Fault-Tolerant CCM

Middleware for Embedded Adaptive Dependability (MEAD)

Real-Time Fault-Tolerant Middleware Support

Contract #: F33615-03-C-4110

Thomas D. Bracewell
Dave Sperry, Maureen Mayer
Raytheon
Priya Narasimhan
Carnegie Mellon University

CCM Workshop, Nashville, TN
December 10, 2003

Areas of Investigation

- **Current Areas of Investigation**
  - **Defining the Fault Tolerant CORBA Component Model**
    - □ precursor to a real-time fault tolerant CCM
  - **Real-Time Fault Tolerant CORBA Standard RFP**
    - □ launched at London OMG meeting
  - **Fault Injector for Middleware Applications (FIMA)**
    - □ a CORBA-based software fault injector

- **Collaborators**
  - Dr. Priya Narasimhan, Carnegie Mellon University (MEAD)
  - Nanbor Wang, Vanderbilt (Real-Time CCM)
  - Dr. Douglas Blough, Georgia Tech (FIMA)
  - Boeing OEP

Defining a Fault Tolerant CORBA Component Model

Tom Bracewell, Maureen Mayer, David Sperry

Background

In the beginning...

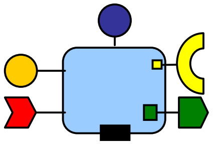
OOP let us encapsulate related data and operations

CORBA linked distributed objects, hid platform dependencies

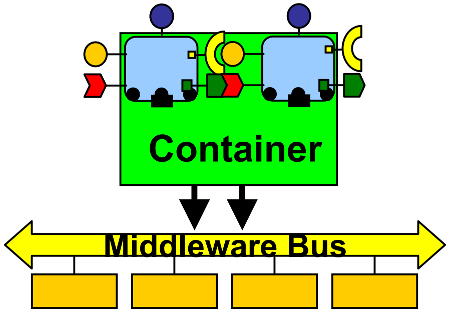
FT CORBA added fault tolerant (FT) support for distributed objects

Then came...

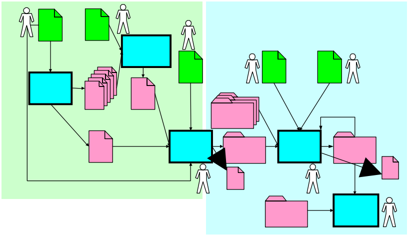
Components - assemble/deploy objects into larger applications



CORBA Component Model (CCM)
distributed component model



Model Integrated Computing (MIC) - tools & process to implement/package/assemble/deploy distributed components



PCES Challenge:

How can we add fault tolerant support for distributed components ?

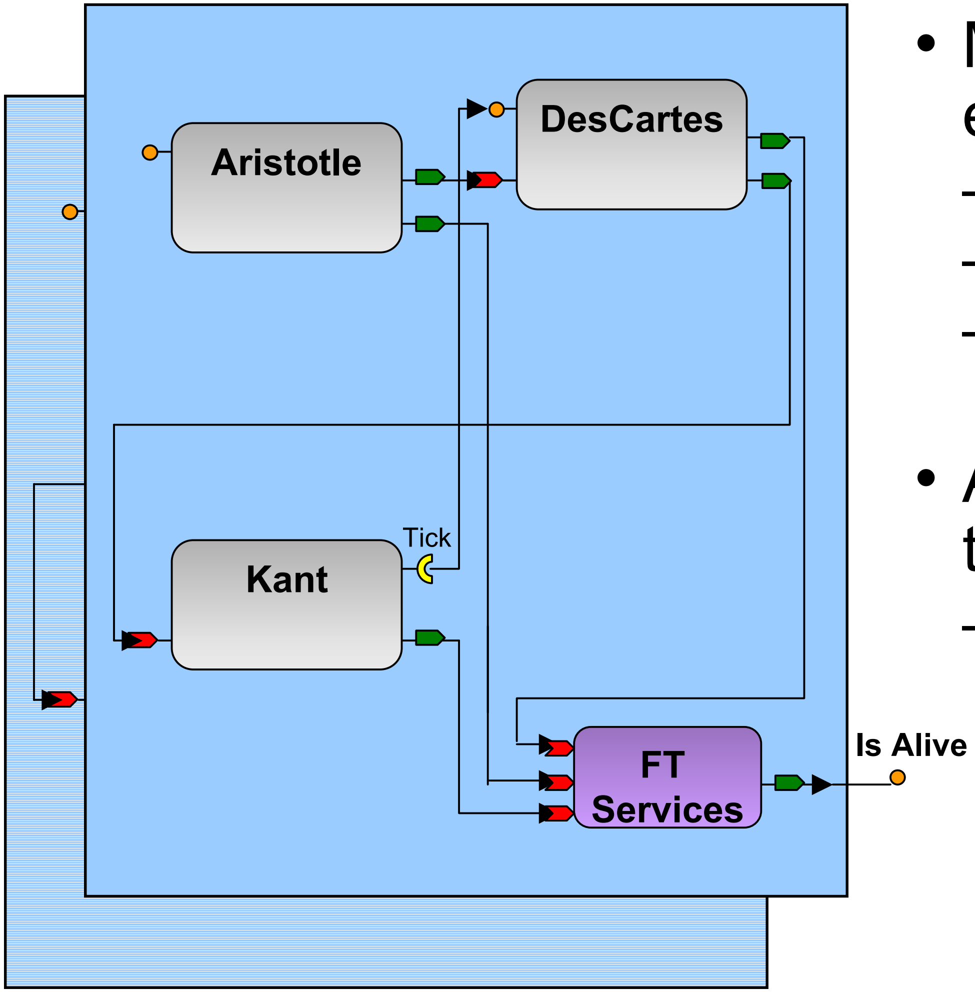
Why Have a FT CCM ?

