



Critical

Dependable
Technologies
For Critical
Systems

Assessing the Formal Development of a Secure Partitioning Kernel with the B Method

Critical

Dependable
Technologies
For Critical
Systems

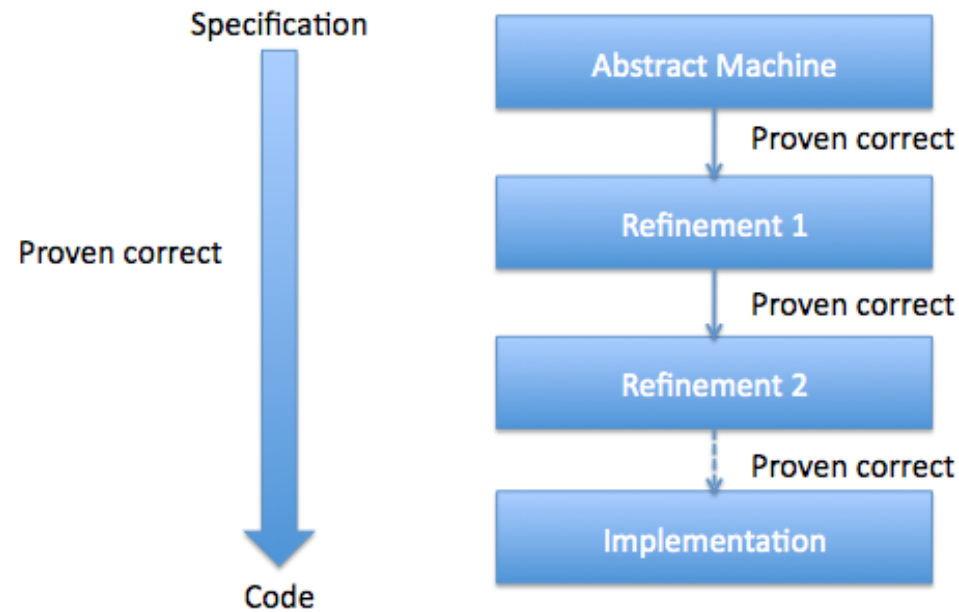
Outline

- The B Method
- Secure Partitioning Microkernel
- Developed Work
- Conclusions and Future Work

What is the B Method?

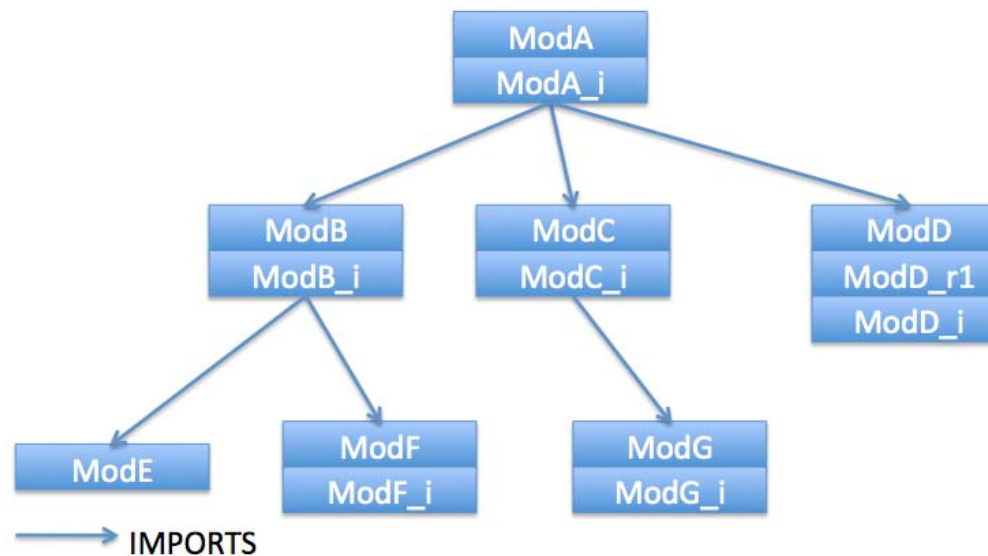
- *The B Method is a formal method for the development of specifications and their refinements down to an implementation [J.R.Abrial];*
 - Model-based approach similar to Z or VDM;
 - Addressed to functional requirements;
 - Imperative-like features;
 - Solid industrial applications (Line 14 – Paris Subway).

