



Remedy IT

Your challenge - our solution

Modernizing SCA through new Object Management Group (OMG) standards

Johnny Willemsen (jwillemsen@remedy.nl)

CTO Remedy IT

<http://www.remedy.nl>



OMG Standards

- SCA can be modernized through several OMG standards
 - Upcoming standards
 - IDL4
 - Unified Component Model for Distributed, Real-Time, and Embedded Systems (UCM)
 - Available standards
 - IDL to C++11



IDL4

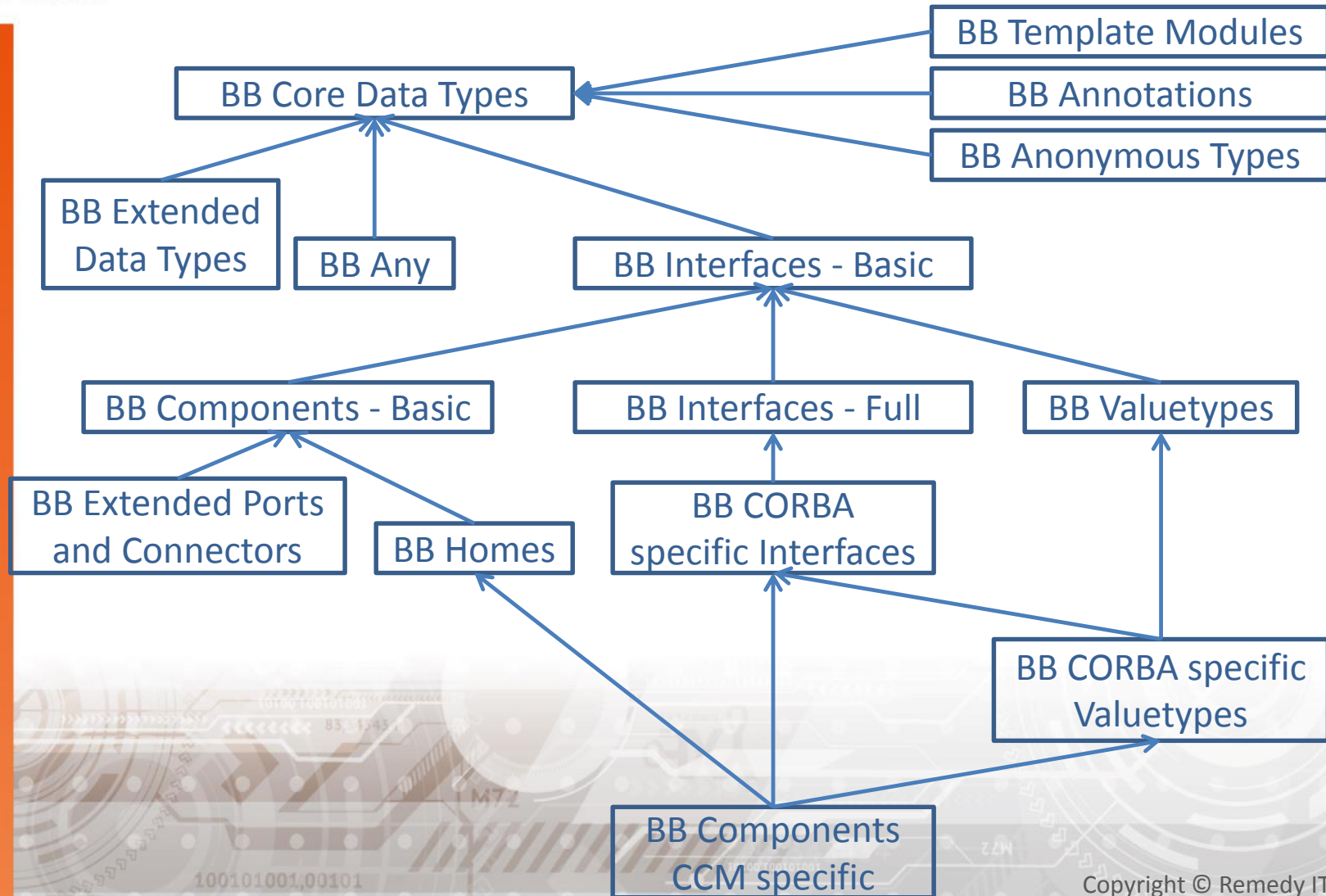
- The upcoming IDL4 standard will
 - Move IDL out of the CORBA specification and into its own OMG specification
 - Integrate this with the IDL extensions coming from DDS X-Types
 - For example annotations, map, bitset
 - Will provide a logical grouping of IDL constructs into different building blocks (BB)



Remedy IT

Your challenge - our solution

IDL4 building blocks





IDL4 and SCA

- IDL4 can simplify SCA with minimal efforts by referring to the new IDL4 building blocks instead of the full CORBA IDL specification
- Reduces complexity of the SCA specification
- First IDL4 RFC draft available; formal standard later this year
- Feedback from the SCA community on the defined building blocks is very welcome!



Remedy IT

Your challenge - our solution

Unified Component Model for Distributed, Real-Time, and Embedded systems

- New software component standard
- Evolution of LwCCM, RTC, SCA, and related efforts
- Platform Independent/Specific Model approach
- Software communication middleware agnostic
 - No mandatory dependency on CORBA or any other communication middleware standard
- Will provide an IDL4 PSM
 - All interfaces are local by default



UCM Interaction Patterns

- Define how components interact with the outside world
 - Request/Reply interaction
 - client, server, asynchronous client, and asynchronous server
 - Event interaction
 - supplier, push consumer, and pull consumer
 - State interaction
 - observable, passive observer, push observer, pull observer, and push state observer



UCM Connector Fragments

- Connector fragments realize a specific interaction pattern role:
 - Sockets, DDS, CORBA, http, ...
 - Infiniband, serial, ...
 - Existing systems, non-CBDDS systems
- Implemented or generated by you as user or by a vendor



Remedy IT

Your challenge - our solution

UCM and SCA

- Experience and input from the SCA community is very welcome
- SCA could use the UCM PIM in the future



IDL to C++11

- Simplified mapping for C++
 - Make use of the standard C++ library as much as possible
- Make use of the C++11 features to
 - Reduce amount of application code
 - Reduce amount of possible coding errors by providing a safer API
 - Gain runtime performance
 - Speedup development and testing
 - Faster time to market
 - Reduced costs
 - Reduced training time



IDL to C++11

- No ptr/var/duplicate artifacts anymore but so called reference types
 - Automatically reference counted
 - Nil reference is represented as nullptr
 - A Boolean operator for comparison is available
 - Invoking an operation on a nil reference results in an exception
- Uses std::string, std::vector, and std::array instead of defining its own types
- Instead of defining all kinds of naming rules all implied C++ types are available through IDL::traits<> including additional meta information as defined in your IDL!



IDL to C++11 and SCA

- IDL to C++11 is already referred by the SCA IDL PSM
- Application code using IDL to C++11 and the SCA IDL PSM doesn't use CORBA specific types
- IDL to C++11 reduces the need for SCA specific C++ language mappings to our idea
- The OMG allows extension of the specification to cover C++03 when there is enough interest to drive that specification and implementation work!
- Other OMG IDL language mappings can be improved; a viable alternative to create SCA specific language mappings



Remedy IT

Your challenge - our solution

Want to know more?

- Let us meet today and tomorrow here in person!
- Check our website at <http://www.remedy.nl>
 - [TAOX11](#), our CORBA implementation supporting IDL to C++11
 - [RIDL](#), our open source IDL compiler front end
 - [AXCIOMA](#), our component framework
- Call us, +31-10-5220139