Remedy IT

Your challenge - our solution

Model Driven, Component Based Development for CBDDS

IDL to C++11

Johnny Willemsen

jwillemsen@remedy.nl

This presentation is based on material from

Northrop Grumman

Component Based DDS (CBDDS)

Remedy IT

Your challenge - our solution

- CBDDS is an integrated suite using seven OMG open standards
  - LwCCM, DDS, DDS4CCM, AMI4CCM, CORBA, IDL, and D&C
- Supports architecture development at a higher level of abstraction
- Encapsulation of event queue/dispatch, threading model, boilerplate code, application lifecycle management, extensions and connection management in a “container”
- CCM Generic Interaction Support (GIS) encapsulates DDS or any other middleware functionality inside a “connector” with APIs defined by local IDL interfaces

Remedy IT

Your challenge - our solution

Advantages CBDDS

- DDS4CCM APIs for DDS access are middleware agnostic and vendor independent
- CBDDS extends DDS to fill in the holes needed to define a complex, full featured DRE architecture with open standard vs. custom solutions

Remedy IT

Your challenge - our solution

Why CBDDS

- 5 Guiding Architectural Tenets:
  - **OA** Open Architecture (MOSA)
  - **MDA** Model Driven Architecture
  - **CBA** Component Based Architecture
  - **SOA** Service Oriented Architecture
  - **EDA** Event Driven Architecture (DOA)

- NGC adopted CBDDS to meet a larger set of goals and requirements
  - CBDDS addresses all five architectural tenets
- DDS by itself only *fully* addresses two of NGC five guiding tenets (OA & EDA)
  - Future OMG RPC4DDS spec anticipated to add SOA support
  - New MDA tooling is much more useful for CBDDS, but *can* help DDS-only users as well
- High performance not compromised to improve modularity, reuse and portability, as well as functionality, scalability and time/cost of development
- CBDDS adds structure, which is the very definition of architecture

Remedy IT

Your challenge - our solution

MDA Tooling

- Multiple MDA Tools have been funded
- Component Based Architecture (CBA) captured as a PIM
- Maps to a CBDDS IDL and D&C PSM
- Key auto-generated OA artifacts drive the overall process (IDL 3.5, D&C 4.0)

CBD Software Lifecycle

Remedy IT

Your challenge - our solution

- Zeligsoft CX
- Artisan Studio

- Eclipse

- Zeligsoft CX
- Artisan Studio

Architecture & System Design

Design Tool

IDE Tool

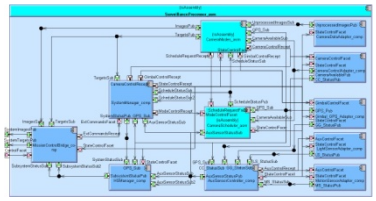
Deployment Planning Tool

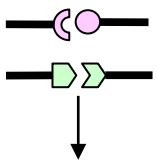
System Integration, Test & Verification

Agile process iterations early & often, incrementally building up from an early executable "skeleton" architecture

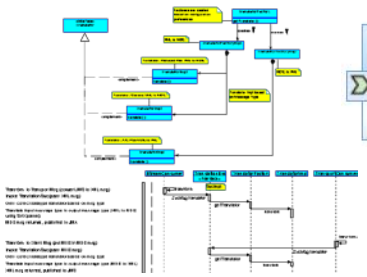
1. System Software Design & Component Definition
2. Component Interface Design
3. Component Software Design
4. Component Implementation
5. Component Packaging & Assembly
6. Component Deployment, Integration & Reuse

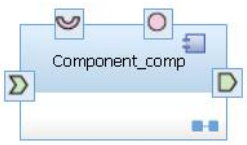
SNA CBD Software Lifecycle Process





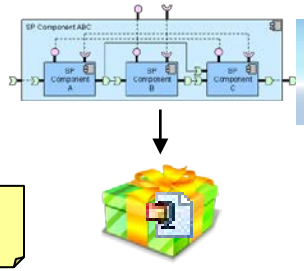
IDL





CPP, H

SO



CDP

CDD

Key Artifacts

- **IDE**: Integrated Development Environment
- **CBD**: Component Based Development
- **SNA**: Scalable Node Architecture
- **IDL**: Interface Definition Language (OMG)
- **CDP**: Component Deployment Plan
- **CDD**: Component Domain Descriptor