The Global View



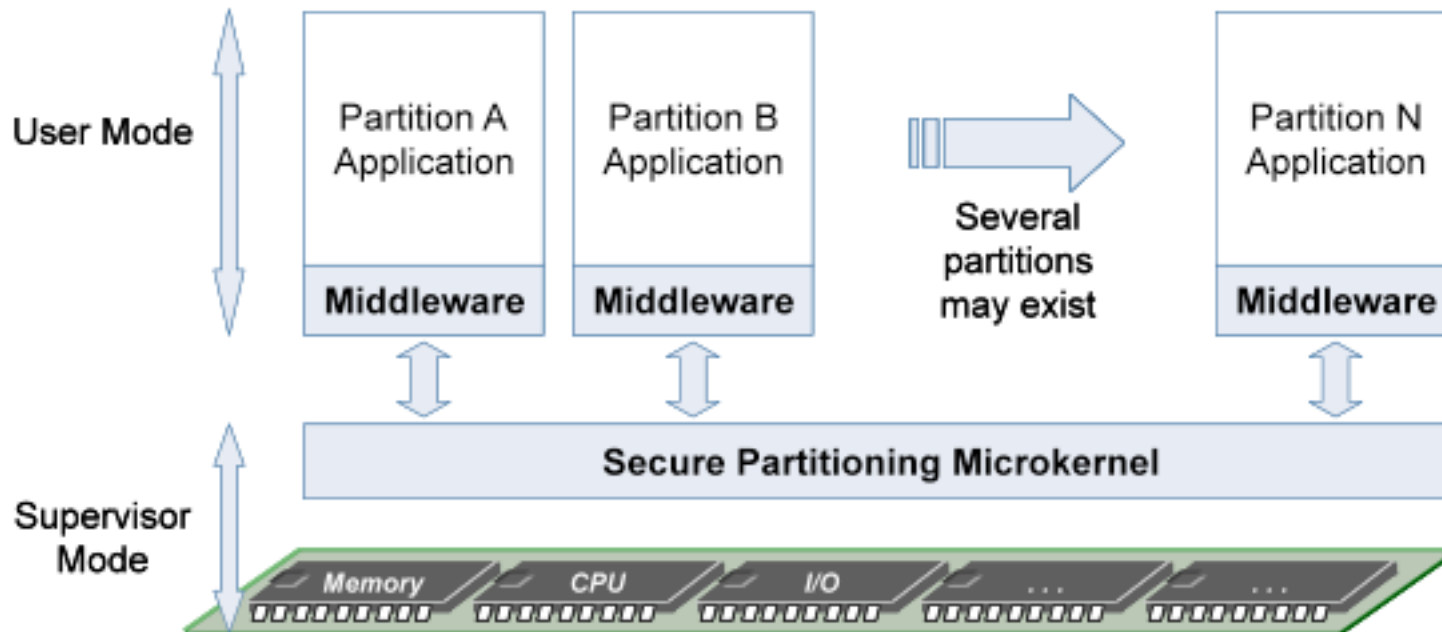
- Basically a B specification is composed by three types of components:
 - Abstract Machines;
 - Refinements;
 - Implementations.

