



**UML 2.0 Specification of a QoS-oriented
Transport Protocol (QoSTP)**

Ernesto EXPOSITO



Introduction to UML 2.0

UML introduction

- Unified Modeling Language.
- June 12, 2003: UML 2.0 standard officially adopted by the Object Management Group (OMG)
- Industry standard mechanisms for visualizing, specifying, constructing and documenting software systems.
 - Documentation (ITU/ETSI/etc.)
- Intended to:
 - Modelers: to understand the problem
 - Designers: to explore possible solutions
 - Developers: to construct solutions
- Model-Driven Architecture (MDA):
 - modeler-> designer-> developer
 - Platform Independent Model (PIM) -> Platform Specific Model (PSM)
- UML tools
 - Specification (model checking) and visualization
 - Simulation and testing (model verifying)
- UML 2.0: 13 diagrams (structure and behavior)

Structure

- Structure:
 - Class:
 - real word entities and their relationships
 - Architecture (or composite architecture):
 - to show how something is made (component-based design)
 - how parts of a (container) class are connected to each other to form an internal structure of the container
 - Component:
 - Show the structure of the system as black boxes with their interfaces (replacement or reuse)
 - Deployment:
 - Run-time architecture of the system, hardware platforms, software artifacts, software environments (OS/VM)
 - Object:
 - illustrative examples of objects and their links
 - Package:
 - to organize model elements and dependencies

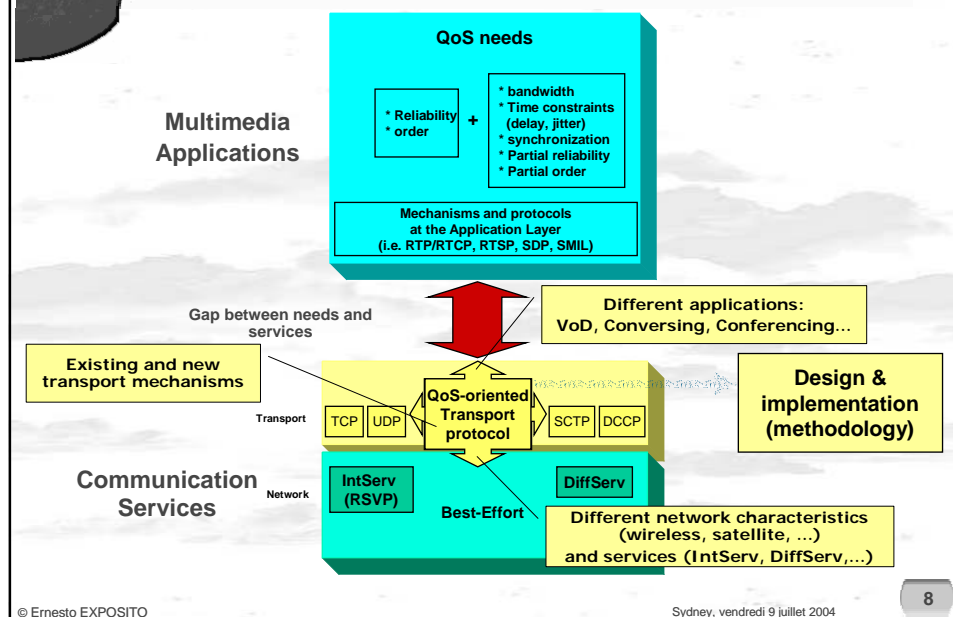
- Behavior:
 - Use case:
 - services that actors can request from a system
 - Sequence diagram:
 - ordered exchange of messages between a group of objects
 - State machine (or protocol state machine):
 - life cycle of an object
 - Activity:
 - modeling of concurrent control and data flow (Petri-nets like semantic)
 - Interaction overview:
 - show many different interaction scenarios for the same collaboration (high-level view of interactions)
 - Communication:
 - Similar to a sequence diagram (old collaboration diagrams)
 - Timing:
 - States of an UML element (such as state machine) in function of time

