REAL AGILITY:

Lessons learnt from implementing MDA

Dr Clive Boughton and Carol Boughton



Outline

- Our path to success
- Real Agility with MDA
- Our (real world) experiences
- Today's path to success
- Our next step

xtUML is MDA

Our Path to Success

- Our learning path
- Our resources
- Benefits to our customers & us

xtUML is MDA

Our Learning Path

- Structured Analysis & Structured Design
- Object-Oriented Analysis
  - Shlaer-Mellor - OOA (*similar to OMT*)
  - Analysis (with tool) then manual translation via *design* to *code*
  - Analysis (with tool) then manual translation direct to *code* within a chosen (simple) architecture
- Action Language (*introduced with tools*)
- *xt*UML (*UML subset plus action language*)
- Vendor Model Compilers
  - Limited languages (C/C++)
- Own Model Compilers
  - Ada95 and Java

xUML is MDA

Our Resources

- **People**
  - Small team on any project
  - Excellent Analysts
- **Methodology**
  - In-depth software engineering knowledge
  - Methods chosen to suit problems
- **Projects**
  - Internal
  - Consultancies and Contracts

AUML is MDA

Benefits to our customers & us

- Clear and well abstracted analysis models
- Increased productivity
- Greater flexibility and robustness
- Early verification and validation
- Early delivery

xUML is MDA

Real Agility with MDA

- **Agility not Agile Development**
  - Earlier delivery of simulations/code to the customer
  - Increments/releases in models - continuous capture of requirements
  - Refactoring - Who wants to do it with code?
  - "Pair modeling"
  - IP is in the models
- **MDA and *xt*UML**
  - We were doing MDA 10 years ago (*Model → Design → Code*)
  - ***xt*UML automates MDA**

xtUML is MDA

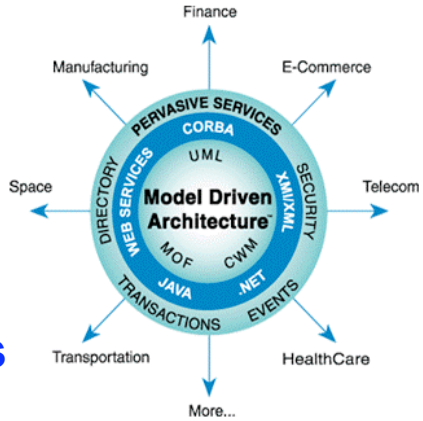
Agility NOT Agile Development

- Working tenets
  - Low overhead software development
  - Workable requirements
  - Thinkers over processes and tools
  - Customer collaboration
  - Welcoming change
- Demonstrated agility
  - Analysis models that business people can understand and comment on
  - Minimal documentation - its mostly in the models
  - Requirements changes
    - easily demonstrated with models
    - very visible to customer
  - Rapid turn-around time to re-generate and re-deploy application

XUML is MDA

MDA and *xt*UML

- ***xt*UML is MDA**
  - Construct models using UML subset
  - Execute models to verify them
  - ‘Compile’ models according to translation rules
  - Architecture drives compiler

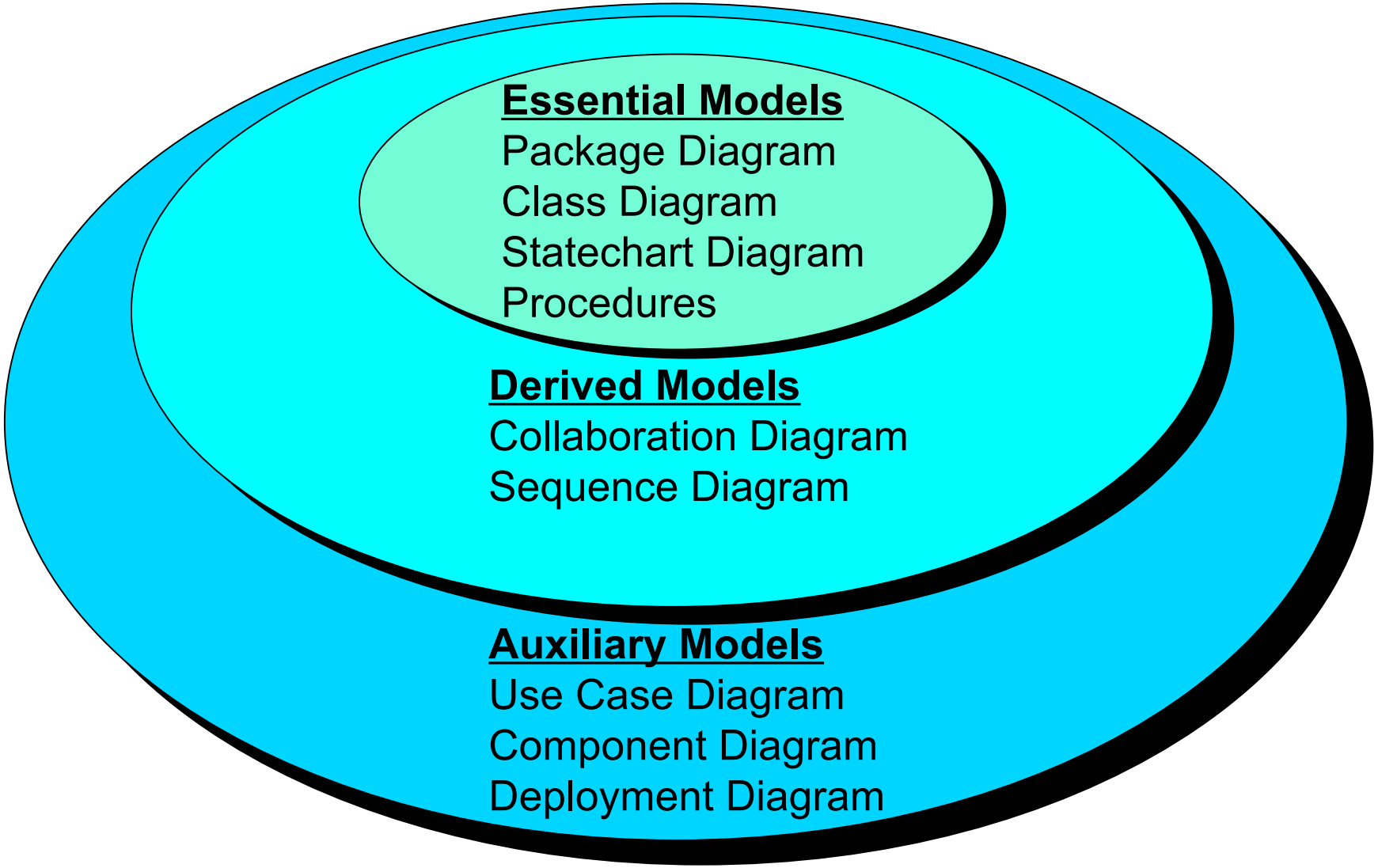


- ***xt*UML tools need**
  - Model Constructor
  - Verifier / Simulator
  - Model Compiler