Layered Approach



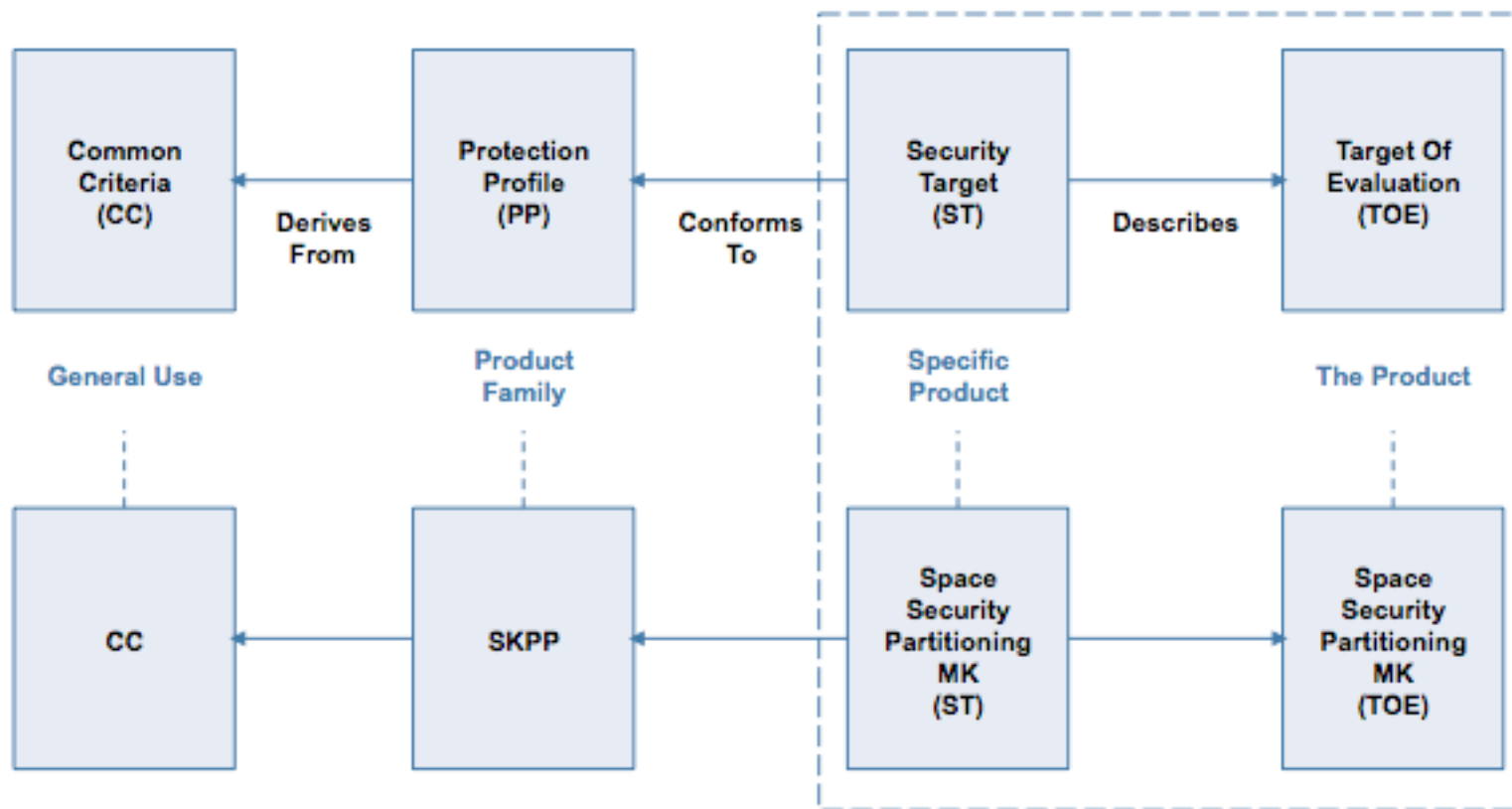
- A project is composed by several modules;
- The modules are made of B components;
- Decomposition in several levels;
- Distribution of requirements through different levels.

Secure Partitioning Microkernel Architecture



- Time Partitioning;
- Space Partitioning;
- Security Partitioning.