- **CCM and MIC can support fault tolerant (FT) systems**
- **Weave fault tolerance into component-based designs**
- **Leverage benefits of component model in FT designs**
  - **separation of concerns at the right levels**
    - e.g. component, container, server level
    - run-time configurations, connections
  - **composition-based FT assembly and deployment**
    - build fault tolerant configurations and connections
    - separate logical from physical deployment
    - automate fault tolerant assembly and deployment
  - **metadata captures FT properties/policies**
- **First step towards a Real-Time Fault Tolerant CCM**
    - a CIAO that goes with MEAD !

FT CCM Goals

- Application-transparent fault tolerance
- Fault tolerant design by composition
- Apply MDA principles
  - One fact in one place
  - Separate design from platform
- Automate fault tolerant component assembly & deployment
  - Hide the details of FT assembly and deployment
- Minimize edits to FT components
- Minimize impact to existing standards

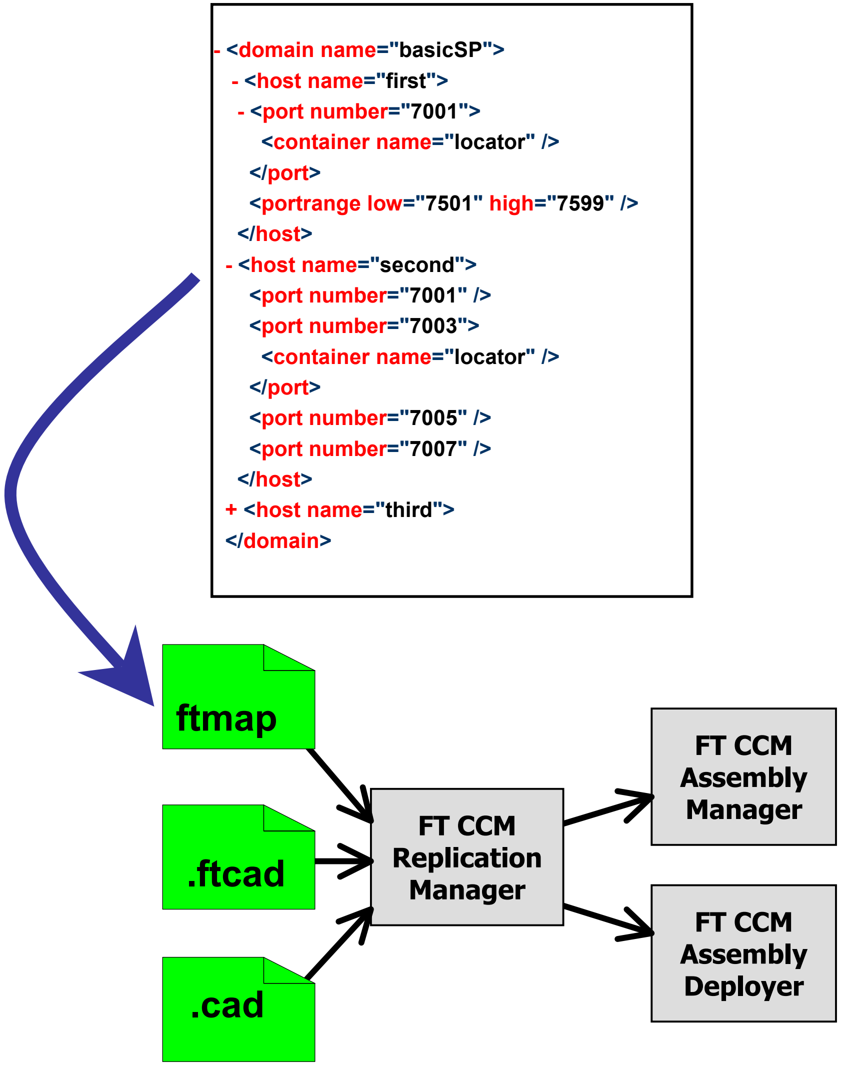
Approach to FT CCM



- Make container (not object) entity of redundancy
  - proper separation of concerns
  - handle container state
  - affects FT CORBA
- Add FT services component to containers
  - link components to FT services at the container level
    - □ fault monitoring
    - □ checkpoint (log) components
    - □ log container state
    - □ implement in CIAO daemon

Do I exist? Only the ORB needs to know...

Approach to FT CCM, ctd.

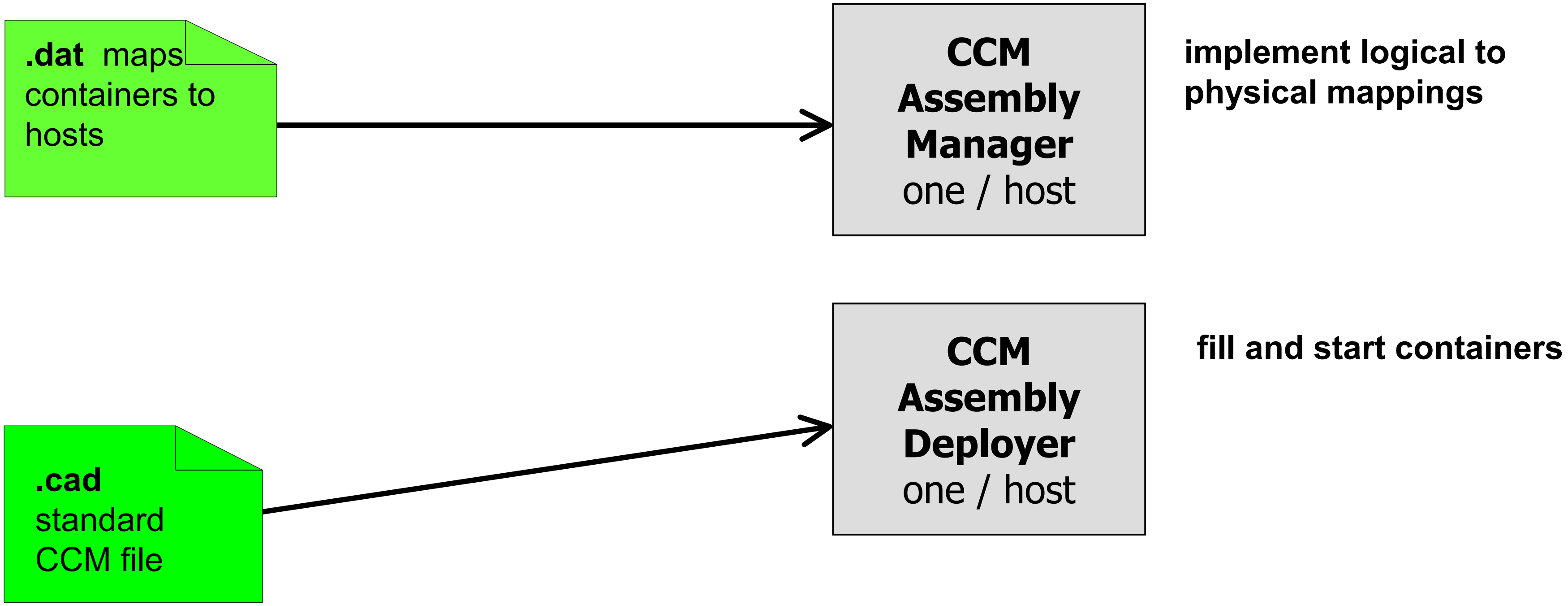


```
- <domain name="basicSP">
  - <host name="first">
   - <port number="7001">
       <container name="locator" />
     </port>
     <portrange low="7501" high="7599" />
   </host>
 - <host name="second">
     <port number="7001" />
     <port number="7003">
       <container name="locator" />
     </port>
     <port number="7005" />
     <port number="7007" />
   </host>
 + <host name="third">
 </domain>
```