xtUML is MDA

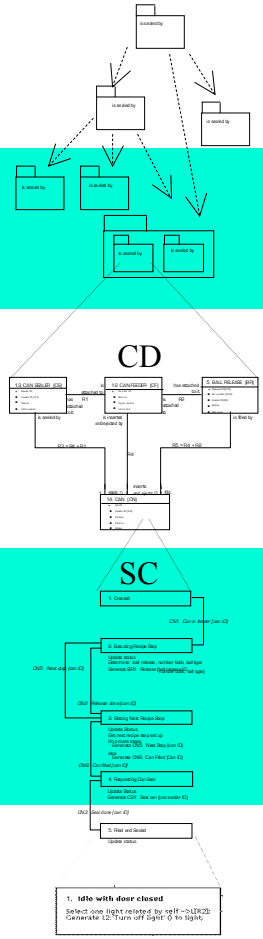
Unified Modeling Language™

What is *xt*UML



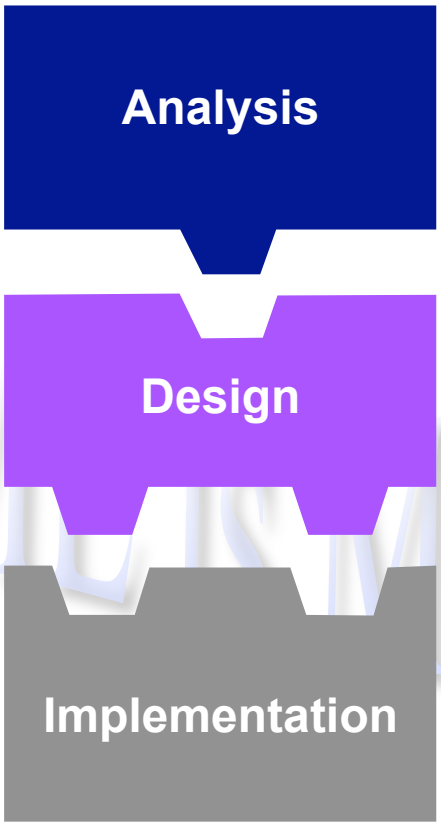
*xt*UML Models a Well-Defined Hierarchy

1. Domains (*separate concerns*)
2. Subsystems (*within domains*)
3. Classes (*Data*)
4. Lifecycles (*State*)
5. Action Specification (*Behaviour*)



Analysis and Design

This is where the engineer captures what the software needs to do.



*xt*UML Model(s)

*xt*UML Model Compiler

This is where the engineer designs the architecture and the details therein.

Various implementation technologies may be used to construct the software - including OS, Language, Network, Database, GUI, etc.

