

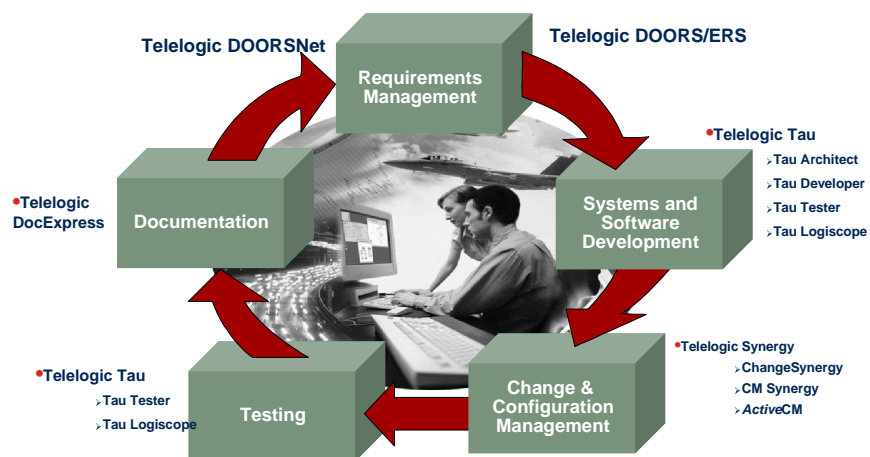
Telelogic

Back to the Drawing Board: Visualizing Requirements Using Graphical Models

Chris Sibbald
Director, Application Engineering Eastern Region

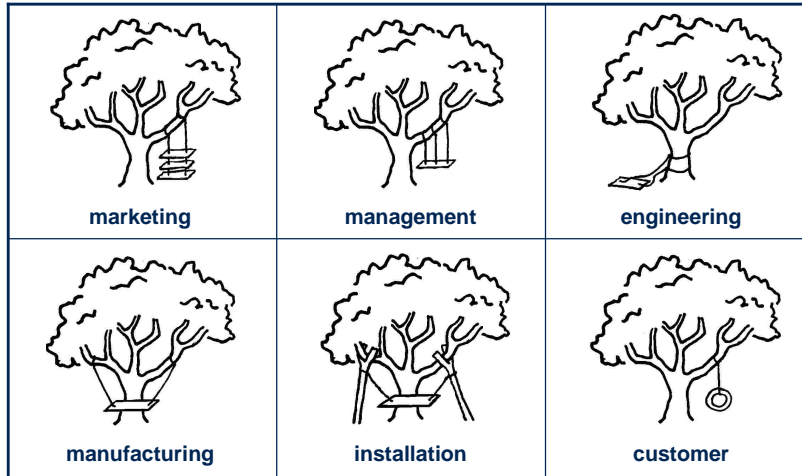
Telelogic

An Automated, Integrated and Visual Development Environment



swing, n

9. A seat suspended from above, as by ropes, on which one can ride back and forth for recreation.



Sources: www.dictionary.com and www.businessballs.com/treeswing.htm

3

© Telelogic AB **Telelogic**

The Requirements - DOORS

The screenshot shows the DOORS software interface. The title bar reads "Formal module '/MFCs/System requirements' current 0.0 - DOORS". The menu bar includes File, Edit, View, Insert, Link, Analysis, Table, Tools, Tau UML Suite, TauSDL, user, and Help. The toolbar contains various icons for file operations and editing. The main window is divided into two panes. The left pane shows a tree view of system requirements:

- System requirements
 - 1 Radar data processing
 - The system shall receive radar messages
 - The system shall assign a track ID to each track
 - The system shall check the IFF code to determine whether to engage or not
 - The system shall accept and process Engage commands
 - The system shall issue fire commands
 - 2 Missile System
 - The system shall connect to a missile system
 - The system shall interact with the missile system to fire missiles at targets
 - 3 Display System
 - The system shall be able to connect to an optional display subsystem
 - The system shall display radar tracks on a CRT display

The right pane shows a table of system requirements:

ID	System requirements
1	1 Radar data processing
2	The system shall receive radar messages containing Target Track Information (Position, Heading, Height, Speed, Identification, Friend or Foe (IFF) Code) for all tracks from the radar system.
3	The system shall assign a track ID to each track
4	The system shall check the IFF code to determine whether to engage or not
10	The system shall accept and process Engage commands
11	The system shall issue fire commands
12	2 Missile System
17	The system shall connect to a missile system
13	The system shall interact with the missile system to fire missiles at targets
14	3 Display System
15	The system shall be able to connect to an optional display subsystem
16	The system shall display radar tracks on a CRT display

The status bar at the bottom indicates "Username: Administrator" and "Exclusive edit mode".

