

A UML profile for Distributed Components

E. Madelaine, Antonio Cansado, Emil Salageanu

OASIS Team,

INRIA -- CNRS - I3S -- Univ. of Nice Sophia-Antipolis

OSCAR meeting, Valparaiso, 18-19 dec 2006



Agenda

- **Motivations**
 - **Vercors Platform**
 - **Specification Languages**
- **UML models for distributed components**
 - **TTool**
 - **UML 2.0 components**
 - **CTTool**
 - **Towards a profile for the Grid Component Model**
- **Perspectives**



Distributed Software Components

Definition :

Software modules, composable, with well-defined interfaces, and well-defined black box behaviour

Our interests :

1. **Encapsulation:** Black boxes, offered and required services
2. **Composition**
Design of complex systems, hierarchical organization into sub-systems
3. **Separate administration**
Architecture Description Language (ADL), Non-functional interfaces
4. **Distribution (e.g. Computational Grid)**
Interaction at interfaces through asynchronous method calls

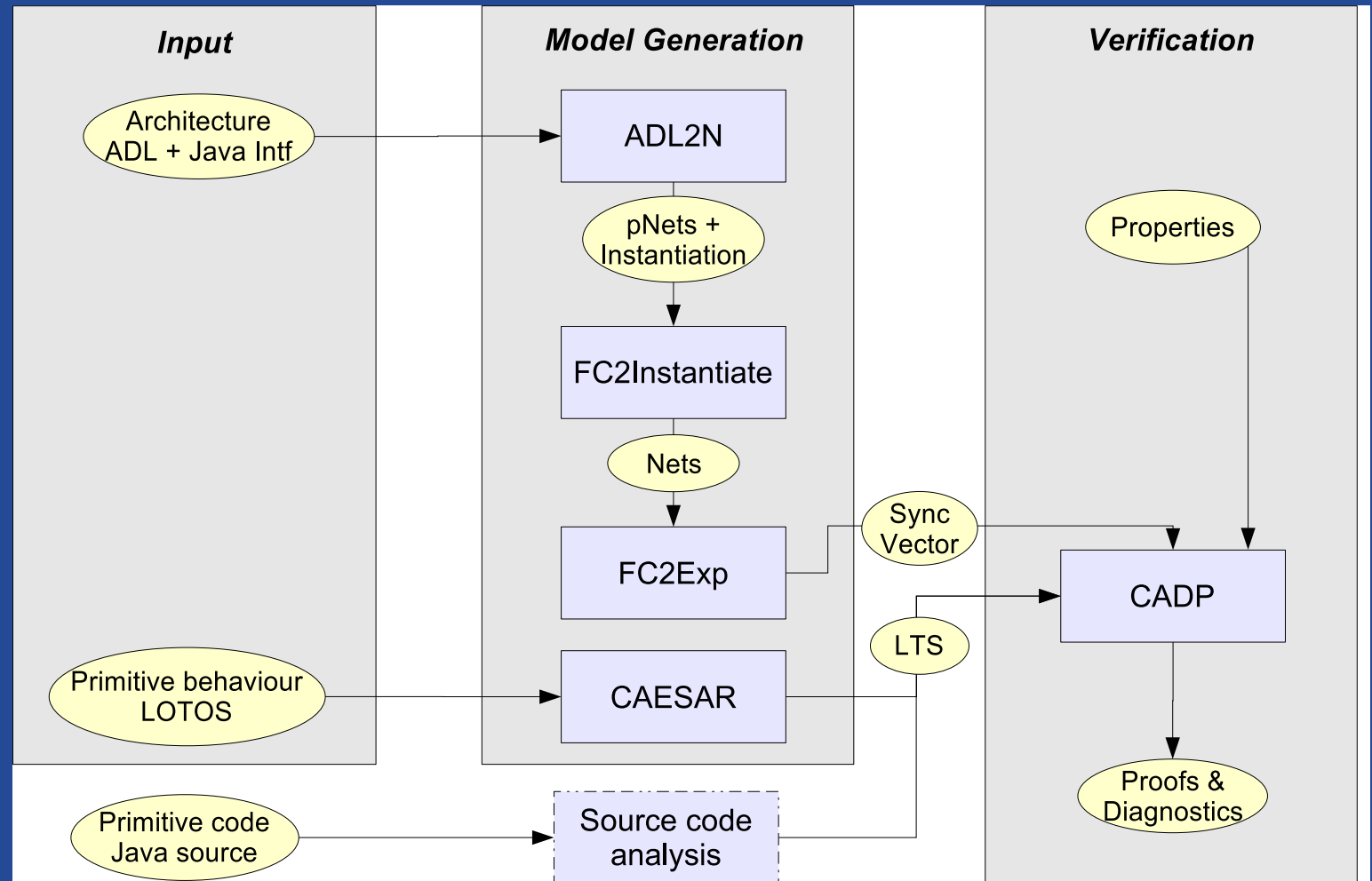
Distributed Applications :

ProActive implementation of the Fractal Model,
Grid Component Model (GCM)



The Vercors Platform

Eric Madelaine
Tomas Barros
Christophe Massols
Marcela Rivera
Antonio Cansado
Emil Salageanu
Hejer Rejeb
...