Remedy IT

Your challenge - our solution

OMG IDL Elements

- IDL offers vendor, programming language, and middleware independent format
  - OMG standards mapping IDL to C++, C++11, Java, Python, C, Ruby, etc.
- A given middleware standard implementation provides an IDL to language compiler
  - Model generated IDL -> IDL compiler generated source = large percentage of design code base
  - NGC's SNA SDK currently uses tao_idl and rtiddsgen IDL compilers (others in future)

Remedy IT

Your challenge - our solution

MDA tool modular IDL 3.5 convention

- Generate 5 fundamental file types for components, connectors, messages, interfaces and basic type definitions
- Taxonomy of 5 modular file types support component/port reuse and modularity (vs. all IDL in one project IDL file)
- IDL import & export feature of all CBDDS MDA tools enables basic model interchange using IDL
- Modular structure leveraged to auto-generate makefiles for entire component-based projects
- Run CCM IDL compiler on all types, only run DDS IDL compiler on *_defn.idl & *_msg.idl files

Remedy IT

Your challenge - our solution

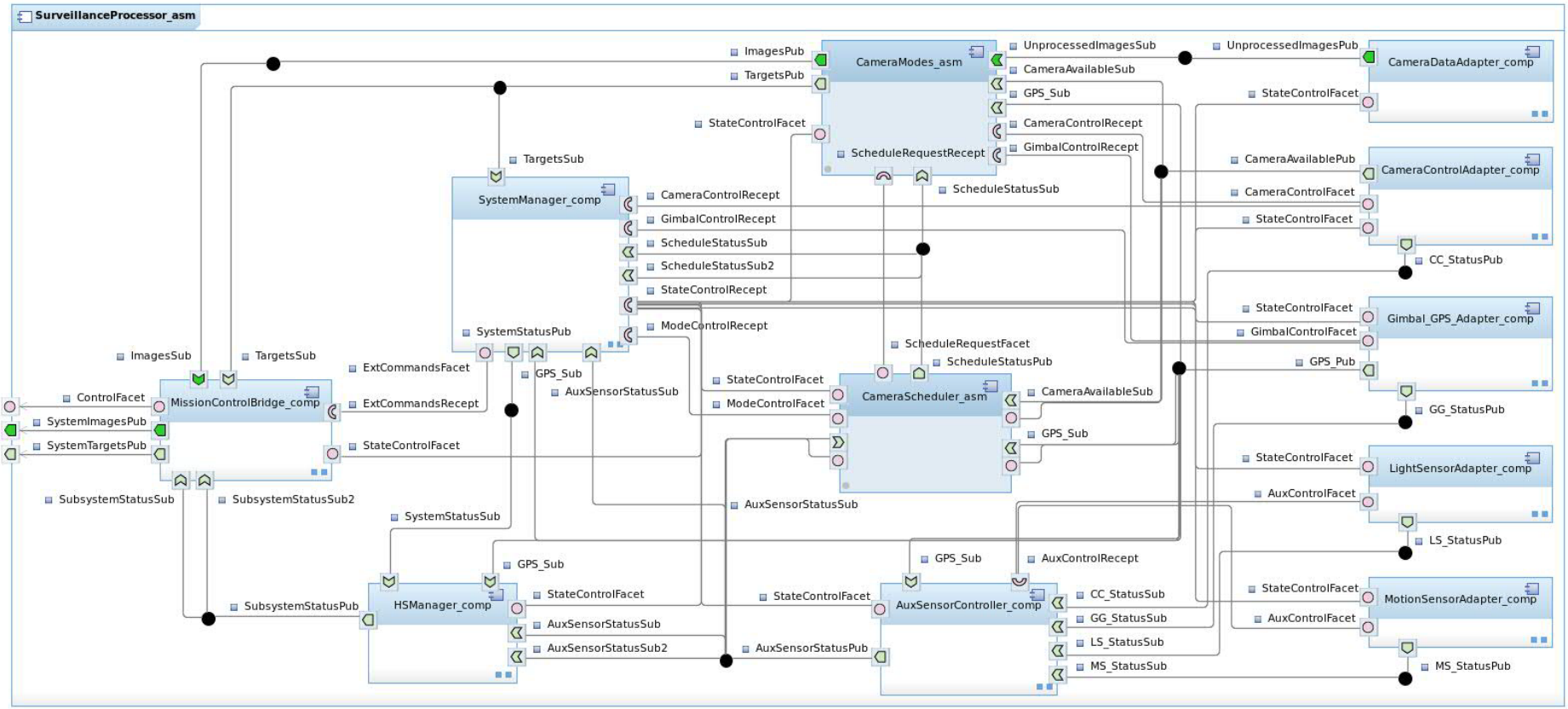
Component Assembly Example

- C&C style Component Assembly diagrams offer a “software schematic” view of your run-time structural architecture
  - Shows system run-time composition using standard “software parts”
  - Similar to hardware schematics connecting standard hardware parts
  - Connections drawn in an MDA modeling tool are automatically established during the deployment launch phase by the D&C deployment framework – big time/code savings to developers

