluator
  - Goal: reduce interpretation of data (2nd example later)
Infopipe Further Details

- **Middle**: Pipe’s processing module
- **Port**: Communication endpoint abstraction
- **Stub**: Data marshalling
- **Connector**: Communication runtime, Handles connection info

Pipe Connectors

- Each port can be plugged with different types of stubs and connectors
- Currently we have 3 types
  - Socket connector
  - IPC (unix pipe) connector
  - Function connector
Replugging (Same Machine)

Pipe 1, 2 are collapsed into one pipe
Pipe 1 loads Pipe 2’s middle code dynamically

Replugging (Same Process)
Port Code Example

**Same Interface:** pipeSource->outPorts["out1"]->push(item);

```c
Port::push(item)
{
    lock(stub); // for replugging synchronization
    stub->push(item);
    unlock(stub);
}
```

```c
StubSocket::push(item)
{
    marshal(item, buf);
    conn->send(buf);
}
```

```c
ConnSocket::send(buf)
{
    send(sock, buf);
}
```

```c
StubIPC::push(item)
{
    marshal(item, buf);
    conn->send(buf);
}
```

```c
ConnIPC::send(arg)
{
    write(pipe_fd, arg);
}
```

```c
StubFunction::push(item)
{
    nextPipe->middle
    ->process(item);
}
```

Benchmark Results

- Send 8 byte data (in the same machine)
- (Manually specialized) Specialization cost

<table>
<thead>
<tr>
<th></th>
<th>Socket</th>
<th>IPC</th>
<th>Function</th>
<th>Socket-&gt;IPC</th>
<th>IPC-&gt;Function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>41.7 us</td>
<td>12.6 us</td>
<td>1.8 us</td>
<td>1127.6 us</td>
<td>1045.7 us</td>
</tr>
</tbody>
</table>
Project Summary

- Infopipe: Distributed information flows
  - ISL, GUI, XIP
- ISG and software engineering tools
  - AOP and AXpect
  - Specialization of Infopipes
- End-to-end QoS properties
  - Performance, maintainability, scalability

Fresh Information On the World