A case study : Transport protocols

Outline

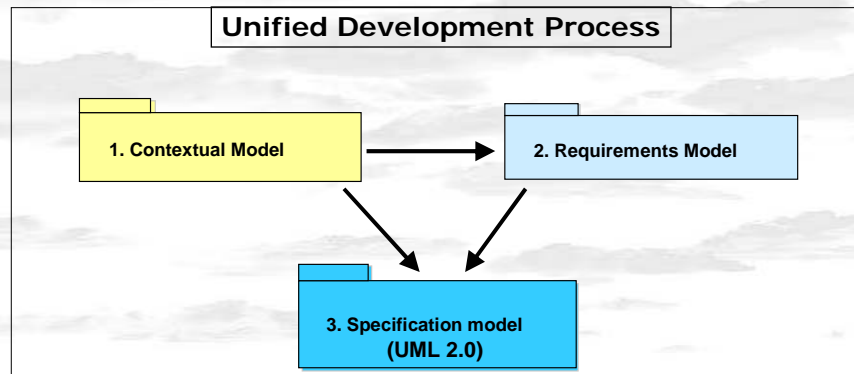
- Motivation
- Objective
- QoS design
 1. Contextual model
 2. Requirements model
 3. Specification model (UML 2.0)
- Conclusions and perspectives

Motivation: why end-to-end QoS?