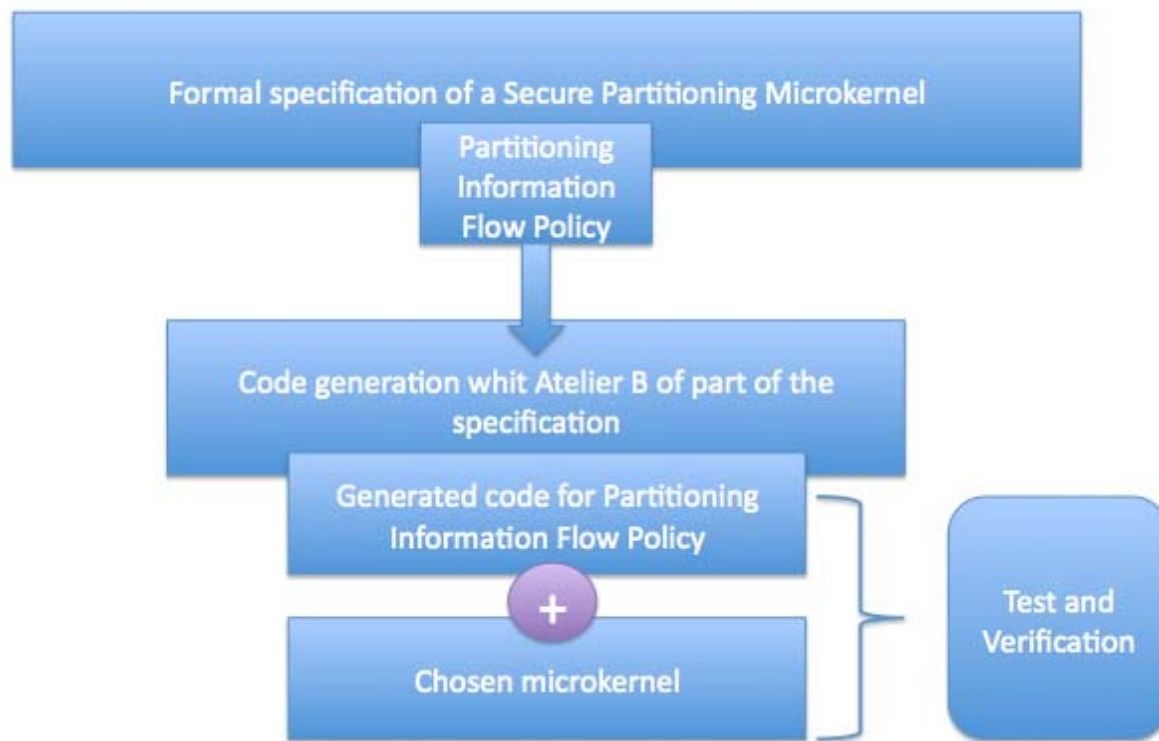
Secure Partitioning Kernel Profile



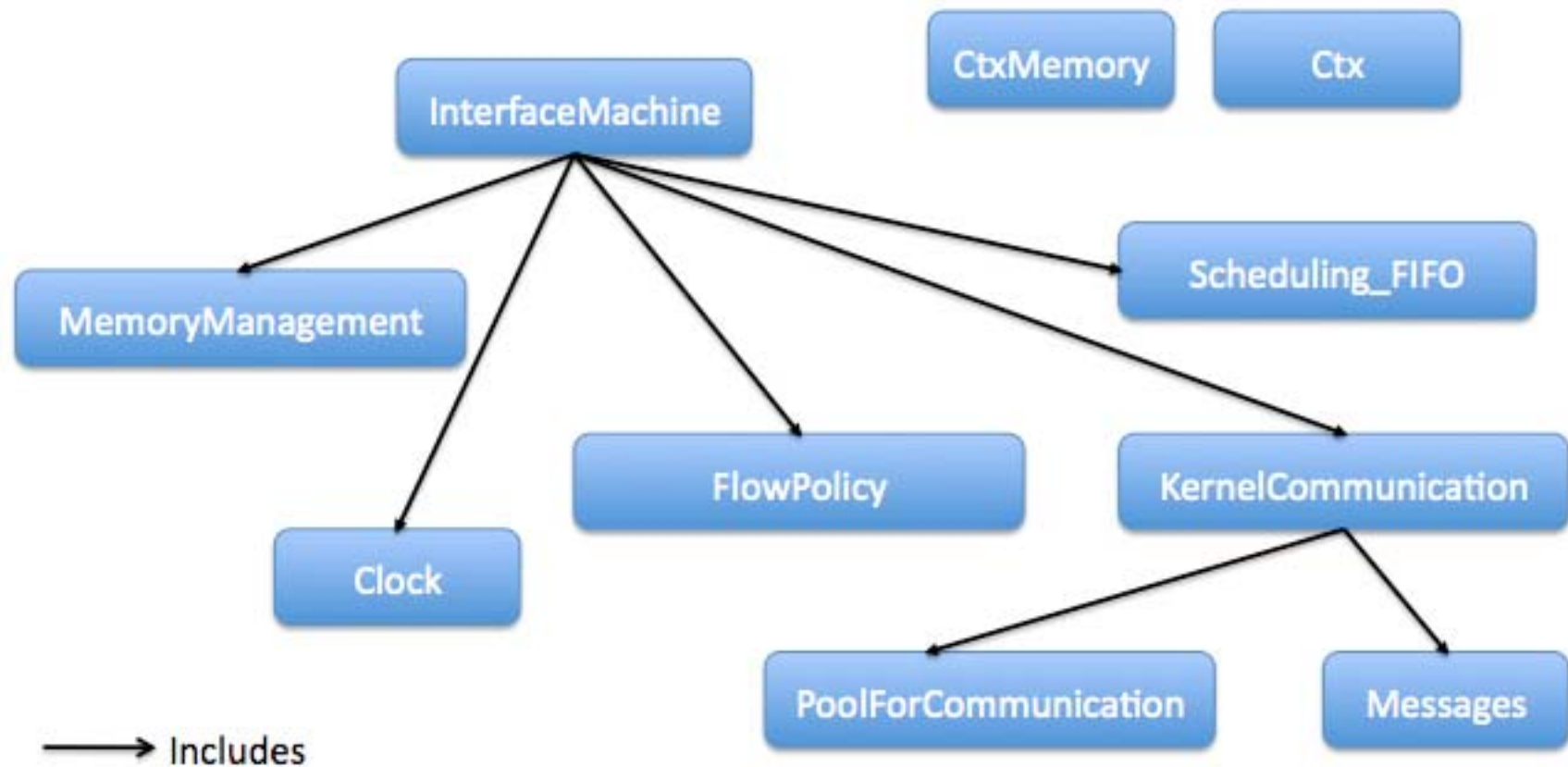
- US Government Profile for Separation Kernels in Environments Requiring High Robustness.