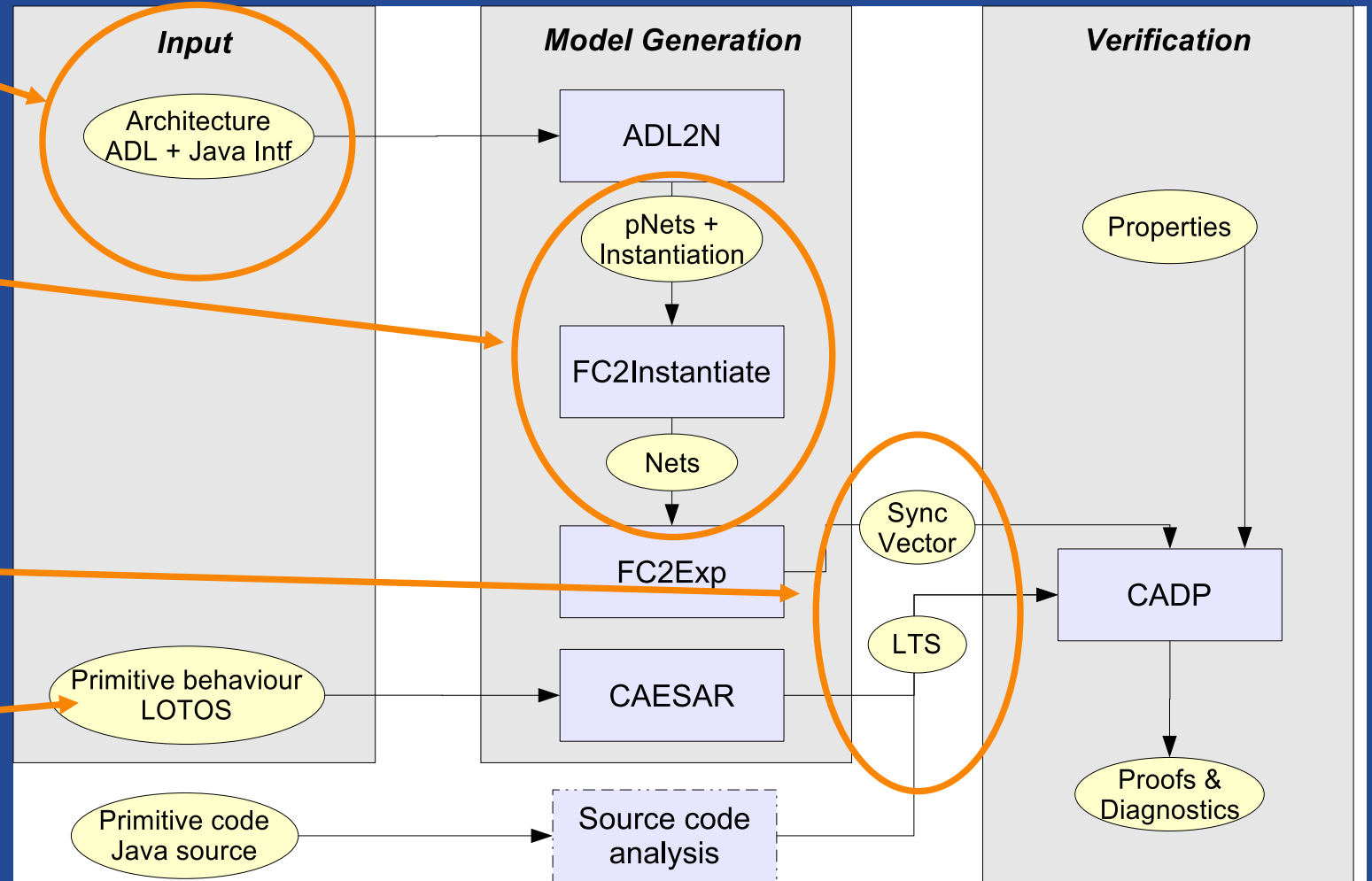
The Vercors Platform : Formats

1: Architecture & Interfaces Descriptions

2: Parameterized Networks of LTSS

3 : Networks of Finite LTSS

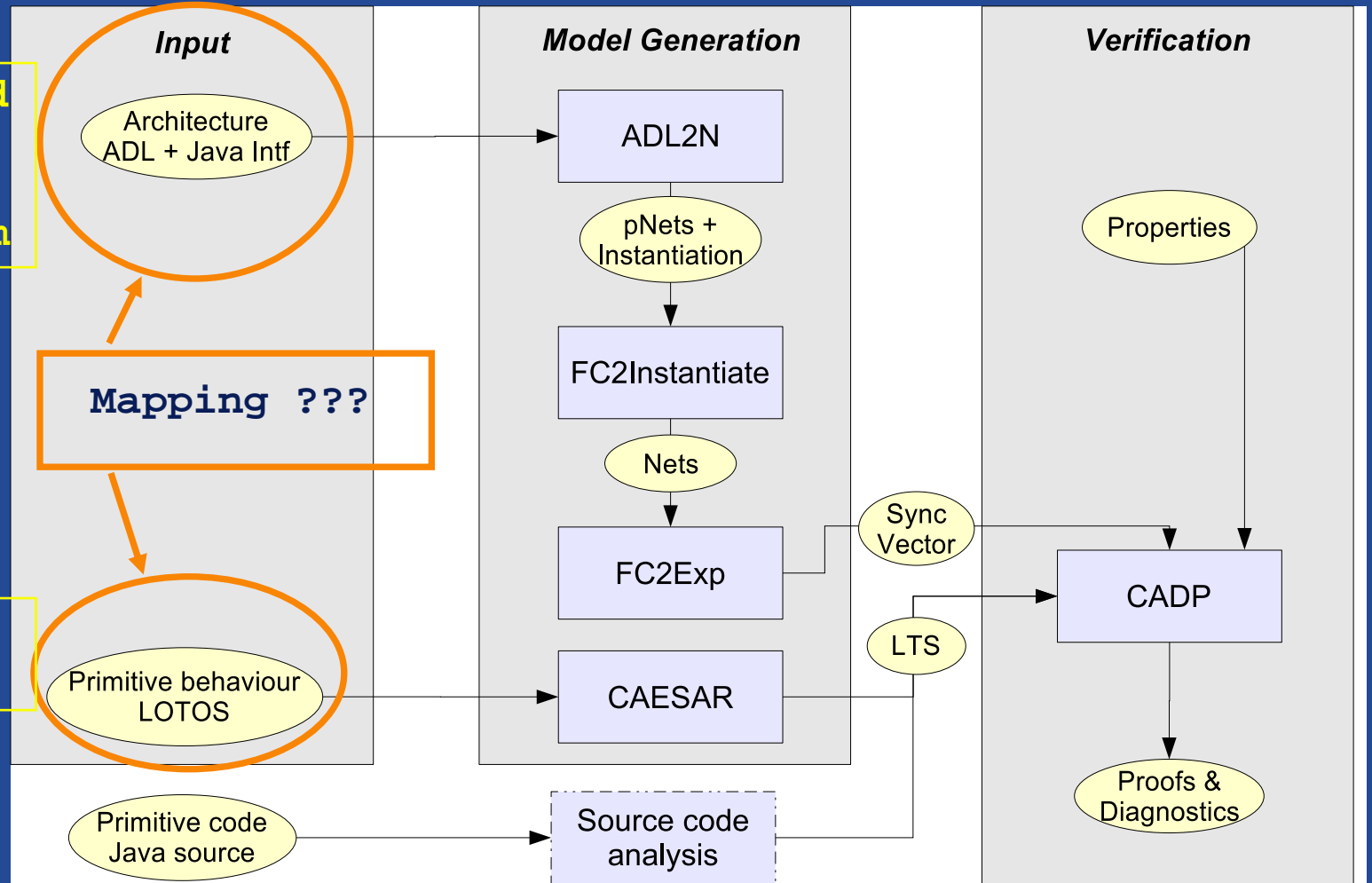
4 : (full) Lotos Processes



The Vercors Platform

Arguments of method calls and return values, in the interface formalism

Values in Lotos data types