Remedy IT

Your challenge - our solution

Component Assembly Example



Basic Port Types

- Service (Facet)
- Client (Receptacle) Sync or Async (AMI4CCM)

Extended Port Types

- DDS_Write, DDS_Update
- DDS_Listen, DDS_Read, DDS_StateListen, DDS_Get
- PSAT_Write
- PSAT_Listen
- SPDM PSAT_Base::PSAT_Write
- SPDM PSAT_Base::PSAT_Listen
- AI_Save
- Discover (Data or Services)

Remedy IT

Your challenge - our solution

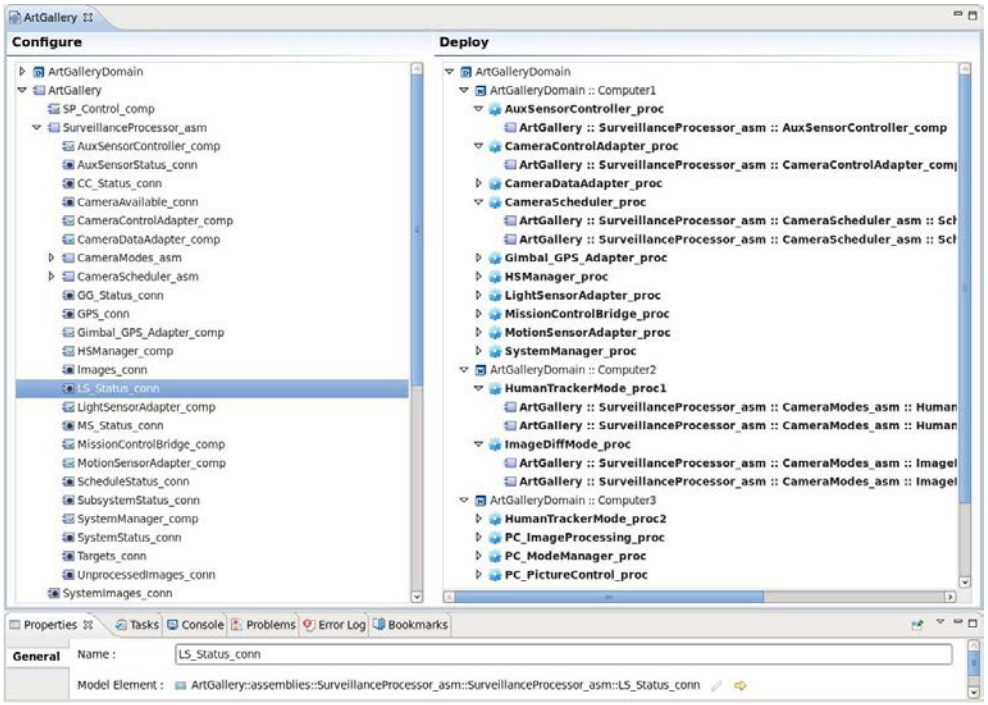
Domain and Deployment Example

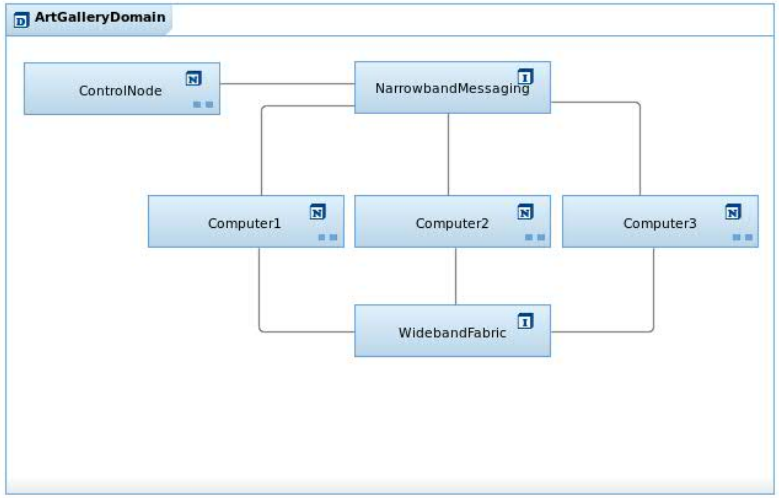
- Allocation style Domain & Deployment diagrams capture QoS, config & aspects of system resource utilization for resource allocation and concurrency
- A Deployment “diagram” maps:
  - Each component instance to a component server process (+ container)
  - Processes to compute nodes (OS instances) defined for a Domain