Proposed Work

- Abstract Model of the Secure Partitioning Microkernel;
- Refinement of Part of the Abstract Model.

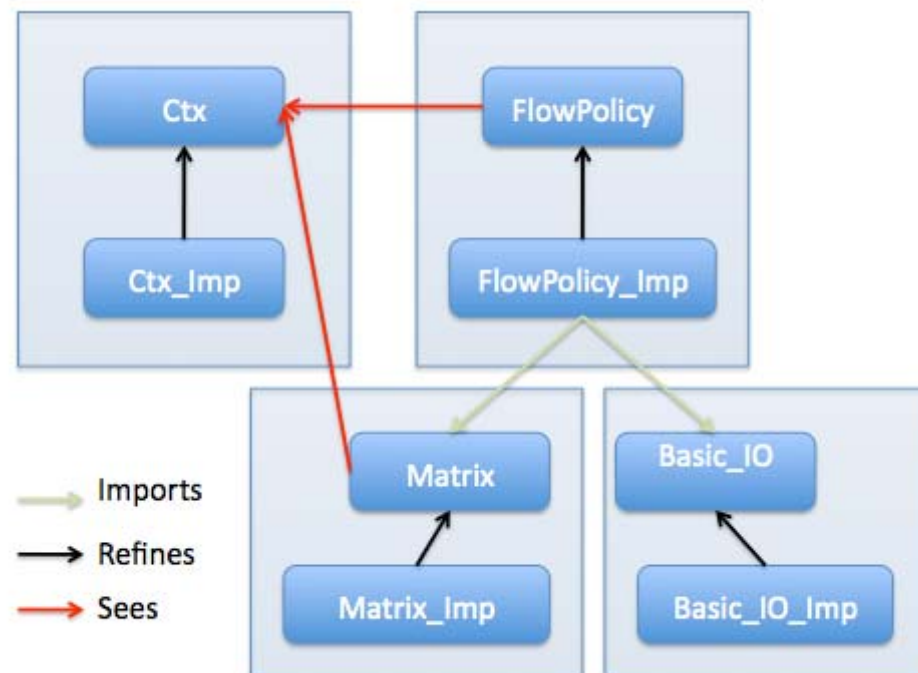


Abstract Model of the Secure Partitioning Microkernel



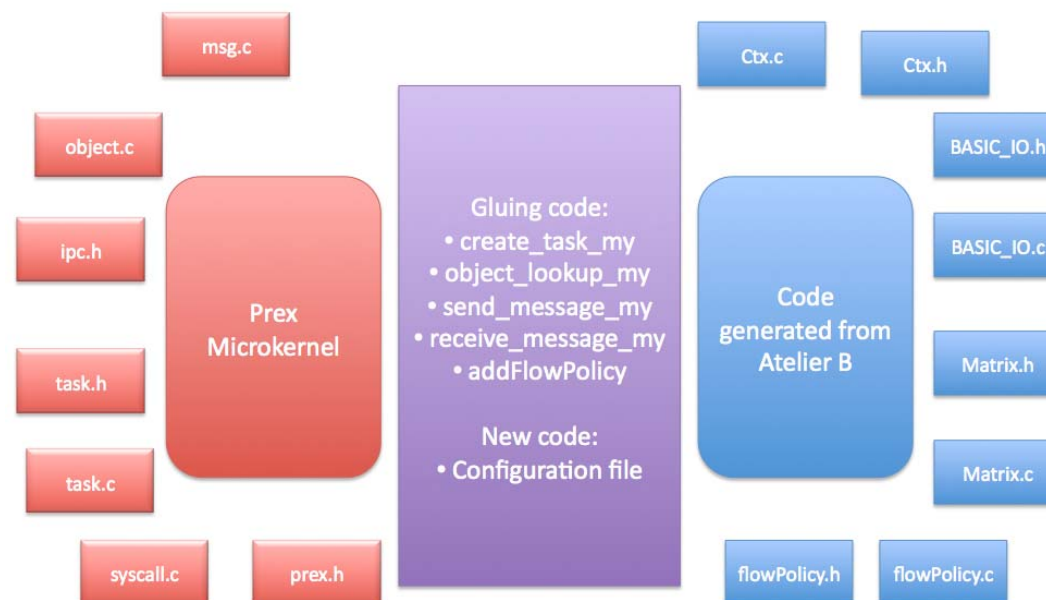
Partition Information Flow Policy

- The chosen policy was **Partition Abstraction** – all processes inside a partition have the same rights. The access mode for communication between partitions is also defined.



PIFP and Prex

- Prex provides the necessary characteristics to add a flow policy;
- Code generated from Atelier B completely transparent;
- The code generated was not changed.



Verification and Validation

- Verification and validation was achieved through the use of testing and formal proof;

Component	TC	POG	nPO	nUN	%Pr	B0C
BASIC_IO	OK	OK	0	0	-	OK
BASIC_IO_Imp	OK	OK	3	3	0	OK
Ctx	OK	OK	0	0	-	OK
Ctx_Imp	OK	OK	1	0	100	OK
FlowPolicy	OK	OK	4	0	100	OK
FlowPolicy_Imp	OK	OK	108	14	87	OK
Matrix	OK	OK	4	0	100	OK
Matrix_Imp	OK	OK	23	0	100	OK

Verification and Validation

- Some numbers about verification and validation:
 - Proof coverage rate is equal to 88%;
 - Non proved obligations are equal to 12%. These proof obligations were visually checked;
 - The remaining 12% are currently being proved interactively;
 - Animation and Model-Checking with ProB revealed no problem.

Tools

- Atelier B
 - Type checker;
 - Project manager;
 - Early validation through the use of formal proof;
 - Automatic generation of C code.
- ProB
 - Animation of the B Models;
 - Model Checking;