Need for an Integrated Specification Language

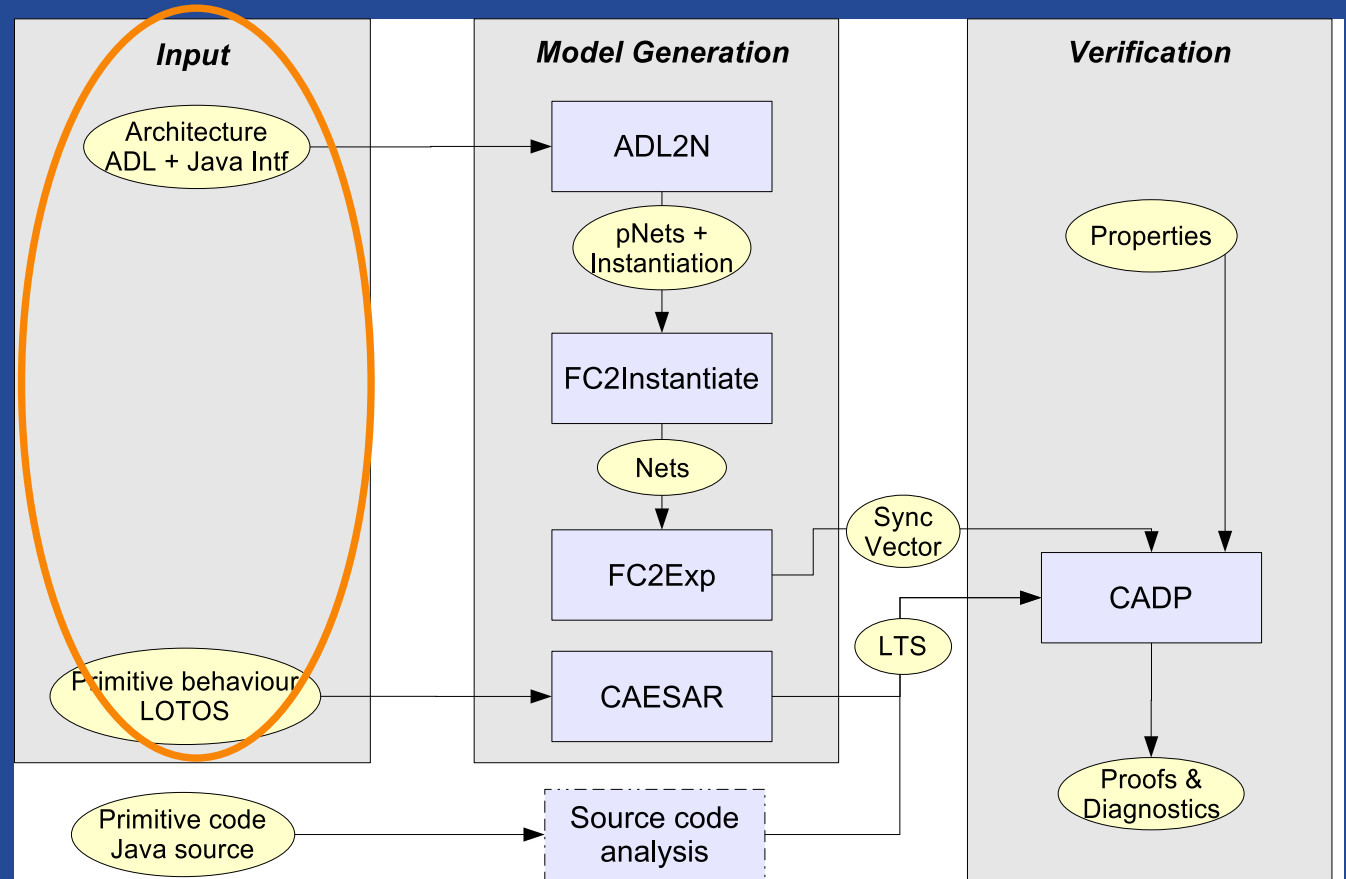
Examples:

- Process Algebras
- Java/A
- ArchWare Π -ADL

⇒ Work of Antonio Cansado...

Including distributed component concepts:

- Request queues, futures, proxies, collective interfaces, non-functional controllers, etc.



Specification Language

- Expressiveness, Completeness,
- Strong semantics, Calculus-level proofs...
-

Next Step: Language for the non-expert user

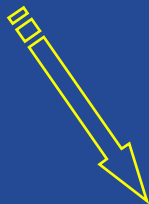
- Do not invent “yet another graphical specification language”
- Required capacities: hierarchical component architecture, signatures of interfaces, FSM behaviour specifications
- Extensibility for higher level abstractions.
- Semantics in term of the specification language.