*xt*UML Target Code

xUMLFORUM

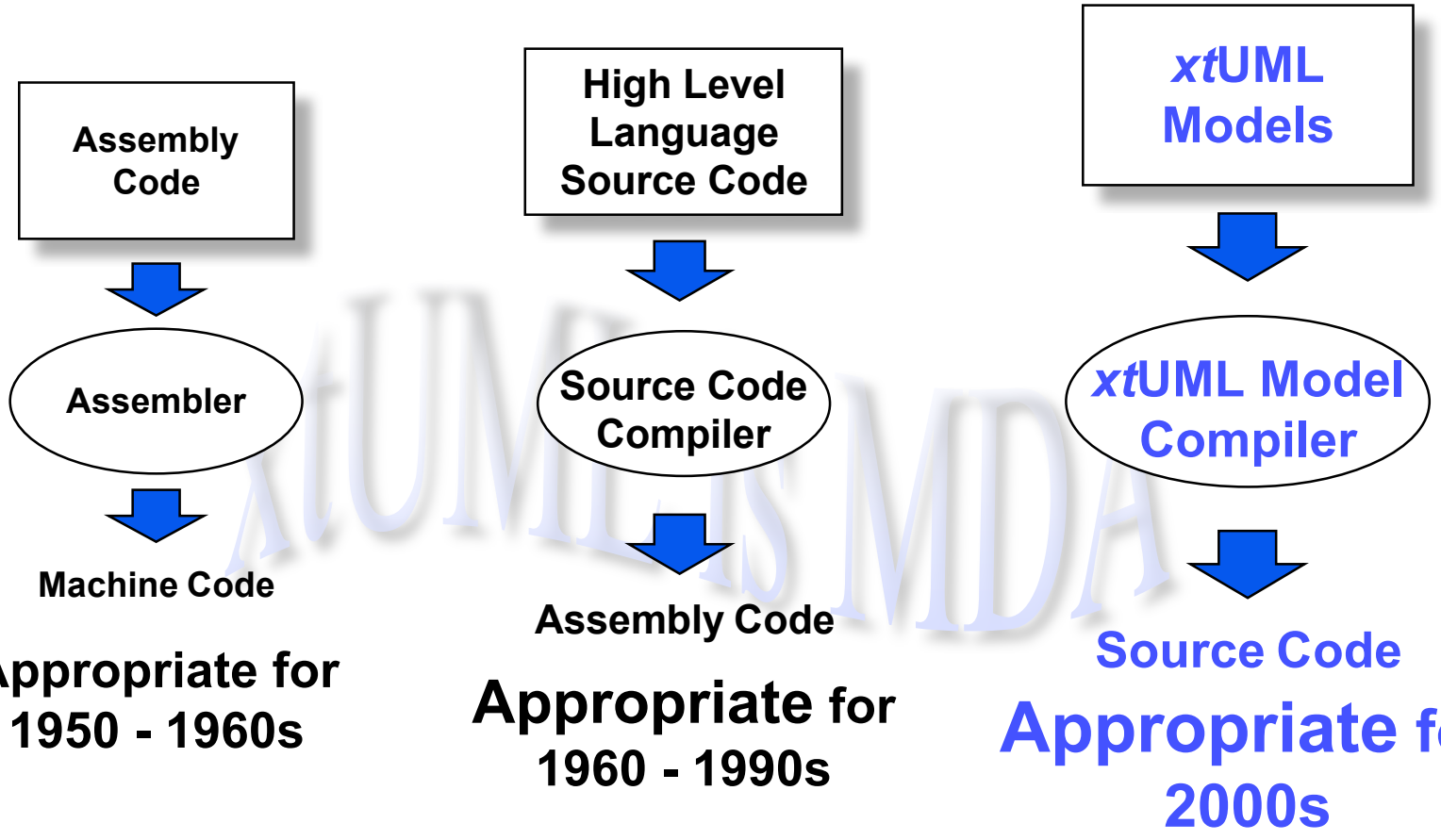
Analysis and Design

- **Analysis**: Thinking about all aspects of a system
  - What are the separate concerns - aspects
  - What are the processes, operations, functions etc.
  - What are the data, material or information content of the system
  - What are the modes and states
  - When is anything to be done and how often

- **Design**: How to structure system using analysis information, quality attributes and constraints
  - Decide on architecture - driven by quality attributes
  - Decide on ways in which to map functional requirements into architecture
  - Provide alternatives

XUML is MDA

Model Based Development



xtUML is MDA

Our (real world) experiences

- Used in many domains
- What's been valuable to us?
- Issues we've had to overcome

xtUML is MDA

Used in many domains

- Business system (government)
- Defence
- Elections
- Emergency services and security
- Medical
- Traffic control
- Transport logistics
- Software engineering
- Ubiquitous computing

xtUML is MDA

TAIL
(Transport Airlift Integrated Logistics)

- The TAIL model is a simulation of the air transport of resources to remote locations in an emergency.



  - High Detail
  - Air Transport only
  - Remote Location
  - Context : Emergency
    (possible infrastructure damage, time critical)

xtUML is MBD

TAIL Model

- In a natural disaster, a variety of emergency resources are needed, and quickly.
  - Natural Disaster
    - Cyclone
    - Volcano
    - Tsunami
    - Earthquake
  - Mode(s) of Transport
    - Ship
    - Truck
    - Rail
    - Air





AUML is M

TAIL Model

- **The modeling of the transportation of these resources is complicated by several issues:**
  - **Heterogeneous Cargoes**
  - **Weather**
  - **Infrastructure Capabilities**
  - **Not limited to Civilian Transport (all available)**



TAIL Model

- **Given :**
  - Set of available aircraft (with certain characteristics)
  - Sets of cargo to be transported (and routes)



- **Then :**
  - How long will it take to transport cargo and evacuate people?
  - Can all cargo be transported?
  - Where are the bottlenecks?
  - How much will it cost?



xtUML is MDA

Table Driven Simulation

- All simulation initial conditions in tables
  - Aircraft Specifications & Numbers
  - Transport Nodes (Infrastructure)
  - Routes
  - Cargo Items
    - Atomic and Bulk Cargo
  - Weather



- Easy to change initial conditions, re-run and compare results

XUML is MDA

Results in XML

- **Full simulation audit trail**
  - Snapshot of initial conditions
  - Tracking of individual cargo items
  - Aircraft use and locations
  - Cost breakdowns
  - Snap shot of final conditions
  - Aggregated costs and usage



- **Flexible**
  - Customer requirements for results not known
  - XML allowed easy extraction of items of interest