4

© Telelogic AB **Telelogic**

The Problem



5

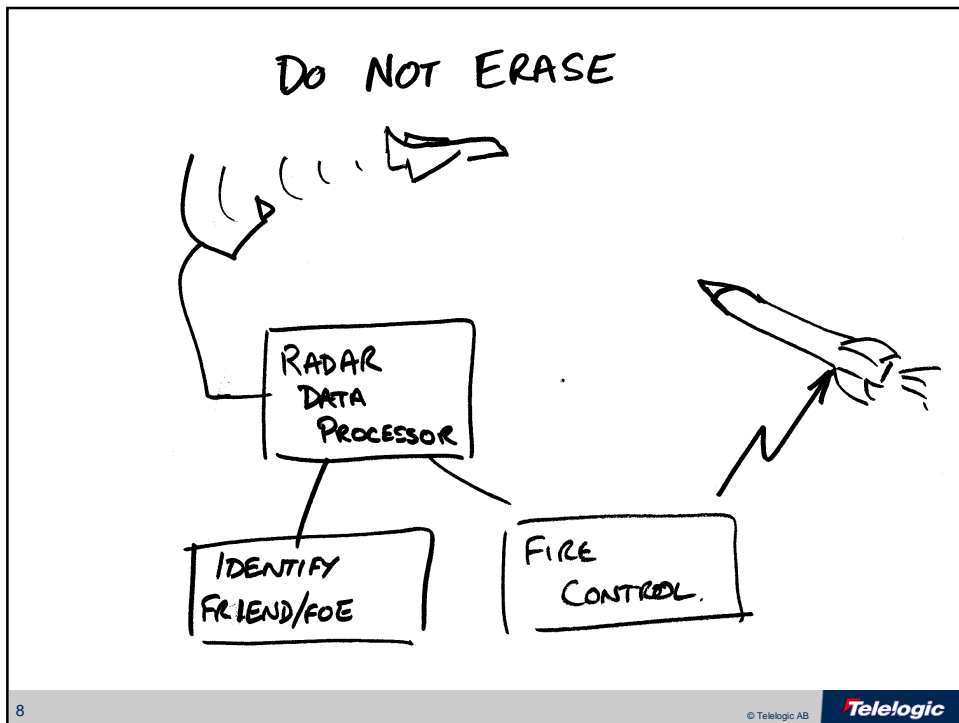
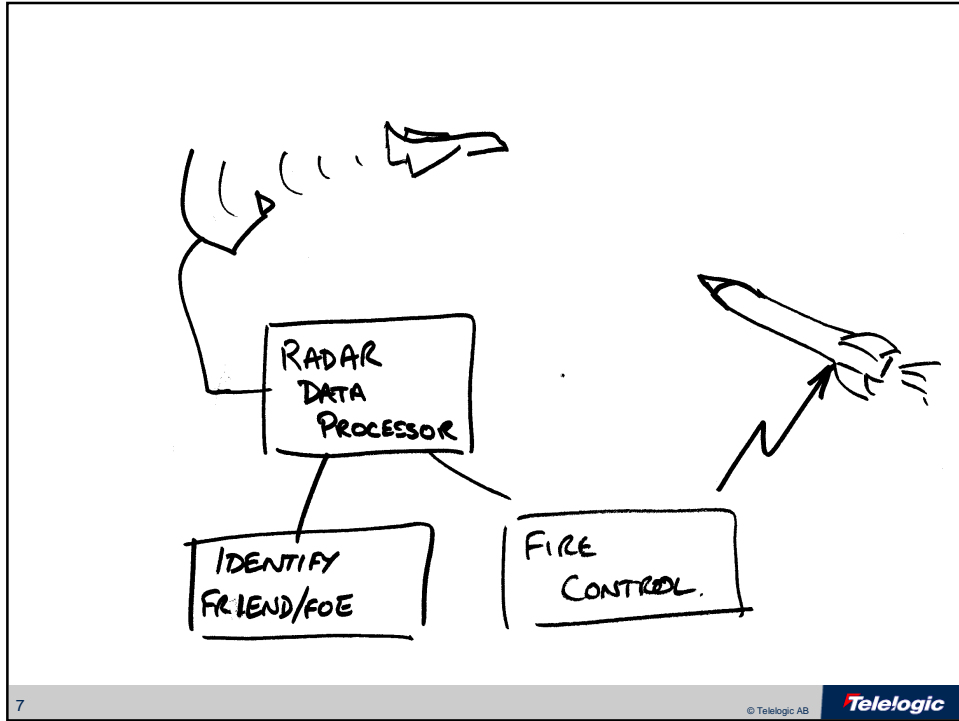
© Telelogic AB **Telelogic**

Do You Visualise Requirements Today?

- When capturing or understanding requirements, do you create a picture in your mind of how the system should work?
- In requirements gathering workshops do you produce storyboards or whiteboard scenarios?

6

© Telelogic AB **Telelogic**



What is UML™?