=> UML diagrams



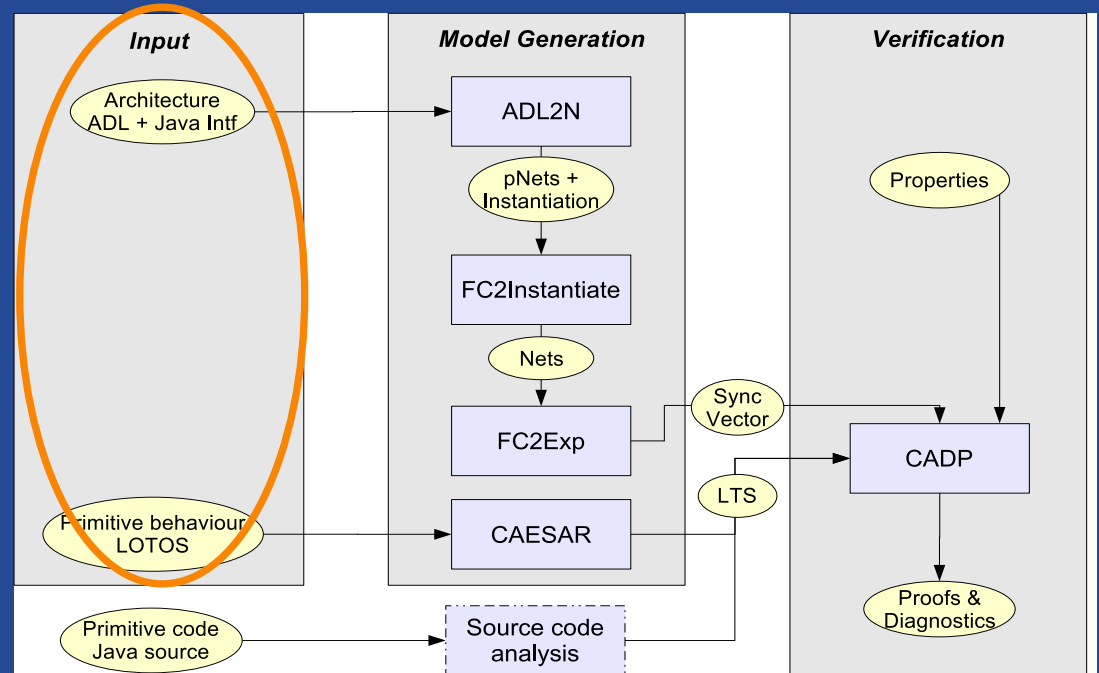
Next Step: Language for the non-expert user

UML
Architecture
+
behaviours



ADL
&
Proactive
code

"correct by
construction"



Agenda

- **Motivations**
 - **Vercors Platform**
 - **Specification Languages**
- **UML models for distributed components**
 - **TTool**
 - **UML 2.0 components**
 - **CTTool**
 - **Towards a profile for the Grid Component Model**
- **Perspectives**