- Use metadata to express FT properties/policies
  - e.g. replication
- Divide logical and physical assembly in CCM process
  - Currently a one-step process
  - Number of replicas is logical
  - Replica placement is physical
  - A Replication Manager deploys replicas during runtime, using
    - □ deployment rules (.ftmap)
    - □ resource declarations (.ftcad)
    - □ container definitions (.cad)

Current assembly/deployment process

Combines physical and logical assembly



.dat maps containers to hosts

CCM Assembly Manager one / host

implement logical to physical mappings

.cad standard CCM file

CCM Assembly Deployer one / host

fill and start containers

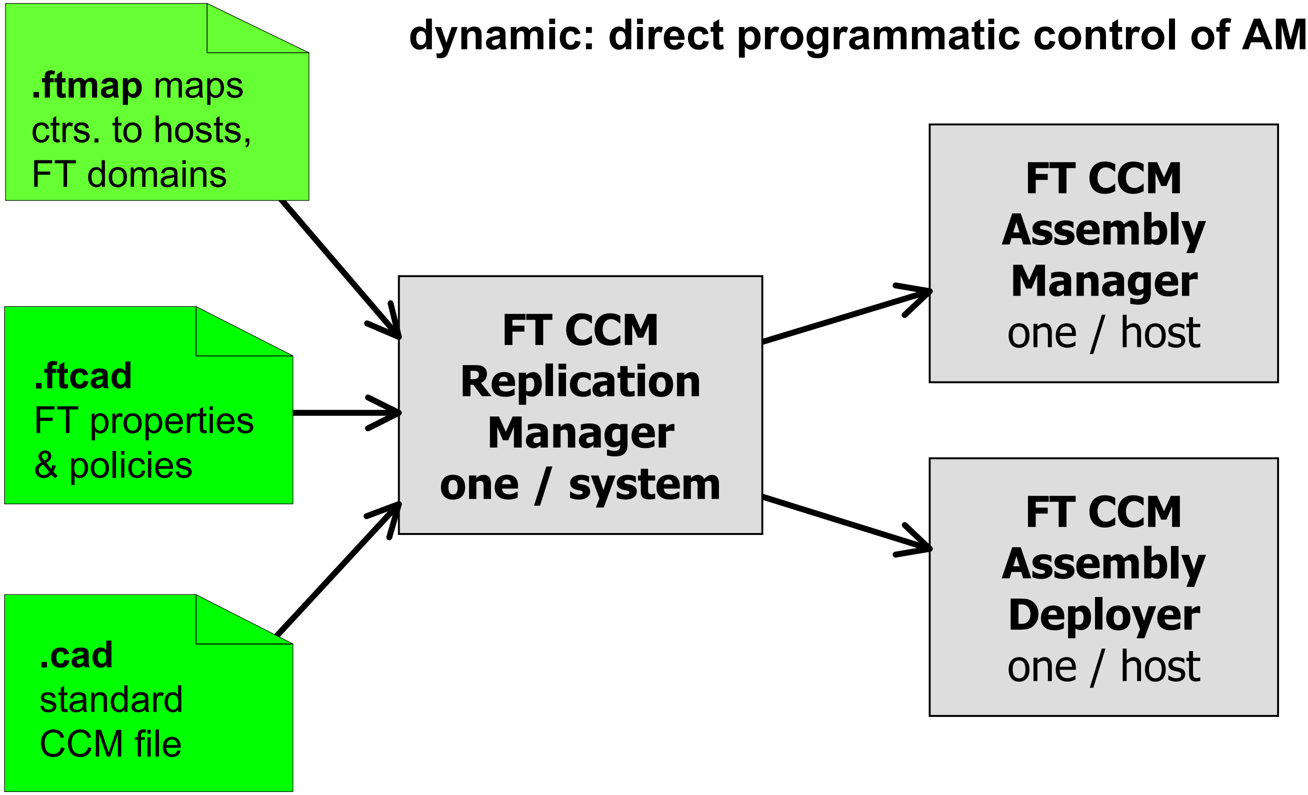
FT CCM assembly/deployment process

Decouples logical and physical assembly

Replication Manager output can be static or dynamic

static: a post-processed .cad file with all replicas assigned

dynamic: direct programmatic control of AM, AD at runtime



.ftmap maps ctrs. to hosts, FT domains

.ftcad FT properties & policies

.cad standard CCM file

FT CCM Replication Manager one / system

FT CCM Assembly Manager one / host

FT CCM Assembly Deployer one / host

implement logical to physical mapping

may also dynamically reconfigure mappings

fill and start containers

start alternate containers after failover and reclaim resources

XML Schemas: FT MAP file

FT MAP file maps containers to FT Domains, host ports

```
- <domain name="basicSP">
  - <host name="first">
   - <port number="7001">
       <container name="locator" />
     </port>
     <portrange low="7501" high="7599" />
   </host>
 - <host name="second">
     <port number="7001" />
     <port number="7003">
       <container name="locator" />
     </port>
     <port number="7005" />
     <port number="7007" />
   </host>
 + <host name="third">
 </domain>
```