- UML - Unified Modeling Language™ is a visual language for
 - Specifying
 - Visualizing
 - Constructing
 - Documenting

... systems and software

- As systems grow in size and complexity, so does the need for models
- Standardized by the Object Management Group™ (OMG™)



9

© Telelogic AB

Telelogic

The Evolution of UML

- UML 1.0: Originally targeted at Software Engineering
 - No support for specifying complex architectures
 - No well-defined component support
- UML 2.0
 - Designed to address all limitations of UML 1.0
 - Improved visualization of requirements
 - Describe complex interactions
 - Improved support for large-scale systems development
 - Definition of components
 - Architecture specification
 - Describe System interactions
 - INCOSE and OMG joint initiative
 - "UML for Systems Engineering"



10

© Telelogic AB

Telelogic

Telelogic - A Leader in Defining UML 2.0

- Co-Chair of 3 sub-committees
- Initiator and key member of U2 Partners consortium



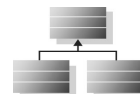
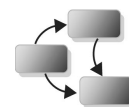
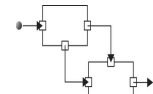
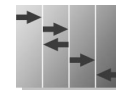
11

© Telelogic AB



Popular UML diagrams types

- Use Case diagrams
 - Capture the intended behavior of the system to be implemented and the way in which users interact with the system being built
- Sequence diagrams
 - Show the time ordering of events exchanged between parts of a system as it executes a particular scenario
- Architecture diagrams*
 - Describe the architecture of systems in terms of components and communication paths and interfaces
- Statechart diagrams
 - Describe the precise behavior of specific components as state machines
- Class diagrams
 - Define the static structure of systems in terms of classes and their relationships, as well as the data structures.