Remedy IT

Your challenge - our solution

Domain and Deployment Example





Remedy IT

Your challenge - our solution

CBDDS Tools Allow Domain Customization (1)

- Publish Subscribe Attachment Transfer (PSAT) connector
  - High performance, general purpose & location independent pub-sub transport of wideband data with DDS signaling
- Signal Processing Data Model (SPDM) connector
  - PSAT extension to support transport of OMG VSIPL++ or VSIPL (Vector, Signal and Image Processing Library) blocks and views for signal and image processing applications

Remedy IT

Your challenge - our solution

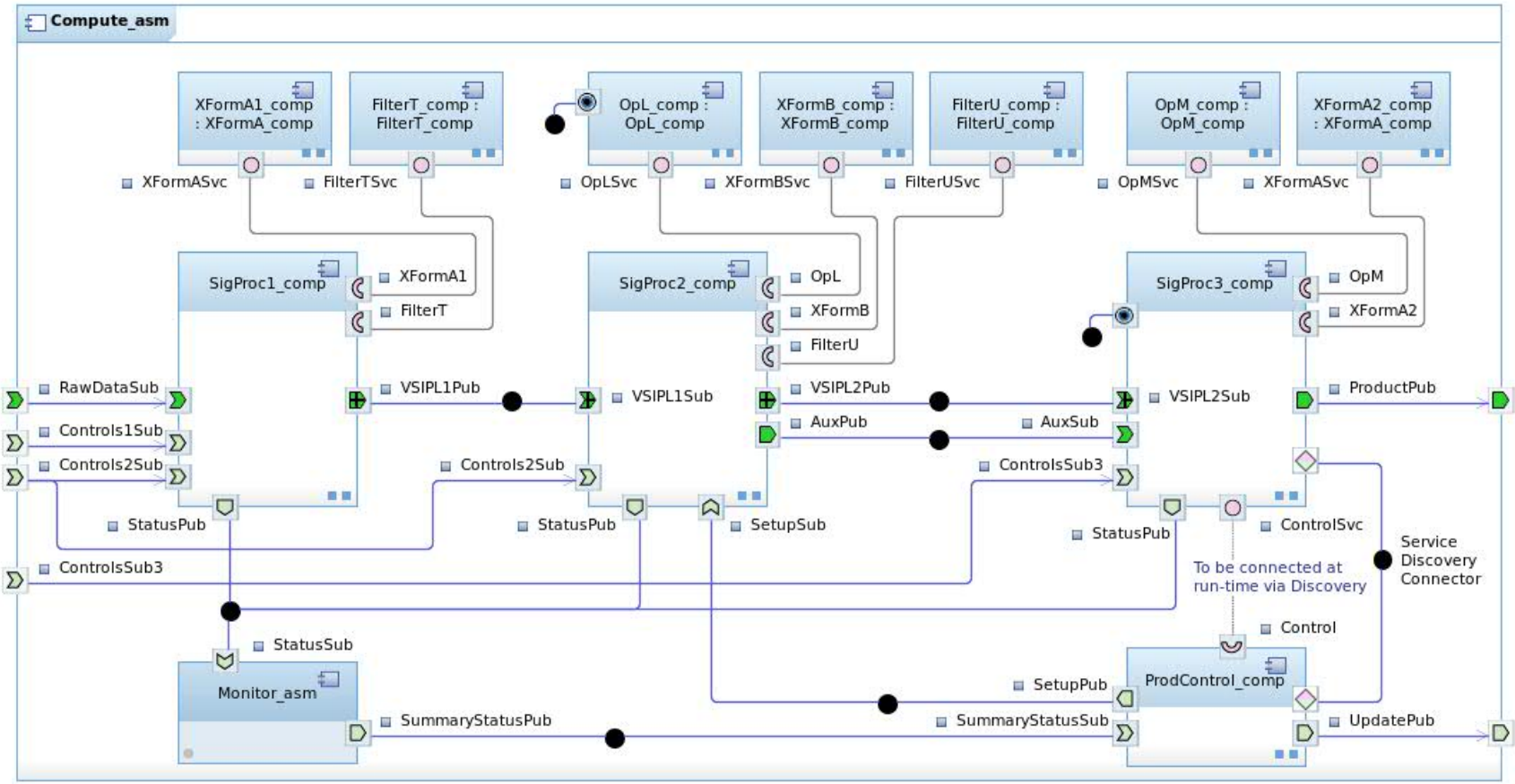
CBDDS Tools Allow Domain Customization (2)