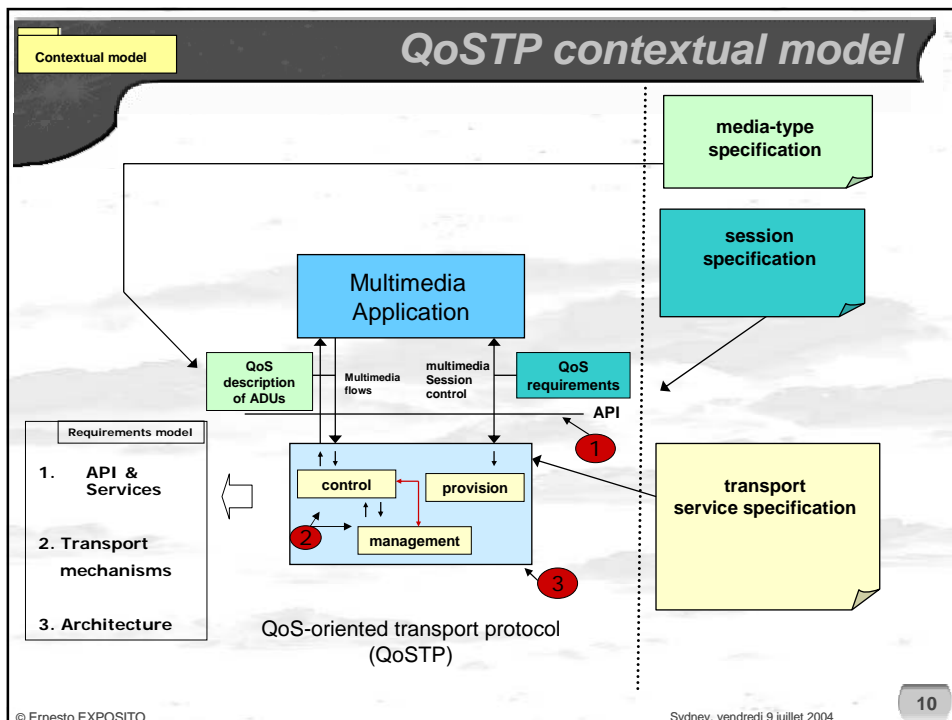


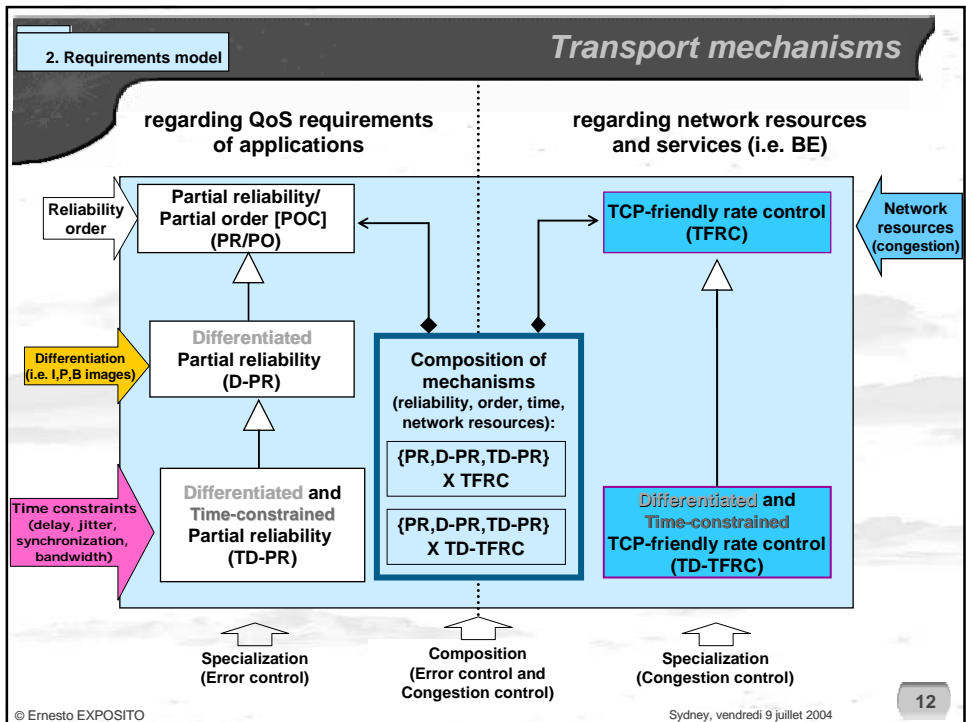
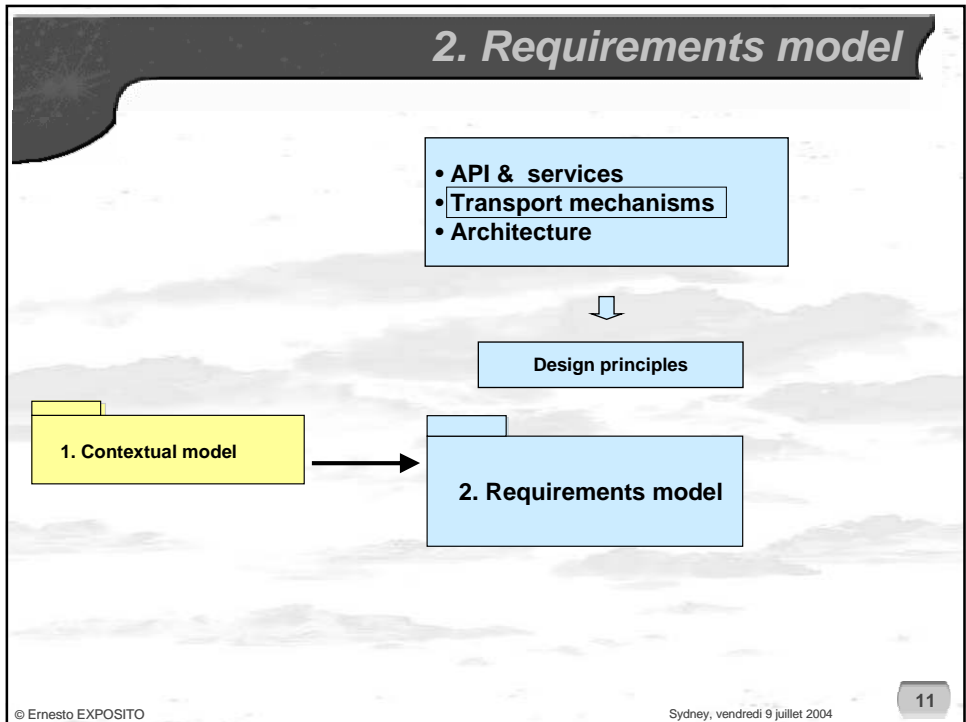
Objective