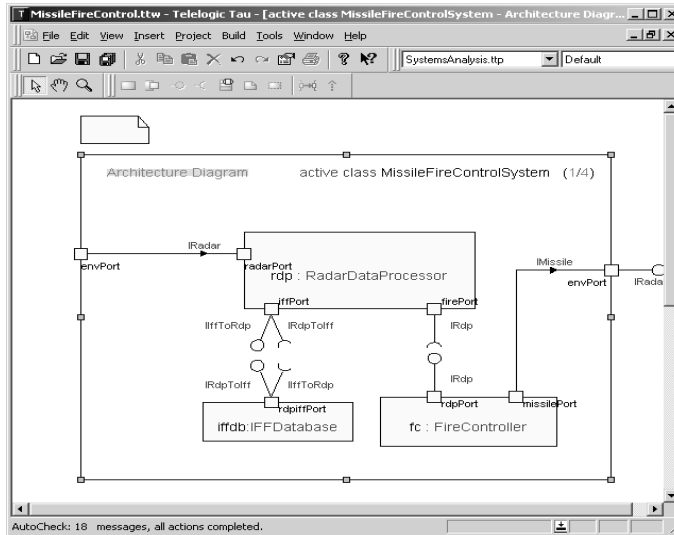


12

© Telelogic AB



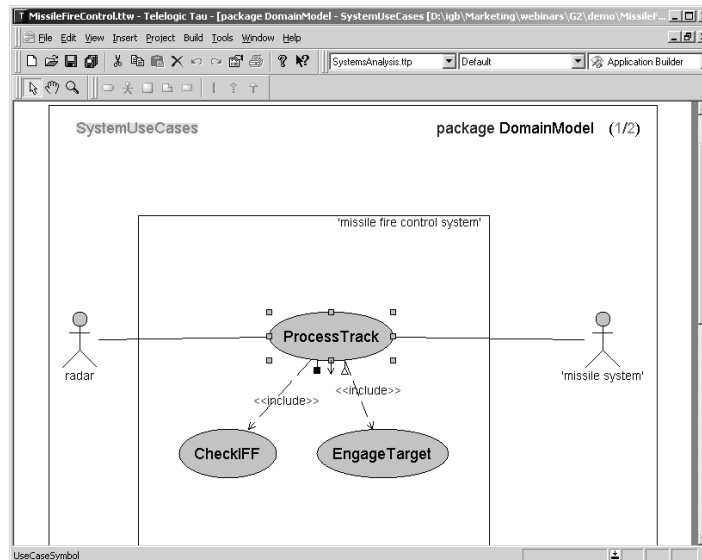
UML 2.0 Architecture



13

© Telelogic AB **Telelogic**

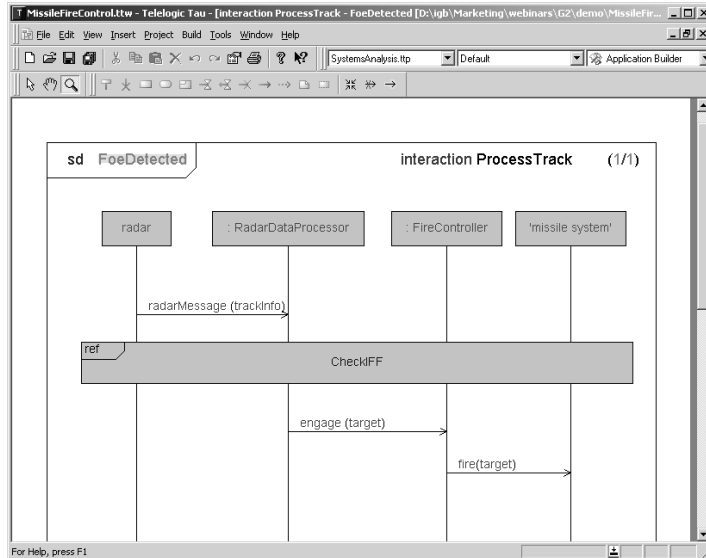
Describe How the System is Used



14

© Telelogic AB **Telelogic**

Describe How the System Interacts

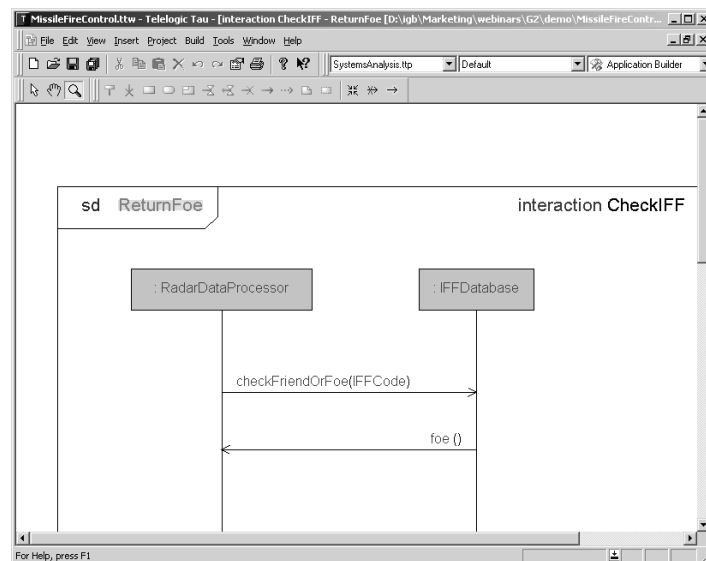


15

© Telelogic AB

Telelogic

Interactions in More Detail...