- Logical to physical mapping
- Replication Manager (RM) assigns container replicas to daemons running on host ports
- If a container name is mapped to a port, the RM may only allocate a replica of that container to that port
- If only a port is specified, RM is free to use that port for any container replica
- One FT MAP file per system replaces one **.dat** file per host

XML Schemas: FT CAD file

FT CAD file defines FT container & component properties

- Applies FT marks atop Component Assembly Descriptor (.cad) files

```
- <ftproperties>
   - <!--  Properties defined at the container level are defaults for the components in the container and the general

     container behavior -->

•    <container name="locator" replication-style="active" initial-no-of-replicas="3" min-no-of-replicas="2"
     membership-style="memb_inf_ctrl" consistency-style="cons_inf_ctrl" fault-monitoring-style="push" fault-
     monitoring-interval="100" fault-monitoring-timeout="500" fault-monitoring-granularity="container" heartbeat-
     policy="48" heartbeat-enabled-policy="49" request-duration-policy="47" checkpoint-interval="200" />

   <container name="viewer" replication-style="stateless" initial-no-of-replicas="1" min-no-of-replicas="1"
     membership-style="memb_inf_ctrl" consistency-style="cons_inf_ctrl" fault-monitoring-style="pull" fault-
     monitoring-interval="10000" fault-monitoring-timeout="20000" fault-monitoring-granularity="container"
     heartbeat-policy=”48" heartbeat-enabled-policy=”49" />
- <!--  Properties defined at the component level override defaults set at the container level  -->
   <component name="display" />
   <component name="rategen" />
   <component name="gps" />
  </ftproperties>
```

Applying FT properties and policies

		may be applied to	
	FT properties and policies	Components	Container Groups
	replication style		X
	initial no of replicas		X
	min no of replicas		X
	membership style		X
	consistency style		X
	fault monitoring style	X	X
	fault monitoring interval	X	X
	fault monitoring timeout	X	X
	fault monitoring granularity	X	X
	heartbeat policy	X	X
	heartbeat enabled policy	X	X
	checkpoint interval (logging)	X	X
	factories	homes	
	FT domain ID		X
	~~obj~~ container group ID	n/a	X
	~~obj~~ container group ref version	n/a	X
	request duration policy	X	
metapolicy	mode_ID (mode driven FT)	X	X
metapolicy	knob settings (various)	X	X

Requirements

- **FT CCM Services Component**
  - provides FT services to its container
  - provides FT services to application components in its own container
    - fault detection, logging
  - requires an FT ORB that supports container-level redundancy
- **FT CCM Replication Manager**
  - redeploys and reconnects container applications
  - no single point of failure
    - must be able to replicate / reconfigure itself
  - handles container-level property management, groups, factories

Redeploying container applications

- **Replication Manager must support dynamic reconnections after fault detection**
  - try to restore the failed replica; or
  - reclaim failed replica’s resources and create a new physical replica somewhere else;
  - update the IOGR version with the new replica.
  - CCM-level FT CORBA would use FT CCM Assembly Deployer and FT CCM Assembly Manager

- **“Aspects” make us refactor what’s CORBA, what’s CCM**
  - e.g. a new CORBA standard would use a new CCM standard

Plain FT CORBA under a CCM won’t do

- **FT CORBA must treat containers as entity of redundancy**

- **Container & CCM issues**
  - Container-level IOGR is needed
    - □ supports transparent client redirection at the container level
  - Container state must be logged
    - □ even stateless components have stateful containers
  - Containers are OS-version-specific and language-specific design
    - □ hard to move containers to arbitrary hosts
    - □ which part of container must be replicated
    - □ what to checkpoint and restore
  - Container thread scheduling
  - Container quiescence
  - Lifecycle issues - e.g. FT cookies

In Summary

- **FT CCM provides FT support to distributed components**

- **FT CCM doesn’t come for free**
  - FT CORBA must support containers as entity of redundancy
  - “aspects” lead us to refactor what’s CORBA, what’s CCM