- Definition of an UML 2.0 based methodology aimed at designing and implementing a QoS-oriented transport protocol

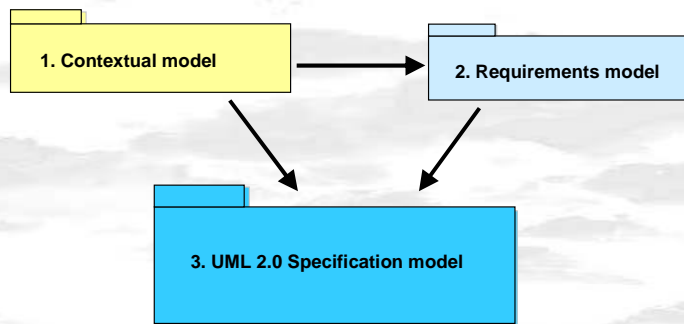


QoSTP contextual model



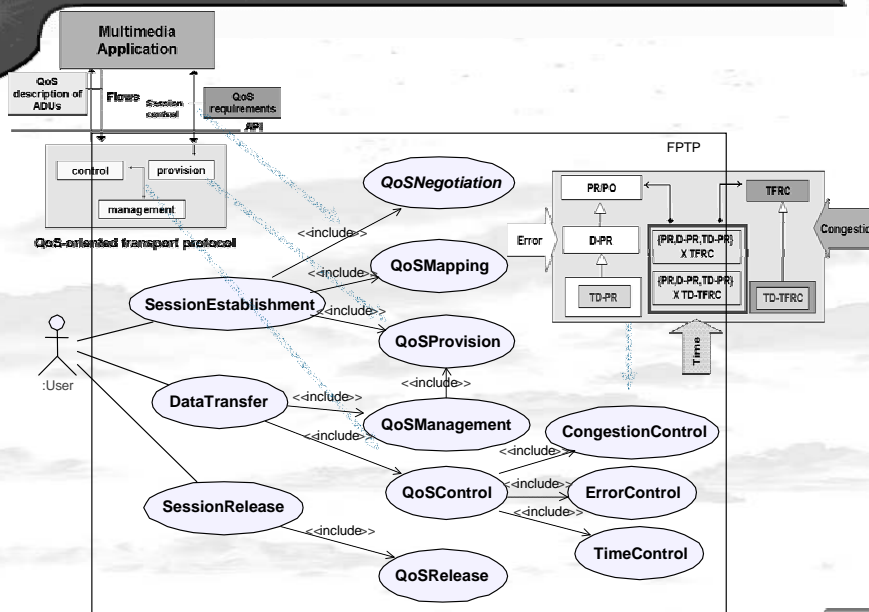


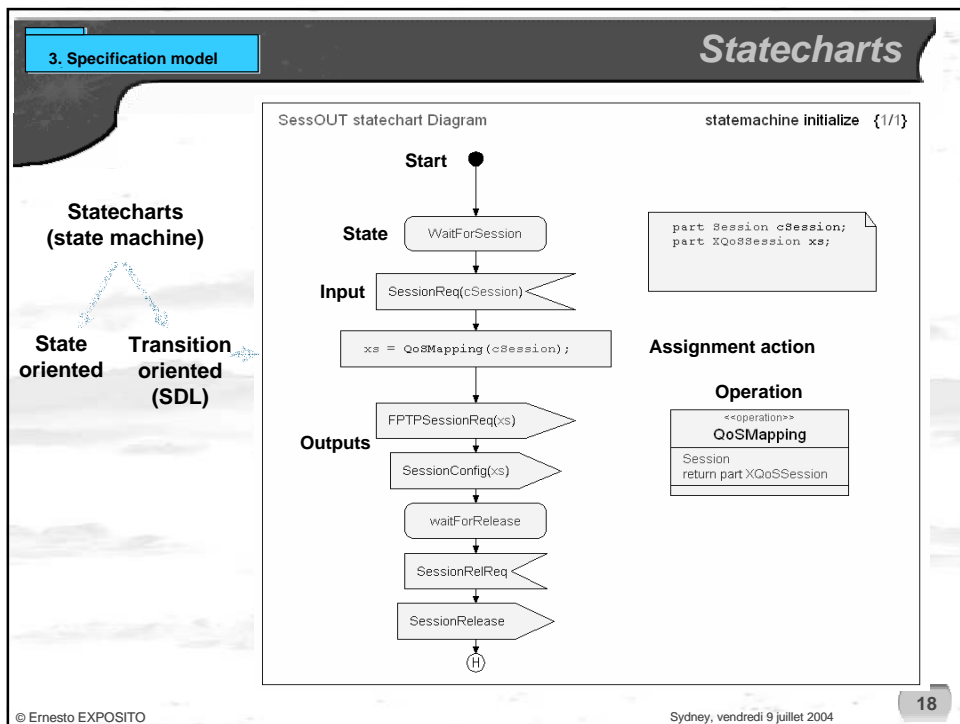
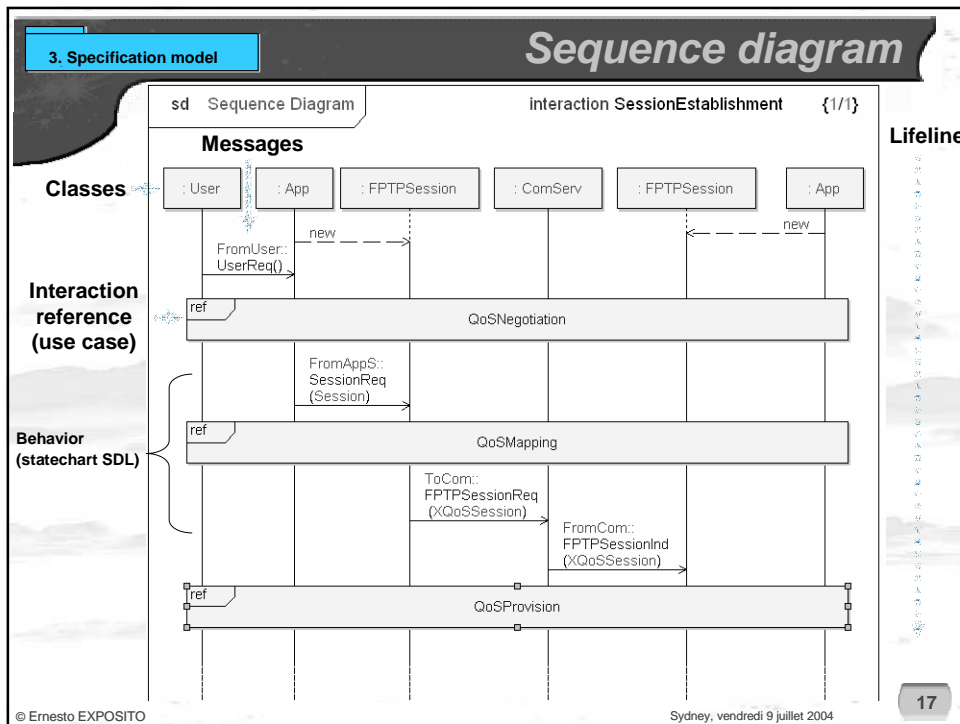
3. Specification model