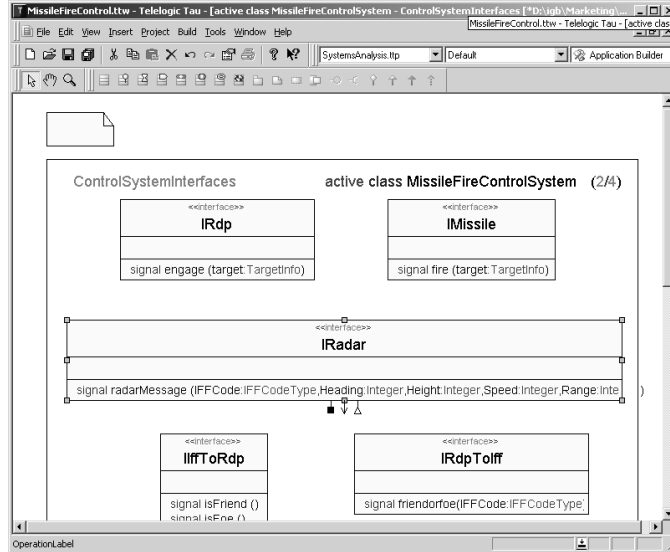


16

© Telelogic AB

Telelogic

Defining Interfaces

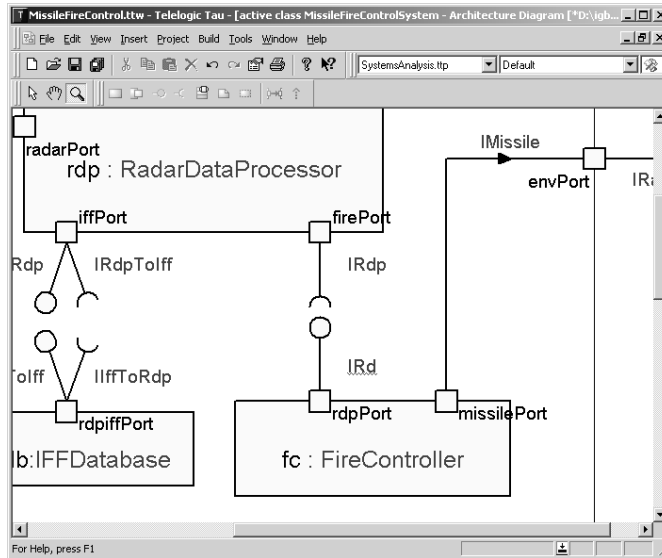


17

© Telelogic AB



Dynamic Error Checking

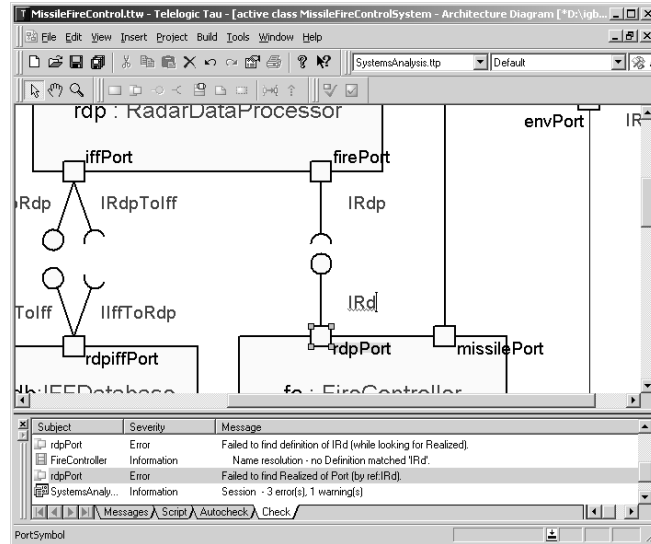


18

© Telelogic AB



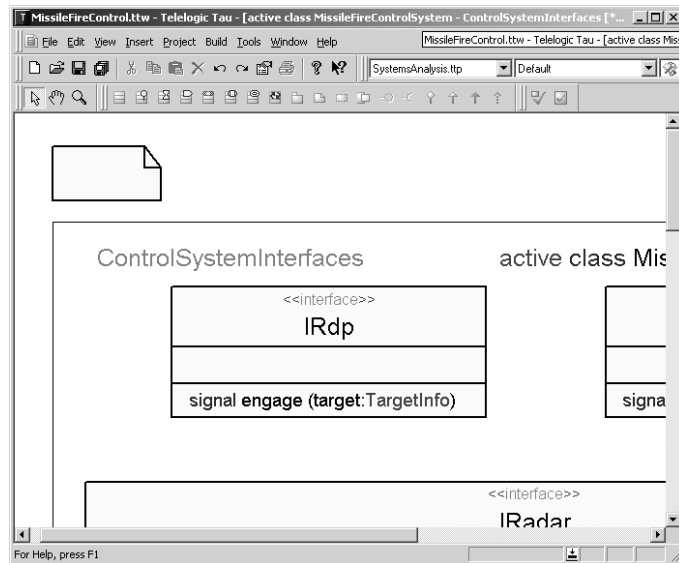
Formal Error Checking



19

© Telelogic AB **Telelogic**

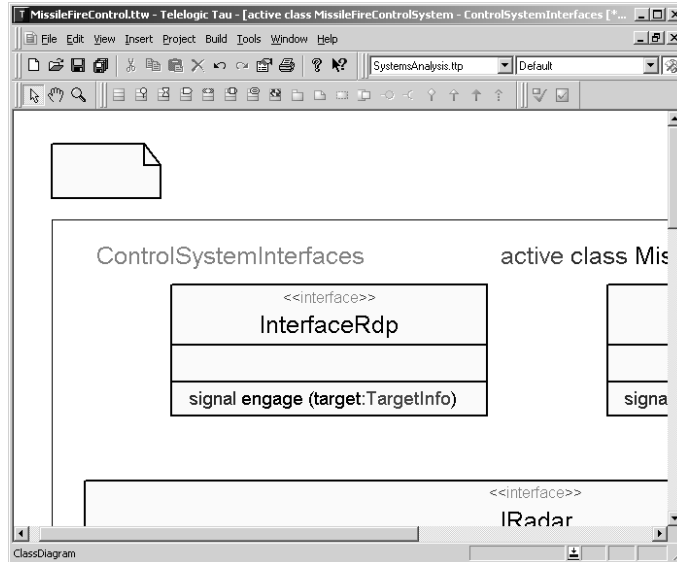
Making Changes – 1



20

© Telelogic AB **Telelogic**

Making Changes – 2

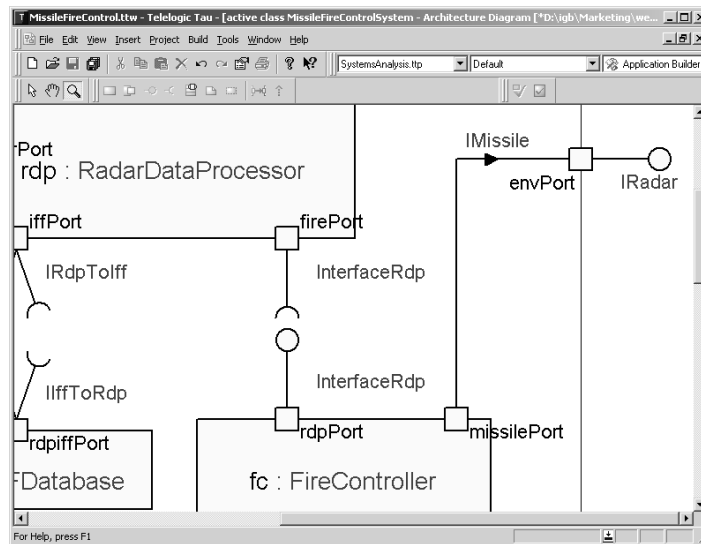


21

© Telelogic AB



Making Changes – 3



22

© Telelogic AB



Integrated CM

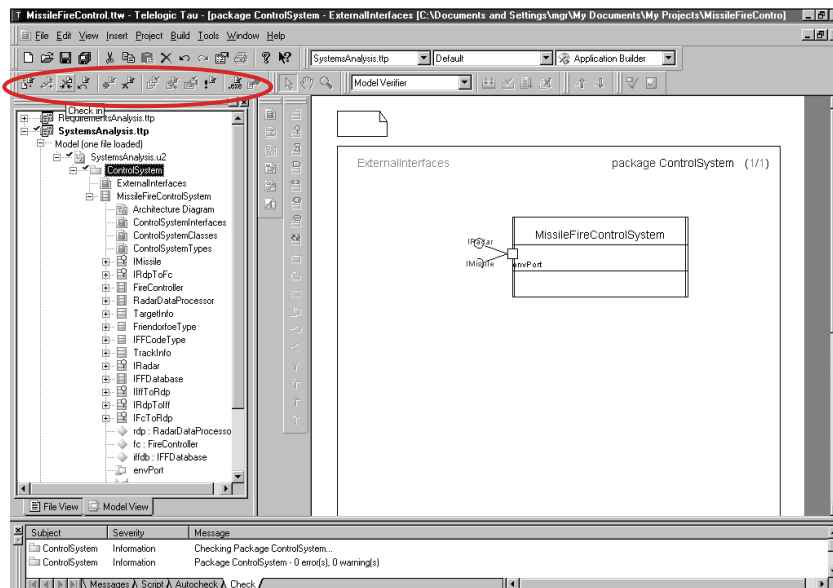
- Easy and accessible
 - Automatic detection of available CM
 - Automatic creation of new models under CM
 - Visual feedback on CM status (checked in/out) in browser
- Context-sensitive menu
 - Check in/out files from source control; Undo checkout
 - Get latest version
 - Add/Remove selected files
 - Refresh status; Show history of files; Show file differences
 - Display file's source control properties
 - Invoke source control system
 - Import module from source control



23

© Telelogic AB

Telelogic

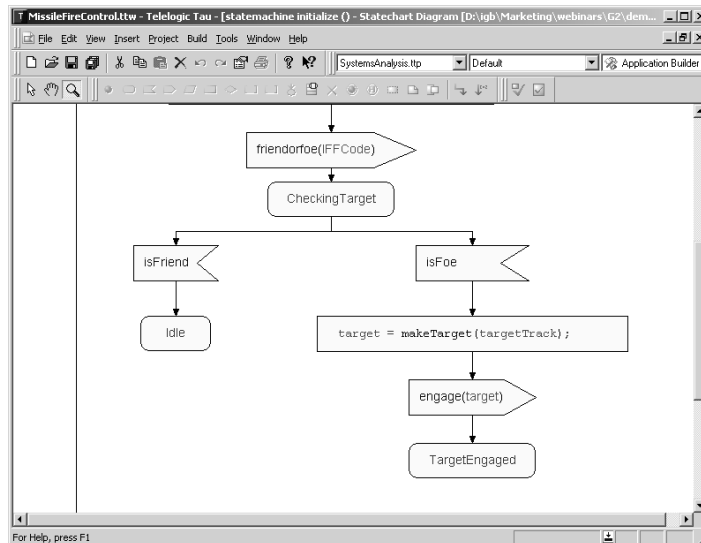


24

© Telelogic AB

Telelogic

Specifying Behaviour



25

© Telelogic AB

Telelogic

Model Verifier

- Full model simulation
 - Without software coding
- Send signals into system to stimulate it
- Animate state machines
- Examine variables and signal queues
- Capture sequence diagram trace
 - Save as record and for documentation
- Re-test against previous simulation trace

26

© Telelogic AB

Telelogic

MissileFireControl.ttw - Telelogic Tau - [Interaction UseCase2 - (2) [no file]]