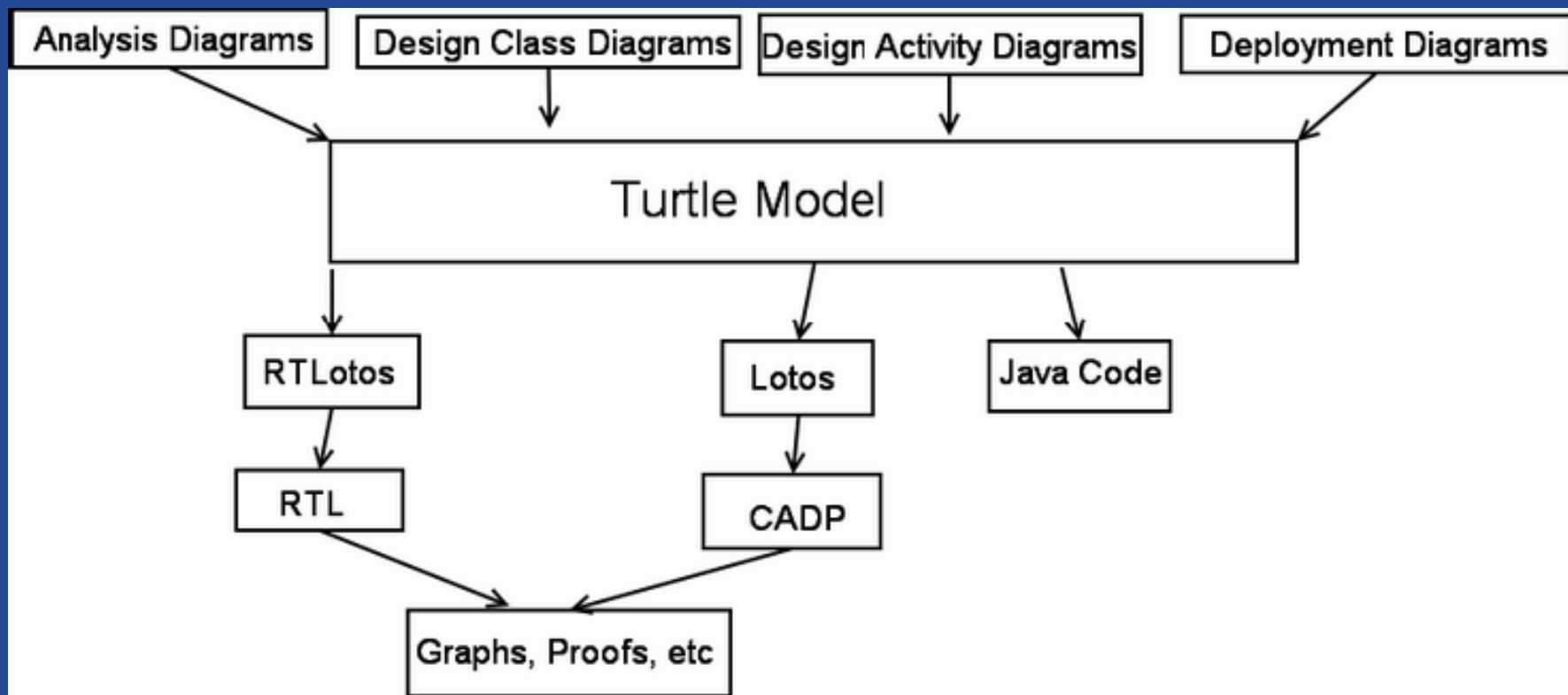


Starting Point: TTool

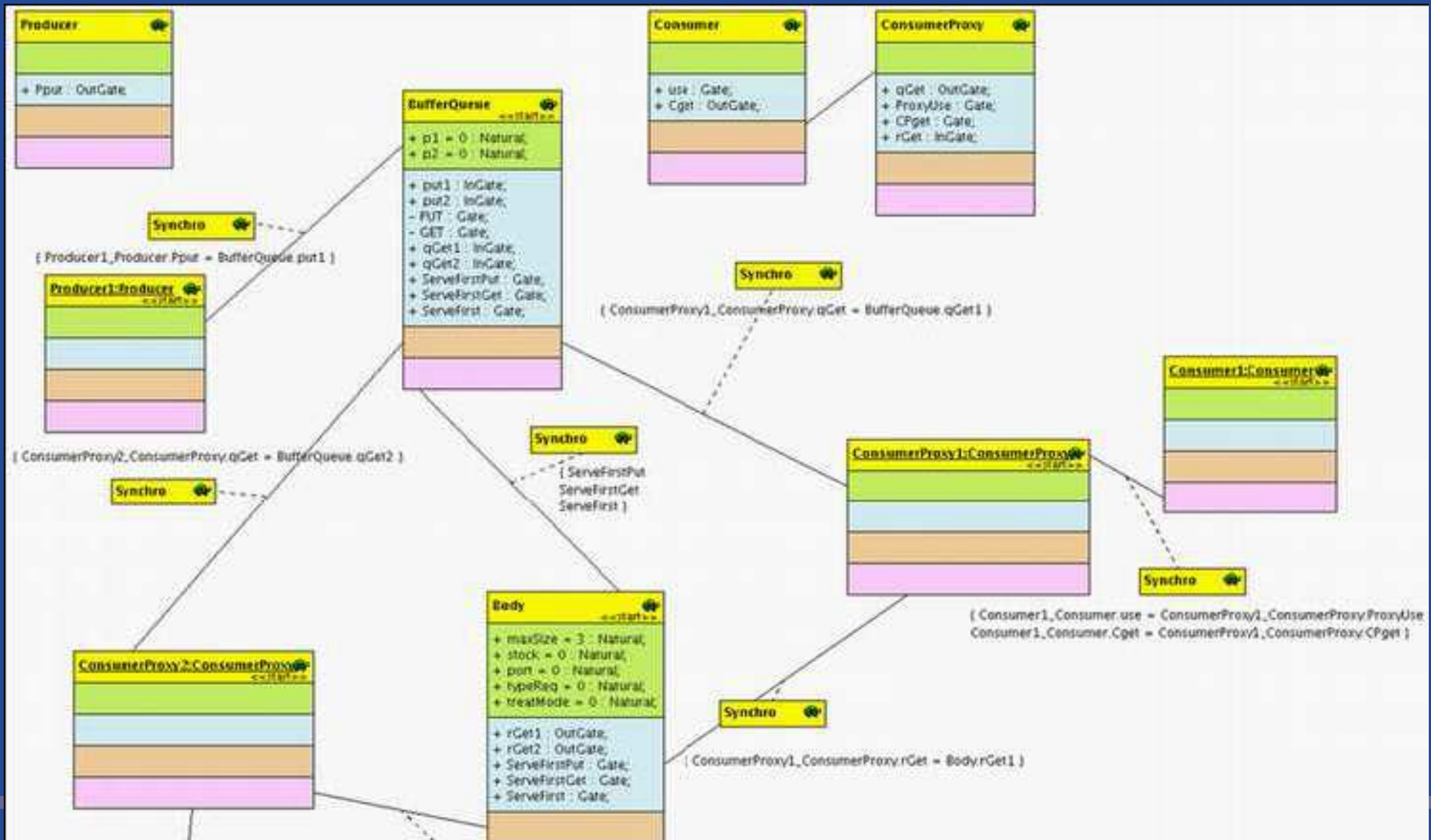
- Labsoc (System On Chip), ENST Sophia, Ludovic Apvrile.
- UML 1.5 diagrams:
 - Class diagrams + Interaction diagrams / Sequence Charts + Real Time
- Turtle Model:
 - Translation to RT-Lotos, Real-time Model-checking
 - Generation of Java code (simulator)