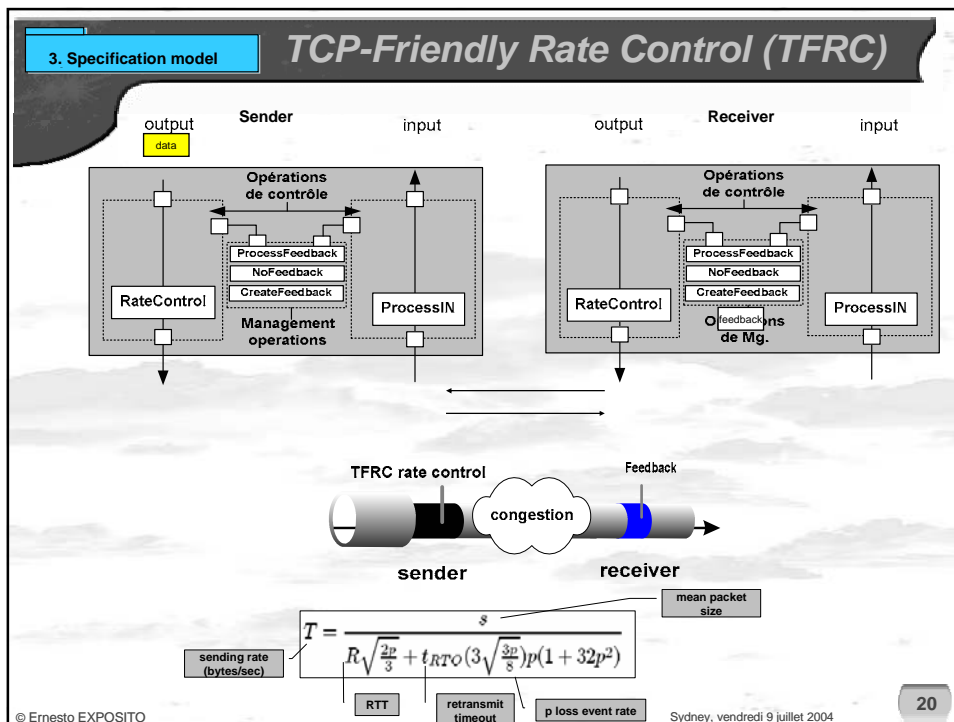
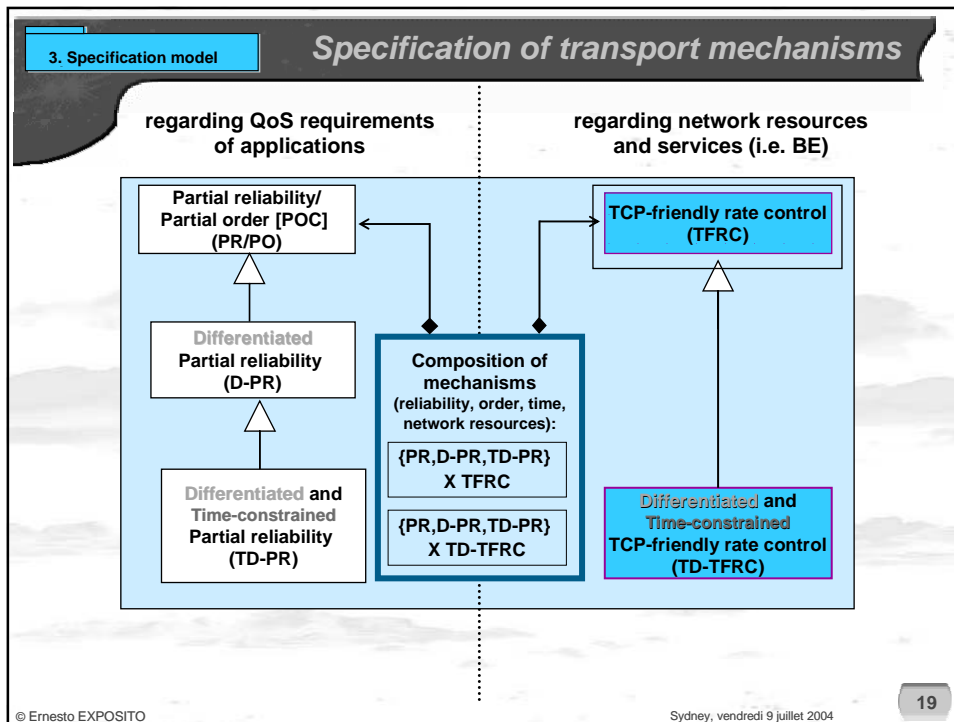