XML is MDA

TAIL Conclusions



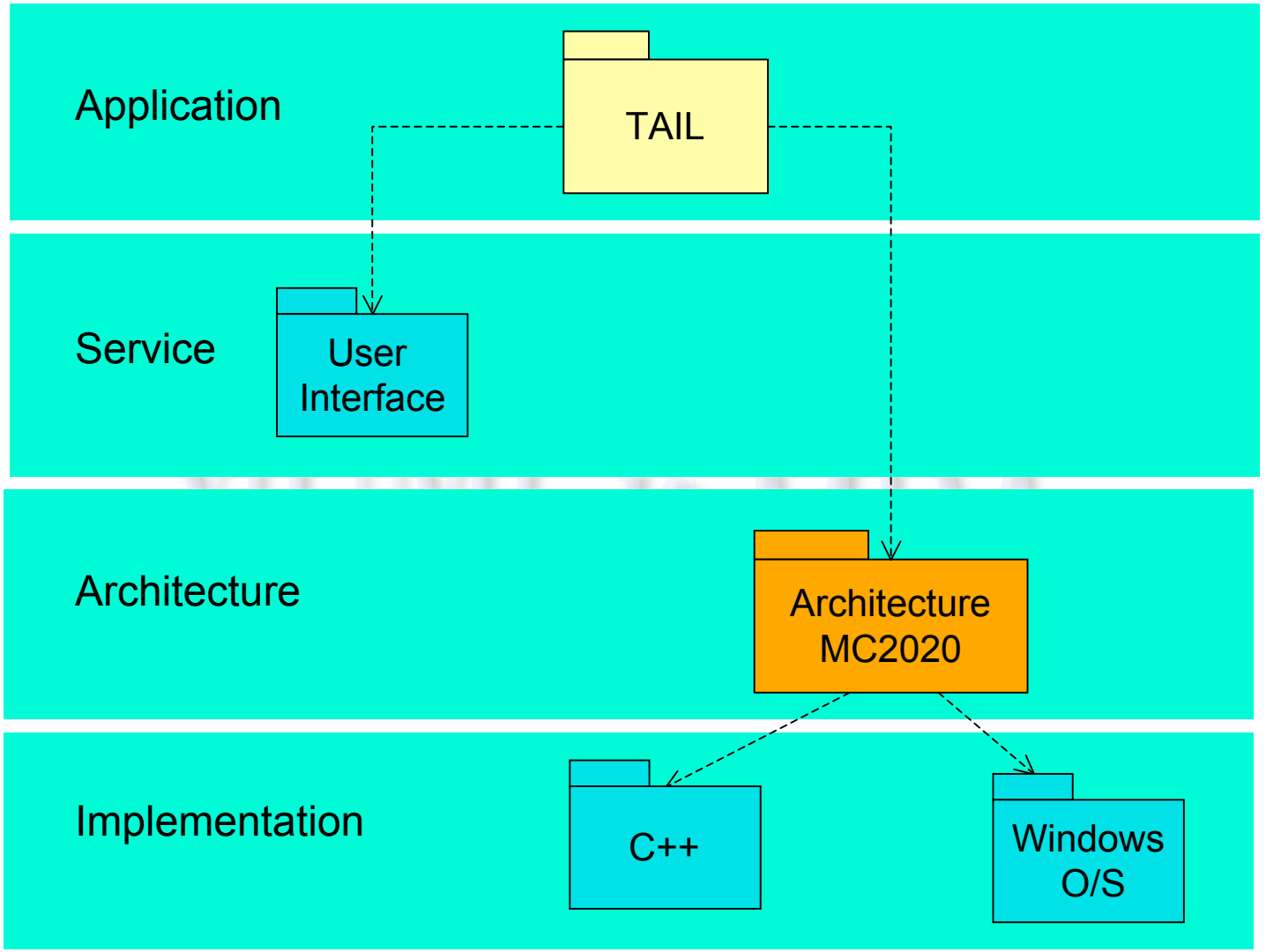




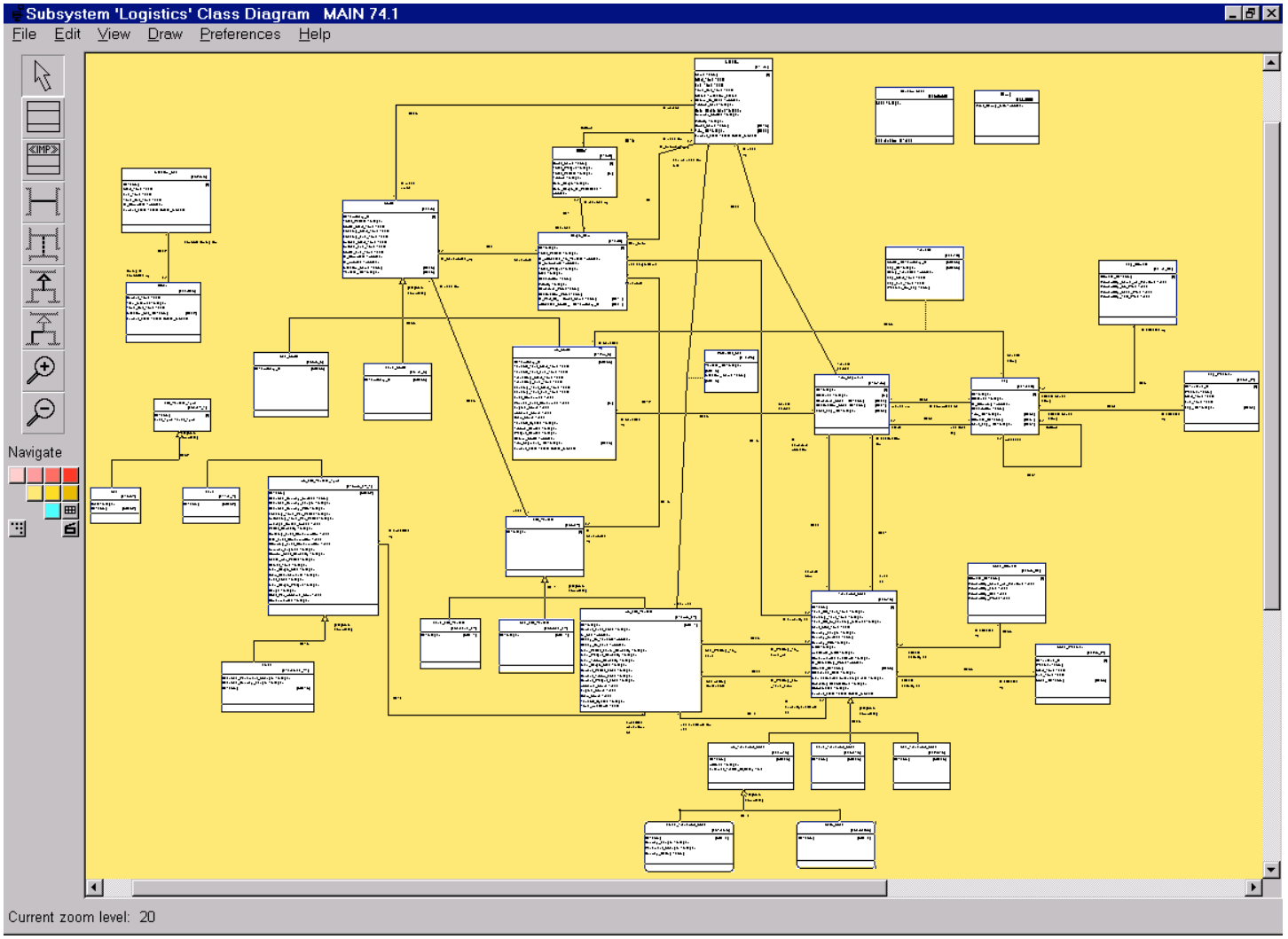
- Infrastructural Limitations
- A few capable aircraft are better than many less capable
- Aircraft with own offload capability very valuable
- Good Air Traffic Control early on a 'force multiplier'
- Improvements to Pavement Classification Number (PCN) recommended