File Edit View Insert Project Verify Build Tools Window Help

SystemsAnalysis.ttp Default Application Builder

Sequence diagram trace generated by Tau for MissileFireControlSystem

sd (2) interaction UseCase2 (1/1)

env_0 rdp{1} fc{1} iffdb{1}

RequirementsAnalysis.ttp

SystemsAnalysis.ttp

- ReadyQueue (3 elements)
 - #1 = rdp{1} <catz
 - #2 = fc{1} <catz
 - #3 = iffdb{1} <catz
- TimeQueue (0 element)
- MissileFireControlSystem
 - rdp #1 <start> >St
 - fc
 - ifdb
 - env

MissileFireControlSystem.rdp{1}:
 MissileFireControlSystem.fc{1}: fr
 MissileFireControlSystem.ifdb{1}
 output from env{1} to MissileFireC
 MissileFireControlSystem.rdp{1}:
 MissileFireControlSystem.ifdb{1}

Sender Signal

env{1} :MissileFireControlSystem:radarh
 ...

Loading ModelVerifier Profile: C:\PROGRA~1\NTE

LifeLineSymbol

27 © Telelogic AB Telelogic

MissileFireControl.ttw - Telelogic Tau - [Interaction UseCase2 - (2) [no file]]

File Edit View Insert Project Verify Build Tools Window Help

SystemsAnalysis.ttp Default Application Builder

Sequence diagram trace generated by Tau for MissileFireControlSystem

sd (2) interaction UseCase2 (1/1)

env_0 rdp{1} fc{1} iffdb{1}

Idle Idle Idle

RequirementsAnalysis.ttp

SystemsAnalysis.ttp

- ReadyQueue (0 element)
- TimeQueue (0 element)
- MissileFireControlSystem
 - rdp #1 <idle>
 - fc
 - ifdb
 - env

MissileFireControlSystem.rdp{1}:
 MissileFireControlSystem.fc{1}: fr
 MissileFireControlSystem.ifdb{1}
 output from env{1} to MissileFireC
 MissileFireControlSystem.rdp{1}:
 MissileFireControlSystem.ifdb{1}

Sender Signal

env{1} :MissileFireControlSystem:radarh
 ...