- **FT CCM offers real payback**
  - FT systems will be able to use components, CCM and MIC tools
  - FT assembly and deployment will be easier
  - FT properties/policies will be managed at appropriate levels

Customer Success Is Our Mission

FT CCM Demonstration

David Sperry, Amanda Kelly, Al Butturini

Demonstration of FT CCM Concepts

- Demonstrate fail-over to active replica at container level
- Use the CIAO example$CIAO_ROOT/examples/OEP/Display
- Try not to modify existing components
  - Changed some files to work around value type factory problem in CIAO
- Discover FT issues unique to CCM

Demonstration Strategy

- (use CIAO example)

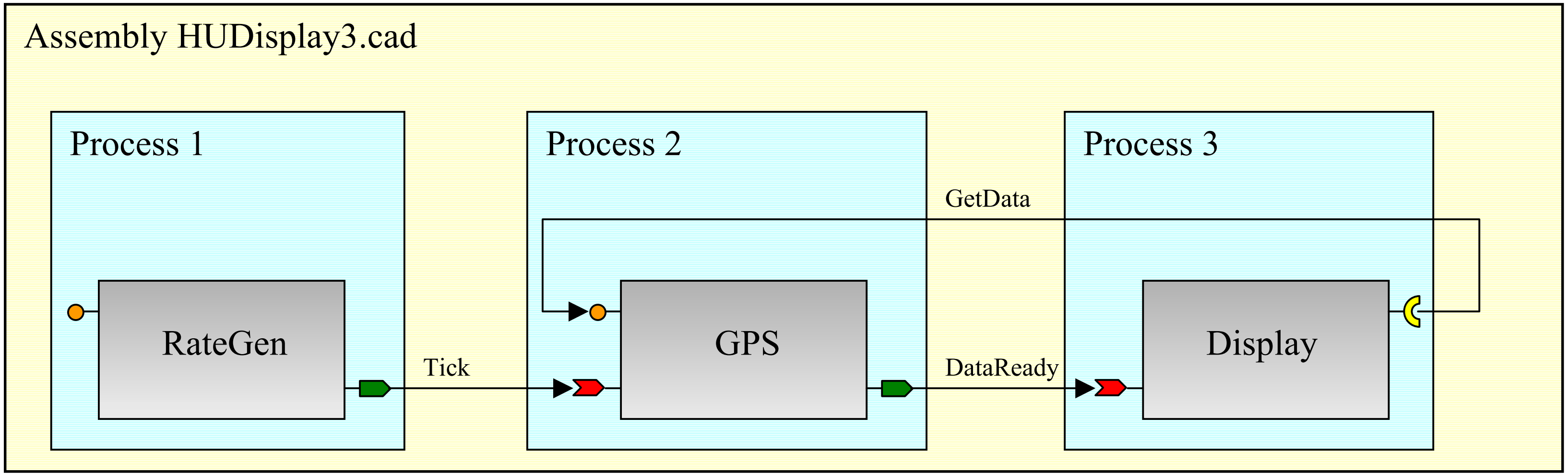
- **Active replication** of GPS component
  - add 1 new component to assembly
  - deploy 3 active replicas of GPS

- Fault is detected by missed update from active GPS
  - Expected update rate = 1 Hz
  - Fail over at no update for 3 seconds

- Recovery by switching to healthy **component-level replica**
  - Fault detection and replica selection performed by new component.
  - All connections are static and setup at assembly time.