3. Specification model

Use Case





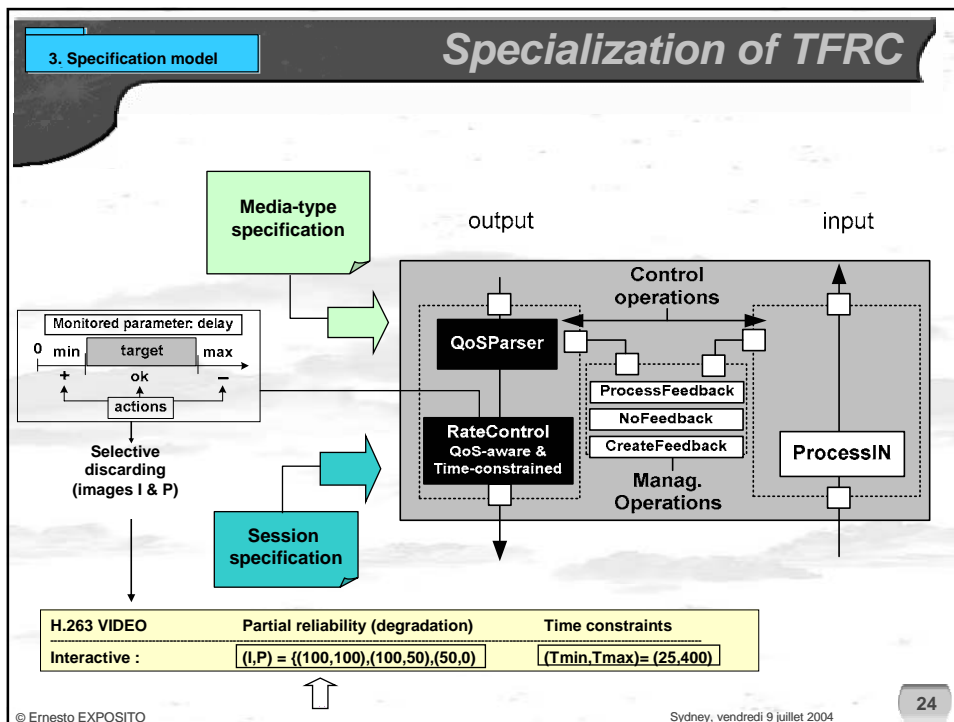
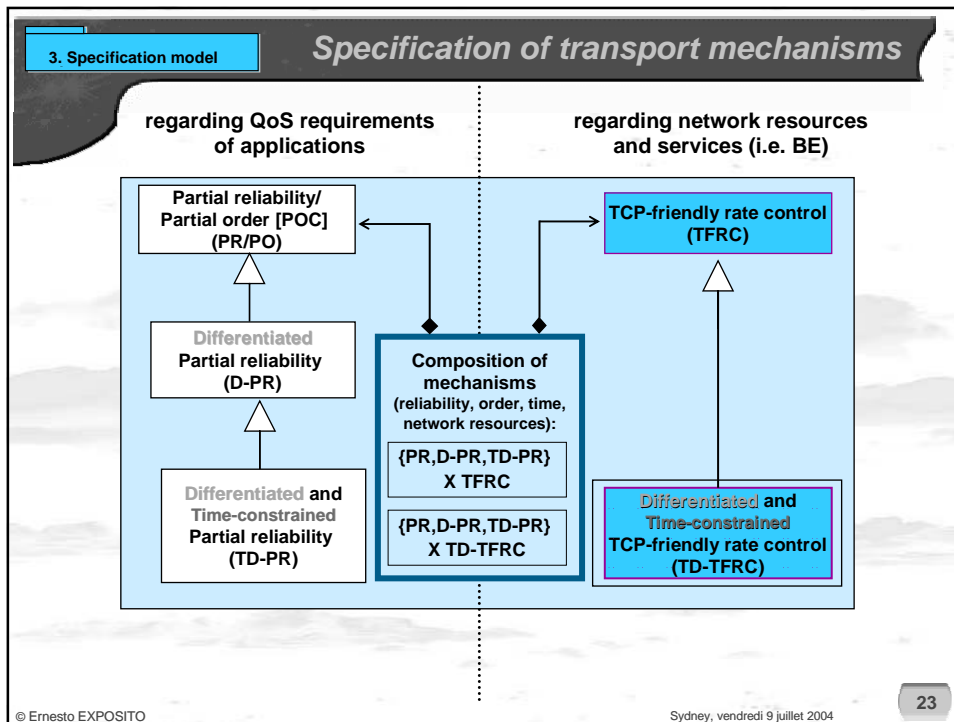