XUML is MDA

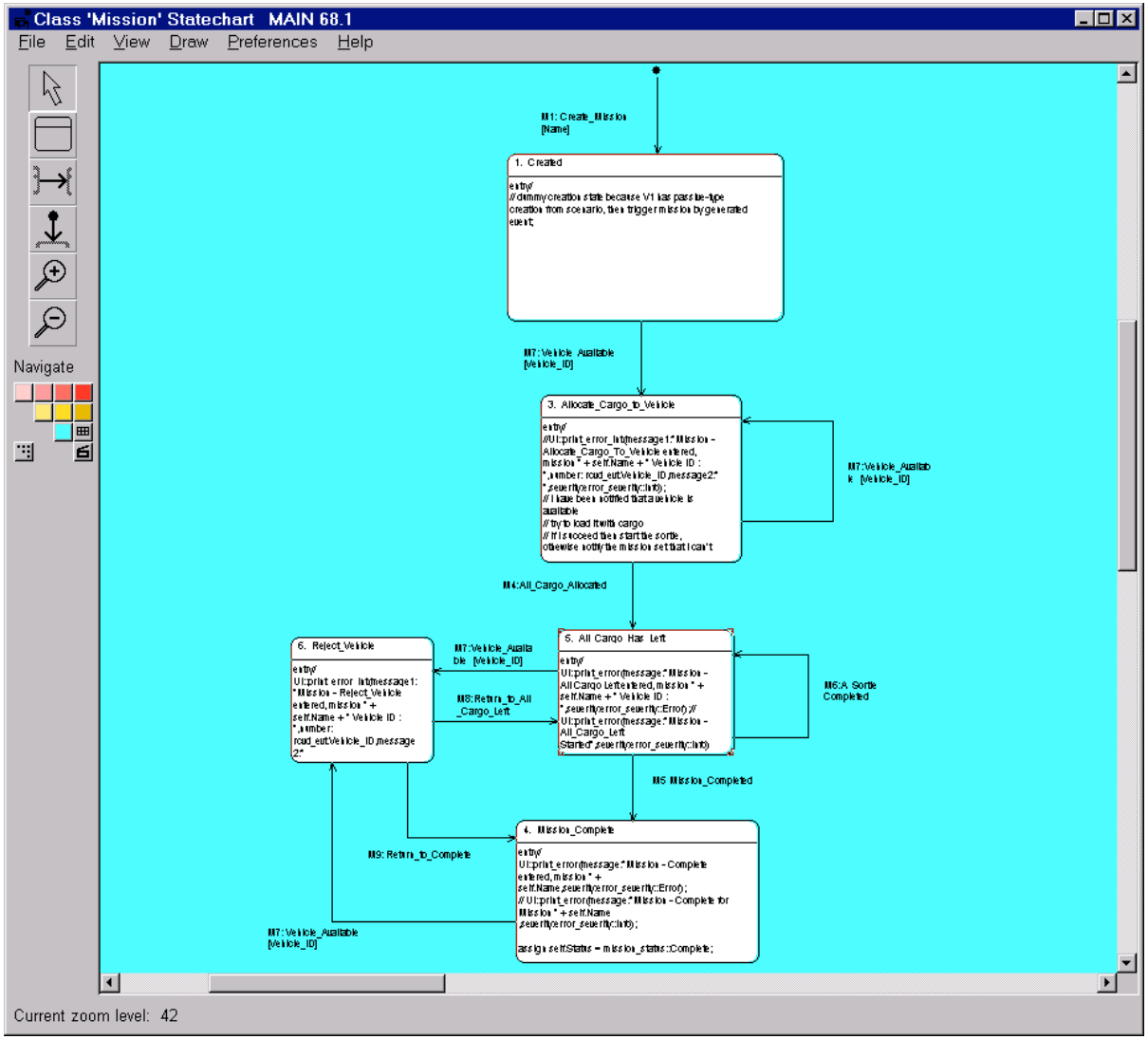
TAIL Domains



Class Diagram



State Transition Diagram



Action Language

Domain 'TRANSPORT' Statechart 'Logistics::Mission_Set' Description and Action Specification

State: 7. Attempt_Vehicle_Allocation

```
…
if not empty vehicle
  // unrelate from all existing possible uses (now redundant as vehicle has moved)
  select many possible_uses from instances of PU where selected.Vehicle_ID == rcvd_evt.Vehicle_ID;
  if not empty possible_uses
    for each possible_use in possible_uses
      select any mission related by vehicle->M[R243];
      unrelate vehicle from mission across R243 using possible_use;
      delete object instance possible_use;
    end for;
  end if; // if none that's OK

 //find trips that start where vehicle is located
  select many trips from instances of Trip where selected.Departure_NodeID ==
     vehicle.Currently_Available_At_ID;
  if empty trips
…
```

ACCUML is MDA

TAIL Metrics (1)

- 6 elapsed weeks
- 10 effort weeks (domain expert & modeler)
  - A standard simulation was still incomplete after 2 years
- 34 classes
  - 6 active (have state model)
  - 28 passive (some place-holders for extensions)
- 39 states

xtUML is MDA

TAIL Metrics (2)

- 1676 lines action language
- 18061 lines generated C++ (*excl. mechanisms*)
- 1566 lines C++ hand coded (*input table driven*)
- 116 builds (*half in last week*)

xtUML is MDA

TAIL Issues

- Late changing requirements (*final week*)
- Late delivery of scenarios from customer
- Weather model ended up not being required
- Model “too accurate”

xtUML is MDA

TAIL Outcomes

- Early feedback to customer (weekly milestones & deliveries)
- IP capture at high level of abstraction
- Project successful
- Unexpectedly useful results
  - Sensitivity to infrastructure capability
  - Sensitivity to minor aircraft capability differences

