Distributed Architecture Definition Language
DADL
Ron Burbach
October, 1997
# Areas of Focus

<table>
<thead>
<tr>
<th>Control</th>
<th>plan repair, re-planning, process changes, plan optimization, chronic problem management, ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure</td>
<td>number of faults both reported and fixed, lines of code, closeness to plan, resource utilization, performance, ...</td>
</tr>
<tr>
<td>Strategies</td>
<td>methodologies, <strong>architecture</strong>, paradigms, mission, risk analysis, scheduling, priority setting, resource utilization, decision making, life cycle management, ...</td>
</tr>
<tr>
<td>Tools</td>
<td>compilers, debuggers, environments, quality assurance, CASE, version control, databases, operating systems, networks, file systems, GUI builders, composition, ...</td>
</tr>
<tr>
<td>People</td>
<td>group interactions, skill development, group dynamics, communications, goal setting, ...</td>
</tr>
</tbody>
</table>
Basic Client/Server

Network

Client

Server
Three Tiered Client/Server

Network

Presentation  Application Logic  Data
SMTP based Electronic Email

email from gio@db to ron@cs
HTTP based Web Example of Client/Server

get http://cs/~ron

web server deamon

~ron/index.html

web client

HTML SSL

cs

db
Defining Characteristics

- Distribution
- Concurrent
- No Global State
- No Global Clock
- Partial Failures
- Asynchronous Communication
- Distributed Control

- Heterogeneous Systems
- Local Autonomy
- Evolution Programming Paradigm
- Constant Change
- Many Transparencies
- Openness
- Interdependence
- Security
What is the Problem?

- Architectures are defined with a few drawings, hand waving arguments, and English based document
- Not precisely defined
- Traditional programming languages concentrate on algorithms and data structures
The Solution?

- We need a language that can describe distributed architectures.
- Extension to existing programming languages.
- DALD (Distributed Architecture Definition Language)
Architecture Definition

- The components and their interfaces, communication, and contractional behavior.
- Traditional programming languages concentrate on algorithms and data structures which define the components but do very little at defining interfaces, communication, and contractional behavior.
Consider this simple program which adds two integers.

- void main () { int results = plus(2,1) ; }
- int plus ( int n, int m) { return n+m ; }

What is the architecture?

There is an implicit architecture, so implicit that it is seldom mentioned. The two components communicate using a shared address space and a call stack frame. The communication is assumed error free and both components are flawless.
Comparison of Architectures

- **Traditional**
  - Not Distribution
  - Not Concurrent
  - Global State
  - Global Clock
  - No Failures
  - Synchronous Communication
  - Centralized Control

- **Distributed**
  - Distribution
  - Concurrent
  - No Global State
  - No Global Clock
  - Partial Failures
  - Asynchronous Communication
  - Distributed Control
Comparison of Architectures

- Traditional
  - Homogeneous Systems
  - REvolution Programming Paradigm
  - Fragile
  - Only one Transparencies
  - Closed
  - No Security
  - Local Autonomy
  - Interdependence

- Distributed
  - Heterogeneous Systems
  - Evolution Programming Paradigm
  - Constant Change
  - Many Transparencies
  - Openness
  - Security
  - Local Autonomy
  - Interdependence
Some DADL Concepts

- *dagents* are the components
- a *contract* determines the *resources* and *performance* demands of a dagent
- *terms* and *sentences* build a *conversation* over *connections* which determines the *behavior*
Example: Plus

alphabet (byte);
term t1 (int n);
term t2 (int m);
sentence s1 (t1, t2) from c1 to c2;
term t3 (int plus (int, int) highly available;
sentence s2 (t3) from c2 to c1;
behavior (s1; s2);
volatile;
open data;
unmarshaled;
unauthenticated;
unauthorized;
A DADL Environment

DADL Runtime

Environmental Services such as authentication, authorization, data privacy, data integrity, marshaling, persistence management, replication, transaction processing, distributed lock management, databases, and GUI systems.
DADL Development

DADL Program

DADL Compile
- Dagent Stub
- Conversation Service

Dagent Stub
+ programmer supplied code
+ library

Link
Dagent Application

Conversation Service
+ library

Link
Conversation Application