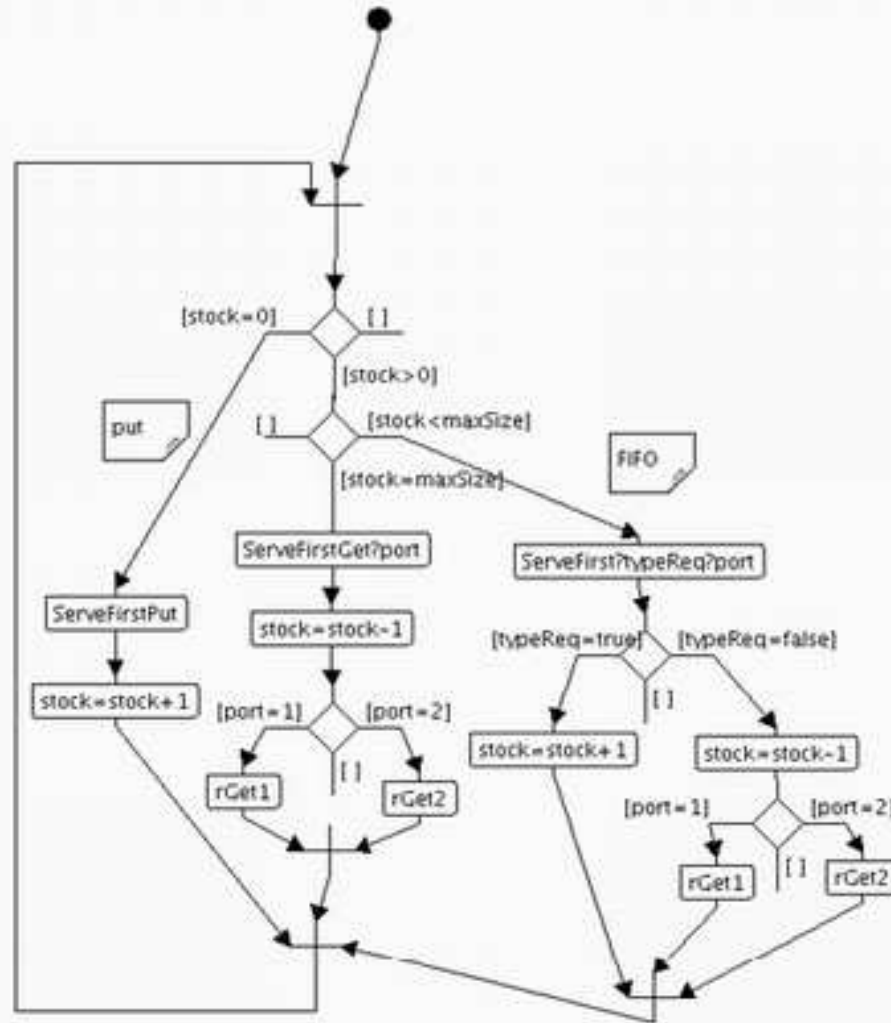
Starting Point: TTool



Class diagrams



Activity diagrams



UML 2.0 : new standard diagrams for components

[V. Mencil, M. Polak, Fractal'06
UML 2.0 Components and Fractal: An Analysis]

Component: now “as we know it”

- hierarchy / nested components
- *provided* and *required* interfaces
- Key concepts:
 - StructuredClassifier
 - functionality decomposed into *parts*
 - EncapsulatedClassifier
 - communication through Ports
 - Port
 - has provided and required Interfaces
 - has multiplicity (=> collection interfaces)
 - Component
 - combines these features, + inherits from Class (attributes)



