Code Generation Strategies for partitioned architectures

Laurent Pautet
Julien Delange
Fabrice Kordon
Issues in the implementation of safe and secure systems

- **Difficulties in writing specifications**
  - 70% of errors are introduced at design-time

- **Need specification review and code certification**
  - For safety and security requirements

- **Several approaches, no common standard**
  - ARINC653, MILS, OSEK/VDX, …
An “idealistic” Process development

1. Specifications
2. Verification
3. Code generation
4. Compilation
5. Generated code
6. Verification OK/error
7. Application code
8. Partitioned runtime
9. POK kernel ARINC653/MILS compliant
10. Binary
11. Ready to run! Overhead <= 500 KB!
12. POK Checker
13. Ocarina AADL toolsuite

AADL models
Code generation

AADL specifications

Analyze each partition
- Resources managements
- Fault management
- Needed runtime functionnalities
- Intra-partition communication

Analyze all partitions requirements
- Configuration of fault-containment across partitions
- Partitions’ resources management
- Configuration code for security services

Enforce safety inside partitions

Partition’s code

Enforce safety AND security across partitions

Kernel configuration code (includes safety and security enforcement)
Conclusion

- **Main goal: a complete development process**
  - Ensure secure and systematic process

- **Today’s focus: code generation**
  - Implementation conforms to the specifications
  - Appropriate code patterns for safety and security

- **To come next**
  - Secure runtime (space & time isolation)
  - Towards analysis of AADL specifications