- Application Instrumentation (AI) connector
  - CBDDS PSM simplification of the DDS PSM, providing a very easy to use encapsulation of in-development OMG AI standard for binary data instrumentation
- Discovery connector
  - Directory services access to support dynamic, run-time registration, discovery/lookup and binding of component service endpoints and topic data

Remedy IT

Your challenge - our solution

Example Model with Domain Specific Connectors



Basic Port Types

- Service (Facet)
- Client (Receptacle) Sync or Async (AMI4CCM)

Extended Port Types

- DDS_Write, DDS_Update
- DDS_Listen, DDS_Read, DDS_StateListen, DDS_Get
- PSAT_Write
- PSAT_Listen
- SPDM PSAT_Base::PSAT_Write
- SPDM PSAT_Base::PSAT_Listen
- AI_Save
- Discover (Data or Services)

Remedy IT

Your challenge - our solution

NGC Teton Project Status (1)

- SNA Platform used on 14 programs and up to 20 IRAD efforts
  - Some efforts are large and complex - 100's of components, deep assembly trees, many nodes
- Emerging themes common include…
  - Significant productivity gains during design, reduced I&T efforts (shift of focus from I&T to design)
  - Complexity & SLOC reductions (up to 56%)
  - Very high stability in executing systems
  - Shortened overall development times (= lower development costs)
  - Excellent and extremely quick application framework portability between disparate target hardware architectures

Remedy IT

Your challenge - our solution

NGC Teton Project Status (2)

- CBDDS is helping to advance and improve MDA for software engineering in general
  - CBDDS ADL proving to be an excellent means of capturing, viewing and sharing high level software architectures between disparate teams
  - Early efforts to extend and integrate with NGC systems engineering SysML community
  - CDP deployment plans are powerful, yet complex, and definitely require a tool to generate them
    - Side benefit: forces teams to keep model up to date – vs. gen documentation, abandon it & start coding

Remedy IT

Your challenge - our solution

IDL to C++11

Remedy IT

Your challenge - our solution

Why a new language mapping?

- IDL to C++ language mapping is impossible to change because
  - Multiple implementations are on the market (open source and commercial)
  - A huge amount of applications have been developed
- An updated IDL to C++ language mapping would force vendors and users to update their products
- The standardization of a new C++ revision in 2011 (ISO/IEC 14882:2011, called C++11) gives the opportunity to define a new language mapping
  - C++11 features are not backward compatible with C++03 or C++99
  - A new C++11 mapping leaves the existing mapping intact

Remedy IT

Your challenge - our solution

Goals

- Simplify mapping for C++
- Make use of the new C++11 features to
  - Reduce amount of application code
  - Reduce amount of possible errors made
  - Gain runtime performance
  - Speedup development and testing
  - Faster time to market
  - Reduced costs
  - Reduced training time

Remedy IT

Your challenge - our solution

Basic types

IDL	C++11	Default value
short	int16_t	0
long	int32_t	0
long long	int64_t	0
unsigned short	uint16_t	0
unsigned long	uint32_t	0
unsigned long long	uint64_t	0
float	float	0.0
double	double	0.0
long double	long double	0.0
char	char	0
wchar	wchar_t	0
boolean	bool	false
octet	uint8_t	0

Remedy IT

Your challenge - our solution

Some basic concepts

- Strings map to std::string
- Enums map to strongly types C++11 enums
- IDL interfaces map to so called reference type
  - References are fully reference counted, no manual reference counting anymore
- Struct/union map to C++ classes with a set of accessors
- Set of IDL traits available for template meta programming
- No name concatenation but a set of traits that have to be used

Remedy IT

Your challenge - our solution

Middleware agnostic

- IDL to C++11 is middleware agnostic
- Supports DDS, CORBA, and CCM
- Greatly simplifies development of DDS, CORBA, and CCM based applications

Remedy IT

Your challenge - our solution

Want to know more?

- Check the ORBzone.org, the community site for CORBA, CCM, and related technologies
- Check the Remedy IT provided examples at http://osportal.remedy.nl
- Tutorial and IDL to C++/C++11 comparison available at http://www.slideshare.net/RemedyIT
- Contact us, see http://www.theaceorb.nl or http://www.remedy.nl

Questions?