Simulation (TAU G2) TFRC

- Application:
 - H.263 video stream: I/P pictures (≈ 4 ADUs) every 62.5 ms
- Network service
 - RTT ≈ 150 ms
 - PLR = 0
 - Bandwidth unlimited

TFRC Simulation (TAU G2)

- Model Verifier:
 - to verify the behavior of the UML model.
 - generates a C code executable program
 - simulating using various commands and breakpoints, monitoring values, etc.
 - Interactive or random/exhaustive
- Focus on
 - internal behavior of the model
 - signaling to and from the environment
- Traces:
 - sequence diagrams or logs (observers)

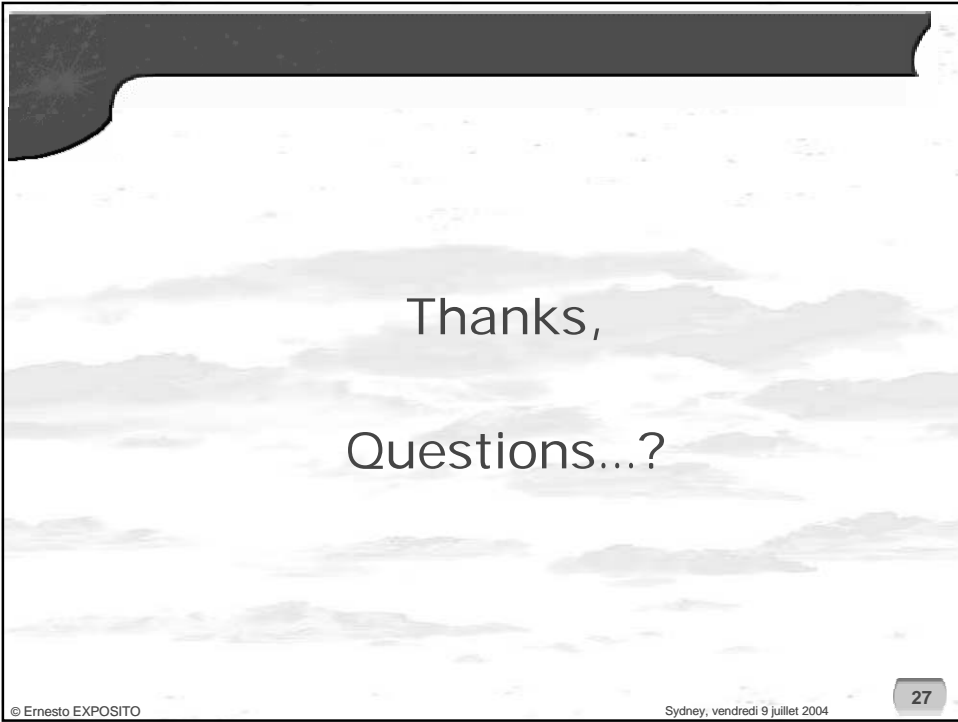


Simulation results

TFRC		TD-TFRC		
Rec App	Rec disc	Rec App	Source discard	Rec discard
20	48	36	32	0
30%	70%	52%	48%	0%

Conclusions and perspectives

- Efficient UML 2.0 based methodology for specifying, simulating, implementing and evaluating a QoSTP
 - In particular
 - Component-based design
 - Specialization
 - Composition
 - Simulation (interactive – non exhaustive)
 - Reuse models (scenarios, components, etc.)
 - To explore
 - Validation: exhaustive simulation?
 - Code generation?
- Perspectives
 - XQoS System:
 - XQoS framework for applications, end-systems, protocols, services, etc.
 - Methodology



Thanks,
Questions...?