*** NEXTSTATE Idle
 *** TRANSITION START
 * PID : iffdb:1
 * State : start state
 * Now : 0 0000
 *** NEXTSTATE Idle

NextStateActionOccurrenceSymbol

28 © Telelogic AB Telelogic

MissileFireControl.ttw - Telelogic Tau - [interaction UseCase2 - (2) [no file]]

File Edit View Insert Project Verify Build Tools Window Help

SystemsAnalysis.ttp Default Application Builder

env_0 rdp[1] fc[1] iffdb[1]

```

sequenceDiagram
    participant env_0
    participant rdp as rdp[1]
    participant fc as fc[1]
    participant iffdb as iffdb[1]

    env_0->>rdp: radarMessage(AA, 1, 2, 3, 4)
    activate rdp
    rdp->>fc: friendorfoe(AA)
    activate fc
    fc->>iffdb: isFriend()
    activate iffdb
    iffdb->>fc: 
    deactivate iffdb
    fc->>rdp: 
    deactivate fc
    rdp->>env_0: 
    deactivate rdp
  
```

*** TRANSITION START
 * Pid : rdp:1
 * State : CheckingTarget
 * Input : isFriend
 * Sender : iffdb:1
 * Now : 0.0000
 *** NEXTSTATE Idle

MessageLine

29 © Telelogic AB **Telelogic**

MissileFireControl.ttw - Telelogic Tau - [statemachine initialize () - Statechart Diagram [D:\igb\Marketing\webinars\G2\demo\MissileFireC...

File Edit View Insert Project Verify Build Tools Window Help

SystemsAnalysis.ttp Default Application Builder

```

stateDiagram-v2
    [*] --> friendorfoe
    friendorfoe --> friend : isFriend
    friendorfoe --> foe : isFoe
    friend --> idle : 
    foe --> idle : 
  
```

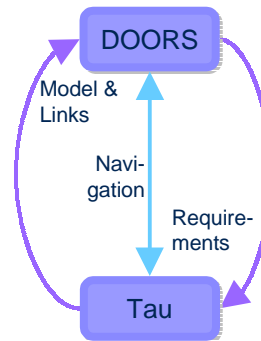
*** TRANSITION START
 * Pid : rdp:1
 * State : CheckingTarget
 * Input : isFriend
 * Sender : iffdb:1
 * Now : 0.0000
 *** NEXTSTATE Idle

For Help, press F1

30 © Telelogic AB **Telelogic**

DOORS-Tau/Architect Integration: Role-based Approach

- DOORS specialists work inside DOORS
 - Create UML models and UML elements
 - Navigate from requirements to UML
 - View lifecycle traceability
- UML specialists work inside Tau
 - View requirements
 - Create and navigate links
- Both sides have access to what is created on the other



31

© Telelogic AB **Telelogic**

Telelogic DOORS®/ERS Overview

- One integrated suite offers requirements management with:
 - A powerful, easy to use, requirements management database
 - A web-based interface for less frequent, remote users
 - A Microsoft-Word based editor for short-term access to small volumes of requirements information
- Established tools means DOORS/ERS components are already tried, tested and trusted
- Market leader
 - Standish Group report “What are Your Requirements 2003”
- Technology Leader
 - Named by French industry analyst firm Yphise
 - “DOORS is the market leader and the most mature product with the broadest set of rich features”

33

© Telelogic AB

Telelogic

Tau/Architect Overview

- For specification and modeling of advanced systems
- Enables systems engineers to visualize system requirements
 - Integration with Telelogic DOORS®
 - Supports architecture, interface control and component specification via UML
- Support for dynamic model execution
 - Facilitates early requirements verification
 - Powerful simulation of system behavior to eliminate errors early
 - Without the need to write source code fragments
- Model-driven, multi-user tool
 - All UML diagrams are fully interconnected
 - Instant error feedback and automatic consistency
 - Seamless support for collaboration and distribution