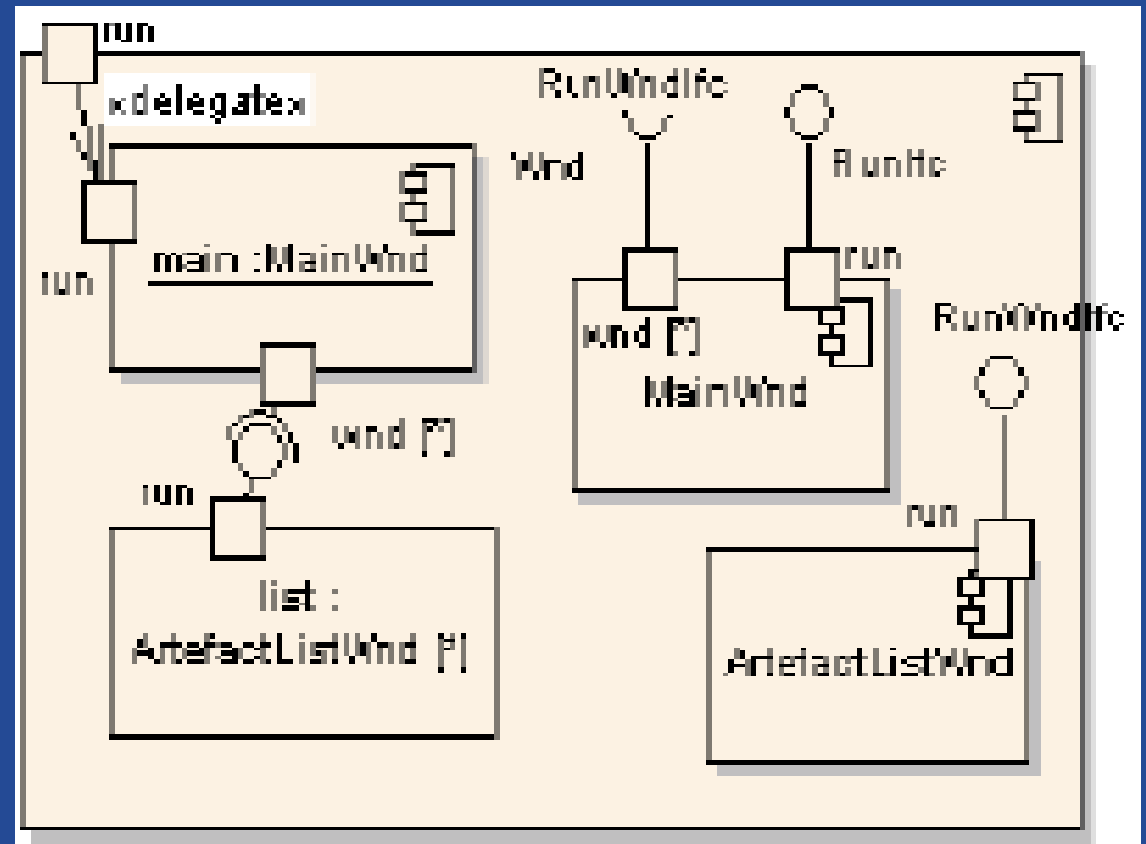
UML 2.0 [continued]

1. As a StructuredClassifier

- may own parts
with type, multiplicity

2. As a NameSpace

- may own components, classes, instanceSpecifications, interfaces
- But you must explicitly instantiate components.



UML 2.0 [continued] : technical details

1. Connectors

- Cannot be linked to interfaces (only to ports)
- Cannot be linked to InstanceSpecifications

2. Interfaces via Ports:

- Only one interface per port.
- Position of interface client/server.
- Port multiplicity determines cardinality+contingency.

3. Subcomponents implemented as Component + InstanceSpecification



UML 2.0 [continued]

1. Full mapping Fractal/Sofa -> UML 2.0

by M. Polak (2006)

1. Prototype Implementation, using Enterprise Architect



Prototype : CTTTool (with Emil Salageanu)

- UML 2.0 diagrams for Composite Structures + State Machines
- Editor + verification environment using TTool code base
 - => generation of Lotos code
 - => bridges to CADP toolset
- No deviance from UML 2.0
- No asynchronous components yet (common base for Oasis / Labsoc)
 - ⇒ Prototype available: www-sop.inria.fr/oasis/Vercors

Demo...

Next Step : ProActive TTool ??

1) Simplify the graphical constructions:

- Interface / Ports
- Delegate interfaces
- Libraries for components, interface descriptions

2) Higher level constructions

- Asynchronous components (future proxies, queues)
- Multicast and Gathercast interfaces



Next Step : ProActive TTool ??

Integration in the ProActive IDE ??

- Eclipse Plugin
- Fractal / GCM GUI
- Early warnings / semantic information :
 - deadlocks, active object dependencies, etc



Next Step : Graphics

Keep as much as possible existing styles...:

1. UML 2.0 composite structures
2. Fractal GUI (??)
3. MARTE
4. Simulink....



Muchas gracias