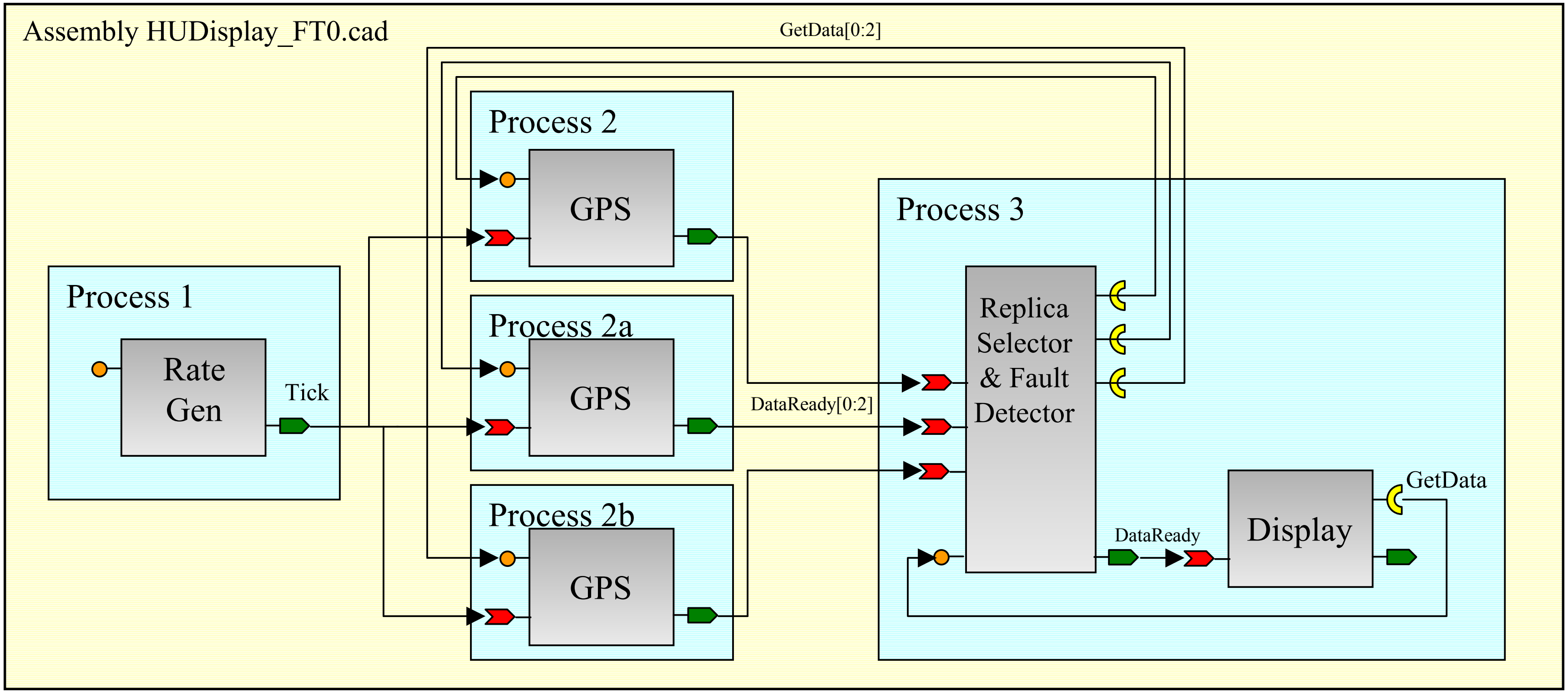
Demonstration of FT CCM Concepts

- CCM assembly layout for Original CIAO Display Example



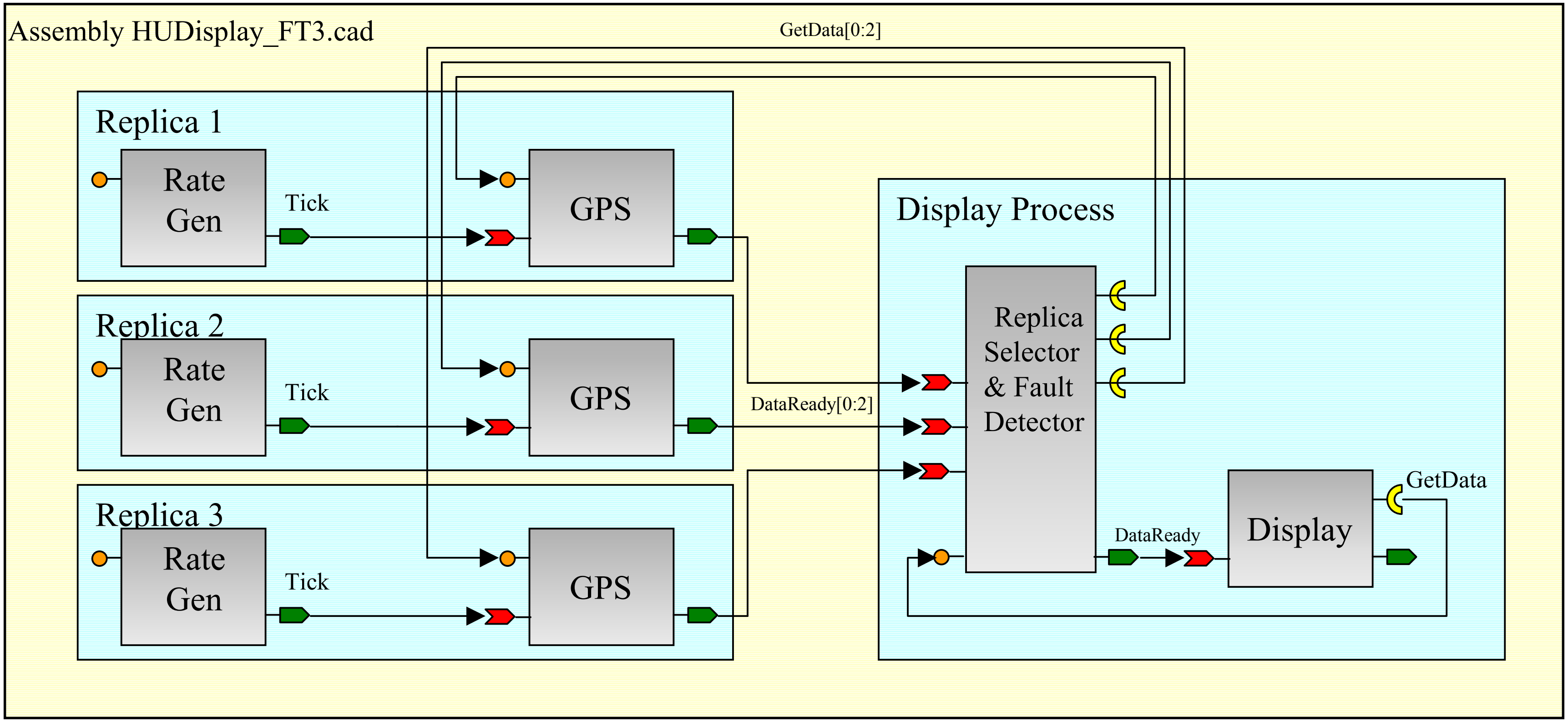
Demonstration of FT CCM Concepts

- CCM assembly layout for FT Display - as conceived...
  - we'd have built this, but for an undesirable CCM event service condition
    - if any GPS replica fails, the Rate Gen ORB hangs; fix the CCM standard!



Demonstration of FT CCM Concepts

- CCM assembly layout for FT Display - as deployed



DARPA PCES

Raytheon
Customer Success Is Our Mission

OMG RFP for Real Time Fault Tolerant CORBA

Tom Bracewell

Real-Time Fault Tolerant CORBA RFP

Mandatory Requirements

- Application-transparency
- Object, crash, msg, timing faults
- Container failover
- Component FT properties
- Bounded recovery times
- Fast fault detection / fault isolation
- End-to-end predictability
- Scalability
- Policy-driven dependability
- Configurable real-time FT
- Order tasks, events, operations
- Multithreading
- Nested operations
- Proactive dependability

Mandatory Requirements, ctd.

- Platform heterogeneity
- Incremental checkpointing
- ORB interoperability
- Clock synchronization
- New replication styles
- Self-healing ORB

Optional Requirements

- Extendable fault model
- Software rejuvenation
- Real-Time Java
- Security and survivability hooks
- Live software upgrades