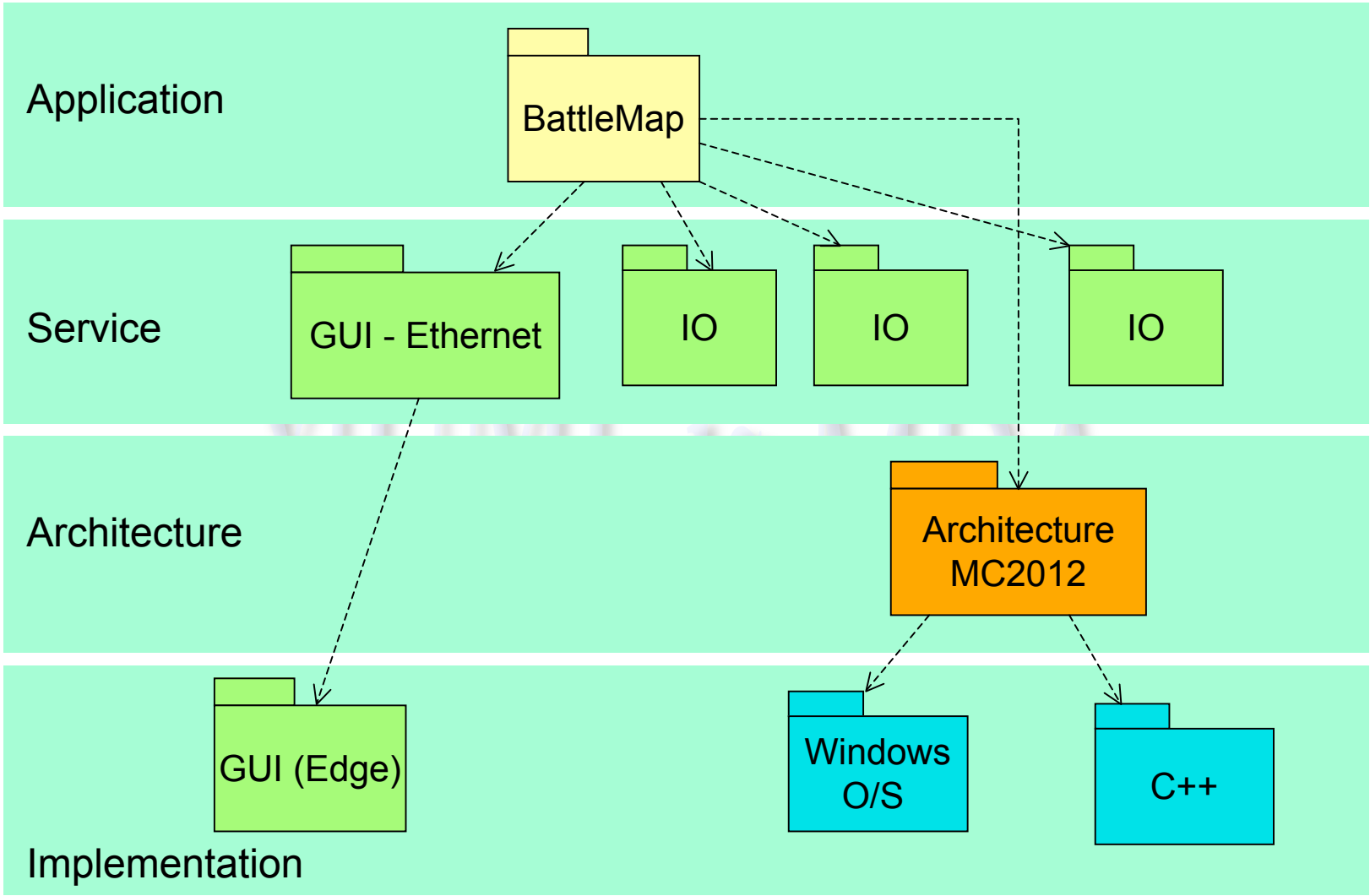
XUML is MDA

BattleMap™

- Development undertaken by ATI
- Software Improvements provided training and consultancy services

XUML is MDA

BattleMap™ Domains



BattleMap™ Metrics

- **Initial Delivery Metrics (1 domain modeled)**

Objects	44 active 13 passive
States	234
Relationships	37
Action Language	14,633 SLOC
Generated Code (MC-2012)	
Total	96,893 (C++)
Domain only	85,876 (C++)
Development Time	1,059 hours

xUML is MDA

BattleMap™ Outcomes

- Used a vendor supplied model compiler
- Integrated generated code with 3rd party GUI
- Initial system - 2 developers over 6 months
  - one developer worked on application domain
- Initial system delivered at end of planned analysis phase for traditional development

- *Estimate of time to develop using traditional methods – 10 person years*

UML is MDA

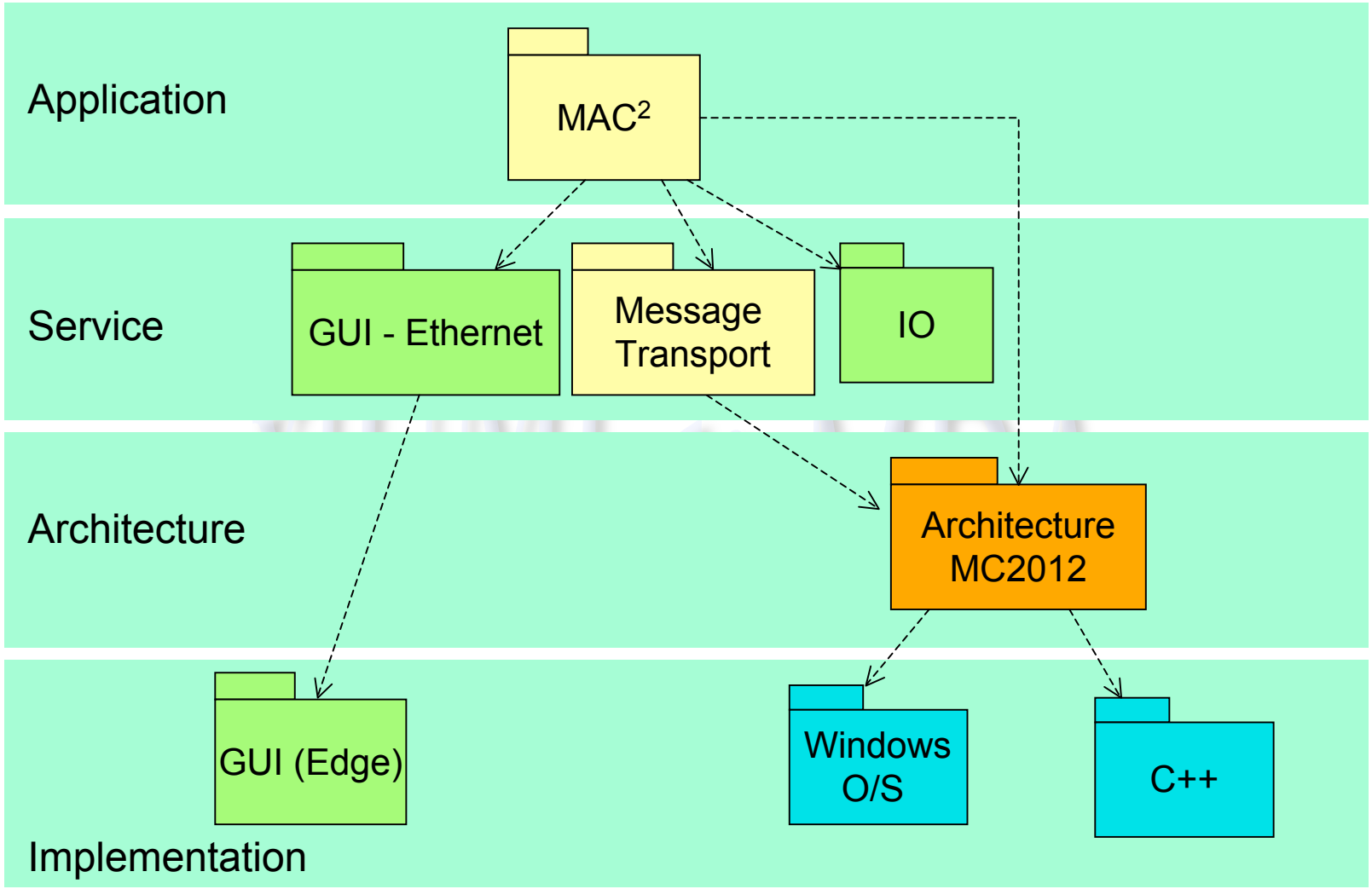
MAC²
(Mobile Asset Command and Control)

