Remedy IT

Your challenge - our solution

Evolution from LwCCM to UCM

Remedy IT

Your challenge - our solution

Draft RFP

- Draft RFP posted to the OMG before the 4 week deadline
- Published as mars/2013-02-05
- Based on initial feedback a revised draft RFP has been published as mars/2013-02-19
- Draft RFP presentation to MARS this morning (see mars/2013-03-01)

Remedy IT

Your challenge - our solution

Main requirements

- Proposals shall specify a component model for Distributed, Real-Time and Embedded Systems. This component model shall be an evolution of the Lightweight CCM specification
- Proposals shall be independent of any particular middleware standard

Remedy IT

Your challenge - our solution

Component Model

- Proposals shall provide a user extensible, event driven programming model, that supports both thread pools as well as a single-threaded/non re-entrant threading model as the default
- Proposals shall follow a component, container, and connector based architecture
- Proposals shall not have CORBA as a mandatory dependency
- Proposals shall provide an IDL defined PSM
- Proposals shall not provide language bindings to specific programming languages such as C, C++, C++11, or Java, but shall instead leverage the OMG IDL language mappings

Remedy IT

Your challenge - our solution

Container

- Proposals shall provide support for a simple, lightweight CIF container that encapsulates a flexible event queue/dispatch and threading model. The container provides the execution environment for UCM application components
- Proposals shall provide support for scheduling timers (sporadic and recurring) with application supplied event handlers, using at least monotonic and system time
- Proposals shall support the integration of technical services through the concept of service connectors which are connectors that have only one instance within a container

Remedy IT

Your challenge - our solution

Generic Interaction Support

- Proposals shall use the CCM Generic Interaction Support (GIS) connector concepts, and GIS extensions to the IDL grammar for the specification of components, ports, mirror ports, template modules, connectors, interfaces, and other supporting types in IDL
- Proposals shall offer support for GIS extended port types that implement a middleware agnostic publish-subscribe information exchange pattern
- Proposals shall support the definition of request-reply pattern extended port types via user-defined IDL “interface” type definitions using the mechanism of GIS template module instantiation

Remedy IT

Your challenge - our solution

Deployment

- Proposals shall address the interface of a UCM component implementation framework (CIF) to a deployment framework that is compliant with the OMG Deployment and Configuration of Component-based Distributed Applications (DEPL) specification
- Proposals shall address standardized component and connector life cycle state model with appropriate callback operations compatible with a DEPL deployment framework
- Proposals shall address reconfiguration and redeployment options for modification of a UCM domain at run-time

RFP Schedule

Remedy IT

Your challenge - our solution

- There are a lot of aspects to UCM
- At the OMG Reston and Berlin technical meetings a Component Information Day (CID) is scheduled to get more feedback from the user community
- We need to give interested parties enough time to work on UCM, we don't want a paper specification

Remedy IT

Your challenge - our solution

UCM Impact

- Within a component based project there are several roles
  - Software Architect
  - Component Designer
  - Component Developer
  - Deployment Planner
  - Software Integrator
- Not all roles are impacted in a similar way by an evolution from LwCCM to UCM

Remedy IT

Your challenge - our solution

Component Developer

- The component developer is impacted by UCM
  - New C++11 language mapping
  - Standardized timer connector instead of vendor specific API
  - Simplified Component Implementation Framework
- A LwCCM C++11 implementation can address the top points easily already
- Such an implementation provides already a lot of simplifications

Remedy IT

Your challenge - our solution

Architect/Designer/Planner

- All use a modeling environment
- The combination of their work results in a deployment plan which is generated by tooling
- UCM will provide them additional architecture and design decisions
- But will not structurally change their job
- The generated deployment plan needs to be extended for UCM
- UCM is mostly a tooling challenge

Remedy IT

Your challenge - our solution

LwCCM to UCM (for C++)

- Start with migrating developers to a LwCCM C++11 implementation
- Work on UCM standard
- Implement UCM with LwCCM C++11 as starting point
- Extend modeling tools and deployment generator
- Minimal changes to component business code for UCM
- Regenerate deployment plans and recompile the code
- Deploy an UCM system

Remedy IT

Your challenge - our solution

LwCCM to UCM (for not C++)

- Migration of application code is heavily dependent on the quality of the IDL language mapping
- Modeling and deployment generator changes are the same as for C++
- Could require improved language mappings