 - Permits integration with Atelier B.

Some Metrics

- 9 M. Months in total:
 - 3 first months for understanding of the problem and requirements capturing;
 - 4 months for high level specification;
 - 2 months for refinement and code generation of the PIFP;
- No significant previous experience with the technologies.

Industrial use

- Tools are sufficient mature to be used in industrial context;
- Reduction of time spent writing and validating the implementation code (no unit and integration tests);
- Interactive proof requires the highest proficiency;
- Reduction of erroneous behaviors at the end, and so, reduction of time fixing it;
- Increase confidence: animated formal models provided behaviour visualisation;
- Non-commercial version of Atelier B code generator presented some limitations.

Conclusions

- Capability of using formal methods in selected parts of the the project;
- Verification makes part of the development;
- Several formalisms should be used in the development;
- Proof of Concept: Formal Methods can be used in the industrial domain.

Future Work

- Complete development of the all microkernel;
- Complete the proof effort;
- Exploitation of an automatic and intensive testing process using the formal specification.

Contacts

- José Miguel Faria
 - jmfaria@criticalsoftware.com
- André Passos
 - ab-passos@criticalsoftware.com



Coimbra, Lisbon, Porto

www.criticalsoftware.com



San Jose

www.criticalsoftware.com



Southampton

www.critical-software.co.uk



Bucareste

www.criticalsoftware.ro



São Paulo

www.criticalsoftware.com.br