- Real-time mobile digital communications network

- Another development where Software Improvements provided consultancy services

xtUML is MDA

MAC² Domains



MAC² Metrics

- **Message Transport Domain**
  - Objects 19
  - States 205
  - Relationships 26
  - Generated Code(C++) 63,542 SLOC
  - Hours 807
- **MAC² Domain**
  - Objects 5
  - States 26
  - Relationships 3
  - Generated Code (C++) 10254 SLOC
  - Hours 250

xtUML is MDA

SoDIS

- **Software Development Impact Statement**
  - Tool for identifying stakeholders and the impact of requirements on each stakeholder or group of stakeholders
- **Internal development**
  - Developed in Java for use on multi-platforms
  - Modeled and then hand-translated using defined translational rules
    - No Java model compiler available from vendor

XUML is MDA

SoDIS Metrics

- 77,515 Java NBNC SLOC
- 1462 Function Points (FP)
  - Based on Capers Jones
- Average US productivity* = 3.8 FP/staff mth
  - Based on Capers Jones
- SIPL productivity = 38.3 FP/staff mth

* Varies with size of project

UML is MDA

Business Management System (BMS)

- Enterprise application
- Government department
- Evaluation and management of research proposals and projects
- Three stages:
  - First stage: Six users, 72 including observers
  - Second stage: 75 users, all internal
  - Third stage: 50,000 users, internal and external

XUML is MDA

BMS - Use of MDA

- Analysis models that business experts can understand and comment on
- Class diagrams, state charts
- Model review sessions have been conducted with success
- Building Java model compiler

xUML is MDA

BMS - Code Generation

- ***xt*UML using Kennedy Carter iUML/iCCG**
- **100% code generation in the database tier, static model components, e.g. classes, attributes, etc**
- **100% code generation in the enterprise tier, static and dynamic model components, i.e. classes, attributes, and action language**
- **0% code generation in the web tier**
- **Turn-around time less than 10 minutes to re-generate and re-deploy application (*not including data migration*)**

xUML is MDA

eVACS®





- Original development
- MDA approach to the Voting Module
- Full redevelopment with MDA
  - Including development of Ada Model Compiler

xUML is MDA

eVACS-*i* Features



- Core xtUML models to provide modules for
  - Election set-up
  - Voting (for standalone, closed LAN & internet)
  - Counting
  - Data Entry (for paper ballots including scanning)



- Specific model for each electoral jurisdiction
  - Requires minor changes to core model to generate system for new jurisdiction quickly, easily and reliably



eVACS photographs courtesy of ACT Electoral Commission

xUML is MDA

Ada Model Compiler

- Generates Ada95 source code from an *xt*UML model for a specific target
- Code is complete - no manual coding required
- Code produced can be compiled and run on any O/S that has an Ada95 compiler

xUML is MDA

Ada Model Compiler Metrics

- 23,342 Ada NBNC SLOC
- 476 Function Points (FP)
  - Based on Capers Jones
- Average US productivity* = 4.5 FP/staff mth
  - Based on Capers Jones
- SIPL productivity = 38.1 FP/staff mth

AUML is MDA

* Varies with size of project

Advantages of MDA (*xt*UML)

Project	Actual Effort (PM)	Hand Coding (FP/PM)	Modeling plus Hand Coding (FP/PM)	Modeling plus Model Compiling (FP/PM)	PI	US* Hand Coding (FP/PM)
BattleMap™	8.0			242	23	0.9
MAC²	8.0			185	22	1.5
TAIL	2.5			157	23	2.5
SoDIS™	31.5		38		12	3.8
Ada Model Compiler	12.5	38			13	4.5

* Capers Jones - Applied Software Measurement - 1997

What's been valuable to us

- Separation of concerns
- Ability to understand all aspects of system
- Simplifies, condenses extent and perspective of system
- Abstraction capability
  - Natural human tendency to complicate with more
- Having access to domain knowledge
- Using "pair modeling"

xtUML is MDA

Issues we've had to overcome

- Ensuring accurate abstraction of the (business) requirements
- Getting people to understand 'less' is better than 'more' for objects
  - While separating concerns
- Deficiencies of the available tools
  - They ain't perfect but they work and add significant value!

xUML is MDA

Today's path to success

- What's changed in 10 years
- What hasn't changed
- To adopt *xt*UML now
- Software developer roles
- *xt*UML resources

xtUML is MDA

What's changed in 10 years

- More trained people available
  - Part of some university software engineering programs
- More tools to consider
  - But, action language not available in most
- Choice for 'real agility'
  - Action language a feature of only a couple of tools
  - Model compilers - limited language choice
- Local success stories across many domains

xUML is MDA

What hasn't changed

- **You don't need to invest heavily or take unnecessary risks to start using *xt*UML**
  - **even on small projects**
  - **but tools (such as Nucleus BridgePoint and iUML) can provide significant value by reducing costs and schedule further**
- **Models can be developed and translated into code manually and with the use of simple tools**
  - **the process is not that far removed from current practice**
- **Finding people with a high level of abstraction capability is still HARD**