- Partition tolerance

RFP Timetable

TC votes to issue RFP	*April 2004*
LOI to submit to RFP due	*June 29, 2004*
Initial Submissions due	*August 20, 2004*
Voter registration closes	*October 25, 2004*
Initial Submission presentations	*November 1, 2004*
Revised Submissions on OMG server	*February 9, 2005*
Revised Submission presentations	*March 2, 2005*
TC votes to recommend specification	*May 2005*
BoD votes to adopt specification	*June 2005*

MEAD - CCM Roadmap

Tom Bracewell, Dr Priya Narasimhan

MEAD - CCM Roadmap

- Identify CCM issues for fault tolerance (today)
- Demonstrate fault tolerant CCM (tomorrow)

- Upcoming capabilities
  – Proactive Dependability for CCM applications
  – Fault Tolerant Configuration Advice for CCM applications
  – Real-time Fault Tolerant CCM

- Standardization efforts
  – Based on insights from MEAD-CCM and MEAD-CORBA, Raytheon-CMU team intends to:
    - participate in QoS4CCM standardization
    - respond to OMG RFP for Real-Time Fault Tolerance

Showcasing MEAD

- MEAD-CCM demonstration
  - Tom Bracewell, Dave Sperry, Maureen Mayer (Raytheon)
  - Active replication; fault detection and failover

- MEAD-CORBA demonstration
  - Priya Narasimhan, Joe Slember (CMU)
  - Active and warm passive replication
  - Fault-detection, failover and recovery
  - Proactive fault-tolerance
  - Resource monitoring and fault-tolerance advice

- Come see us tomorrow!