34

© Telelogic AB

Telelogic

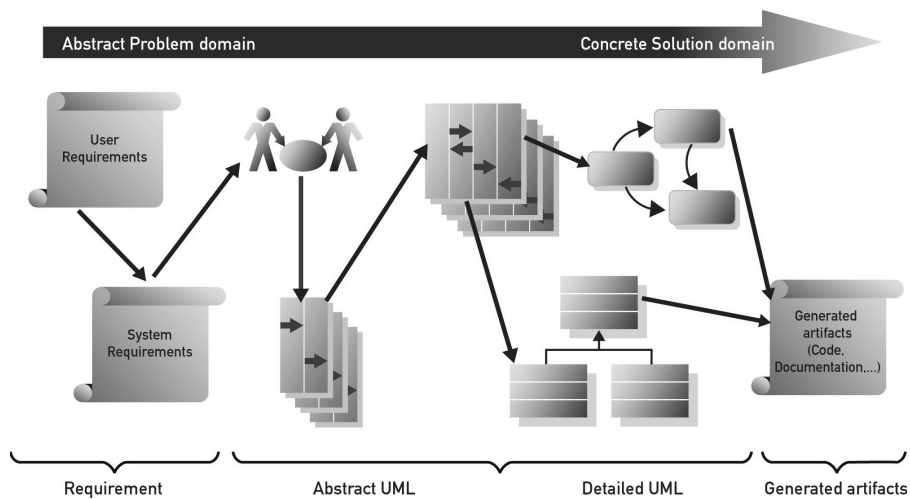
Telelogic DOORS and Telelogic Tau/Architect Will Help the Industry to:

- **Understand** requirements through visual models in UML 2.0
- **Validate** models through dynamic verification
- **Improve** consistency, completeness, and correctness of requirements
- **Trace** requirements throughout the lifecycle
- **Document** systems with correlated requirements and models
- **Communicate** across systems and software disciplines using UML 2.0 as a common language

35

© Telelogic AB **Telelogic**

UML across the lifecycle



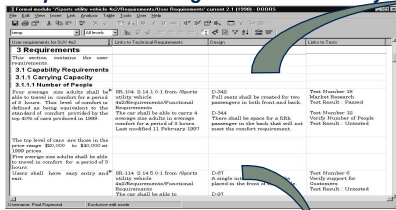
36

© Telelogic AB **Telelogic**

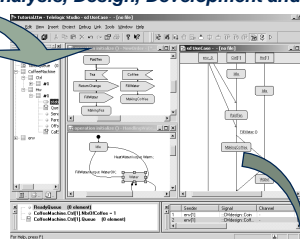
Requirements Driven Development:

Keep engineering teams focused!

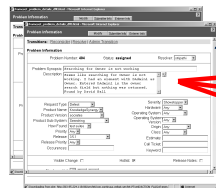
DOORS:
Requirements Management & Traceability



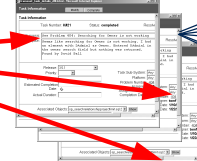
Tau/Architect, Tau/Developer Tau/Tester:
System Analysis, Design, Development and Test



ChangeSynergy:
Work Orders, ECP, RFD/W, NCR



CM Synergy:
Engineering Tasks, ECN



ActiveCM:
Controlled Code Modules



Generate traceability from requirements to design and code!

Need any Help?

Dr. Chris Sibbald

Chris.Sibbald@telelogic.com

613 788 3700

Telelogic Resource Center:

<http://www.telelogic.com/resources/>

Why does visualizing requirements with UML make sense?

- Visualize, transform and refine textual requirements
- Low learning curve
- Standard graphical notation with standard meaning
- Easy to understand and navigate
- Formal enough to force questions to be asked & answered
- Common language across systems and software
- Wide range of tool support
- Interchange between UML tools
- Re-use of trusted components
- Model execution and code generation (*with the right tools!*)

39

© Telelogic AB

Telelogic

Solving Customers' Most Complex Development Challenges

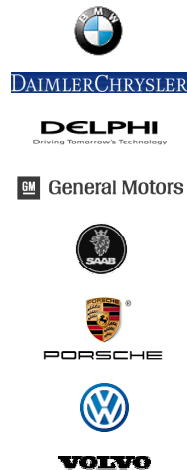
Communications



Military/Aerospace



Automotive



Finance, IT and more

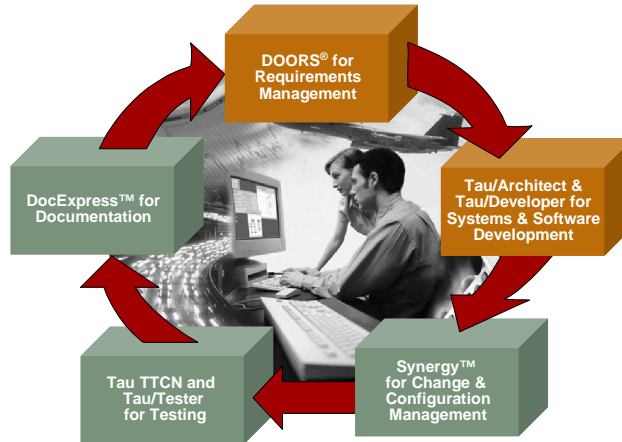


40

© Telelogic AB

Telelogic

Telelogic Solutions Span the Systems and Software Development Lifecycle



Our vision is to be the leading supplier of solutions for companies and engineers developing advanced systems and software

41

© Telelogic AB **Telelogic**

Telelogic

Back to the Drawing Board: Visualizing Requirements Using Graphical Models

Chris Sibbald
Director, Application Engineering eastern Region

Telelogic