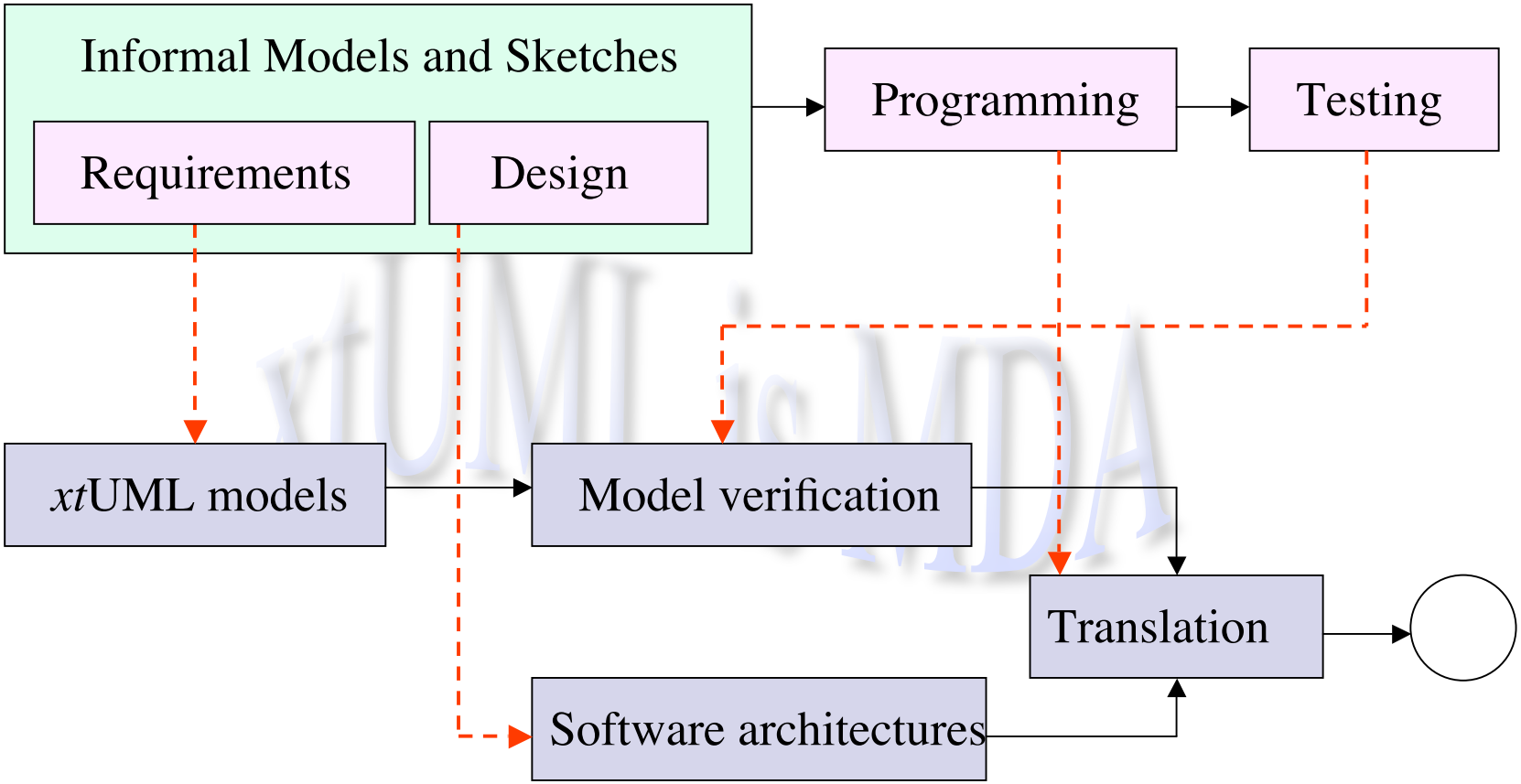
xtUML is MDA

To adopt *xt*UML now

- **Analysts need to develop more formal *xt*UML models**
- **Testers need to move towards the front of the software development lifecycle**
  - **testing (executable) models with analysts as they are developed**
  - **early learning and feedback**
- **Designers need to develop reusable software architectures rather than specific designs**
  - **While analysts and testers are developing and verifying the requirements models**
- **Programmers translate *xt*UML models into code by following reusable software architectures rather than implementing a specific design**

xUML is MDA

Software Developer Roles



*xt*UML Resources

- People
  - The ANU teaches *xt*UML to more than 100 students per year
- Tools
  - Modeling versus code generation
- Literature
- Training
  - Short courses from Software Improvements
  - Graduate and post graduate software engineering degrees - ANU
  - Vendor tool training
    - from vendors
    - from Australian distributor
- Support services
  - Provided locally

xtUML is MDA

Our next step

- More Model Compilers
  - Java (on the way)
  - SPARK (for trusted systems)
- Architecture & Meta Modeling
  - As part of BMS project
  - Research at ANU
- Aspect Oriented Thinking (Shayne Flint)
  - Multidisciplinary problems
  - Formalising the weaving of aspects

AUML is MDA

Real agility means

- Using formal models to capture requirements
- Keeping information & IP in the models
- Changing the models - not the code
- Automating the implementation processes
- Adding real cost savings, value and quality

xtUML is MDA

If we were starting today we'd

- Identify if we had someone with high-level abstraction skills
  - If not, (at least) find an excellent graduate with *xt*UML knowledge
- Give that person access to *xt*UML information
  - Training, Leon Starr's books (Object Models; Executable UML)
- Start with a 'smallish' project
- Provide support
  - Pair the person with a domain expert ("pair analysing")
  - Access external xtUML expertise
- Use available model compilers to begin with
- Build a model compiler only when
  - Project really requires language not supported by an available model compiler and external assistance is accessible, or
  - At least a couple of projects have been successfully completed and the *xt*UML expert has a comprehensive understanding of how model compilers function.



xtUML is MDA