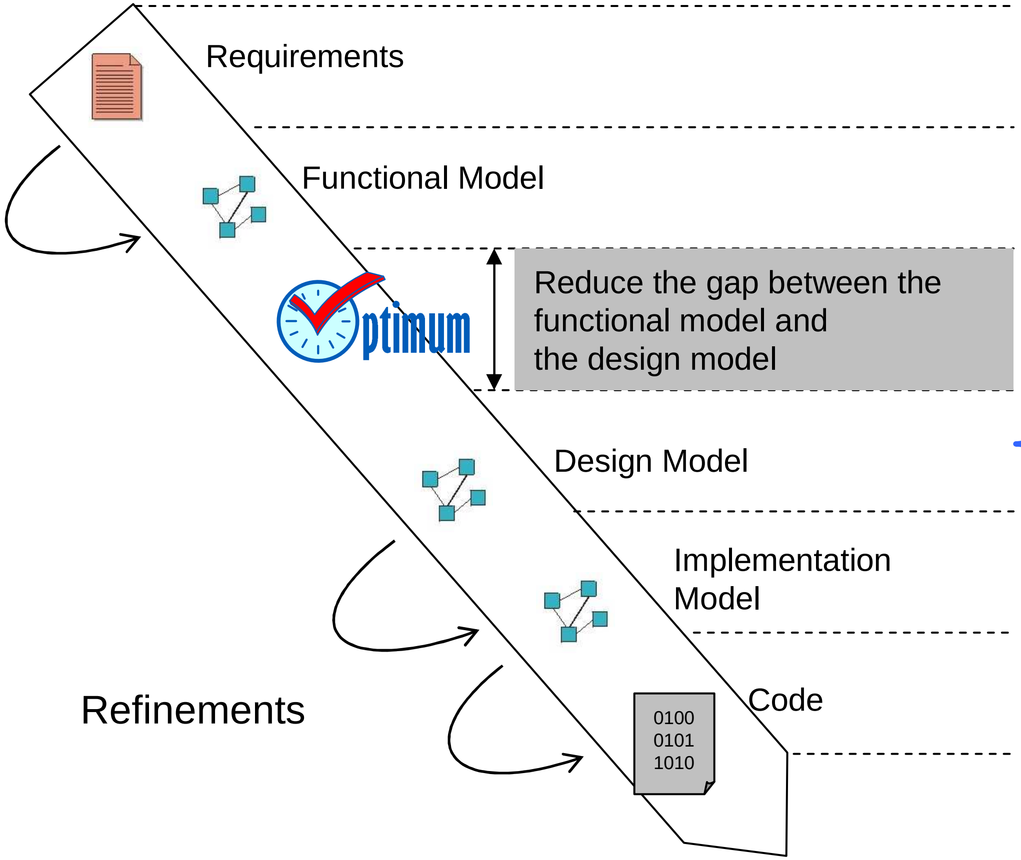
Optimum: a MARTE-based methodology for schedulability analysis at early design stages

Third International Workshop on UML and Formal Methods
November 16th, 2010, Shanghai, China

chokri.mraidha@cea.fr
sara.tucci@cea.fr
sebastien.gerard@cea.fr

Model-based development of real-time systems



- Should define a threading strategy satisfying timing constraints
- Manual phase: relies on the designer experience
- Timing validation (schedulability analysis) is needed

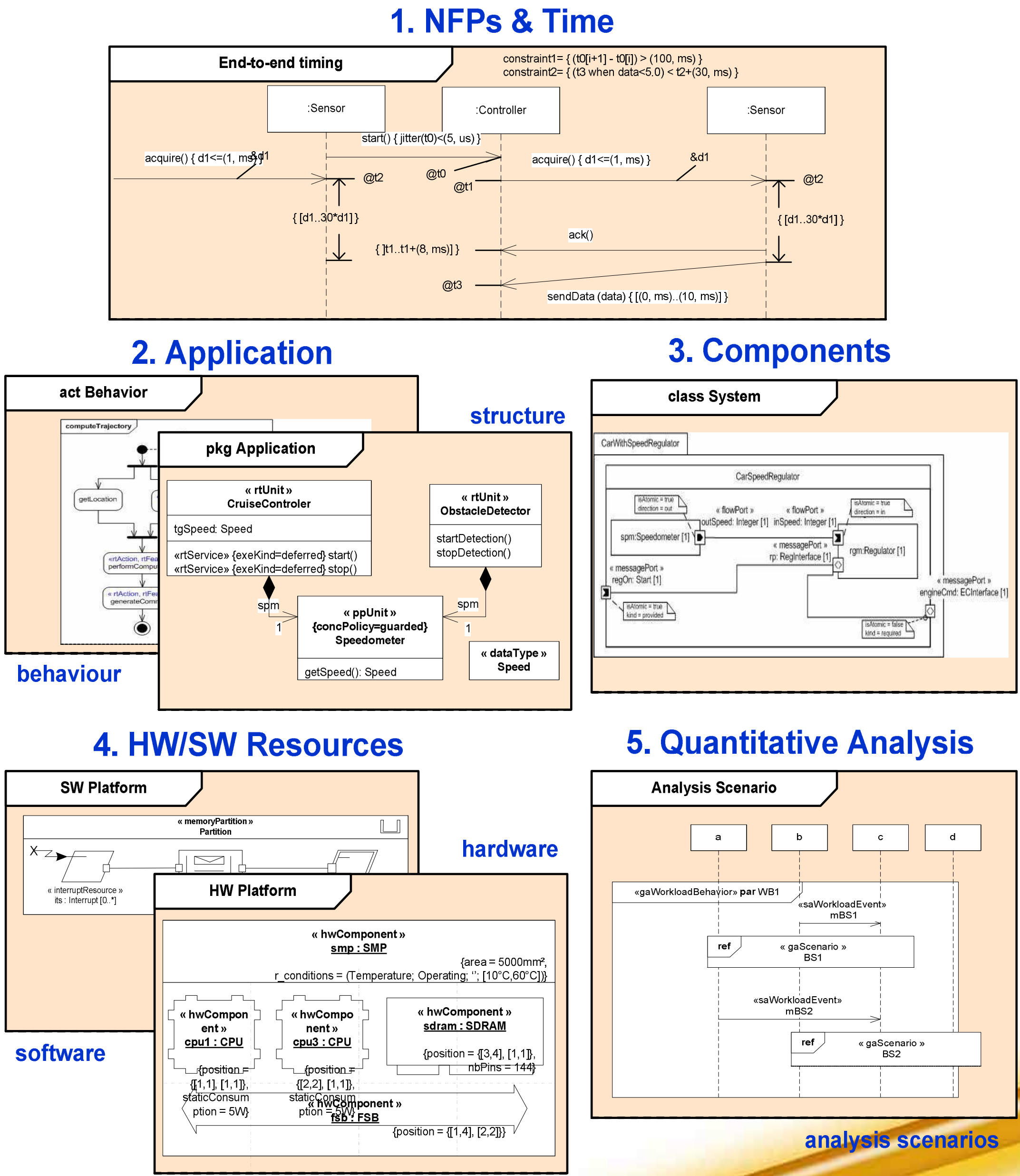
- Timing validation at early design stages
- Optimum: a methodology for schedulability guided design of real-time systems

Agenda

- **Outline**
- The Modeling Language
- The Optimum Methodology for Early Stage Schedulability Analysis
- Conclusions and Future Work

MARTE UML Profile

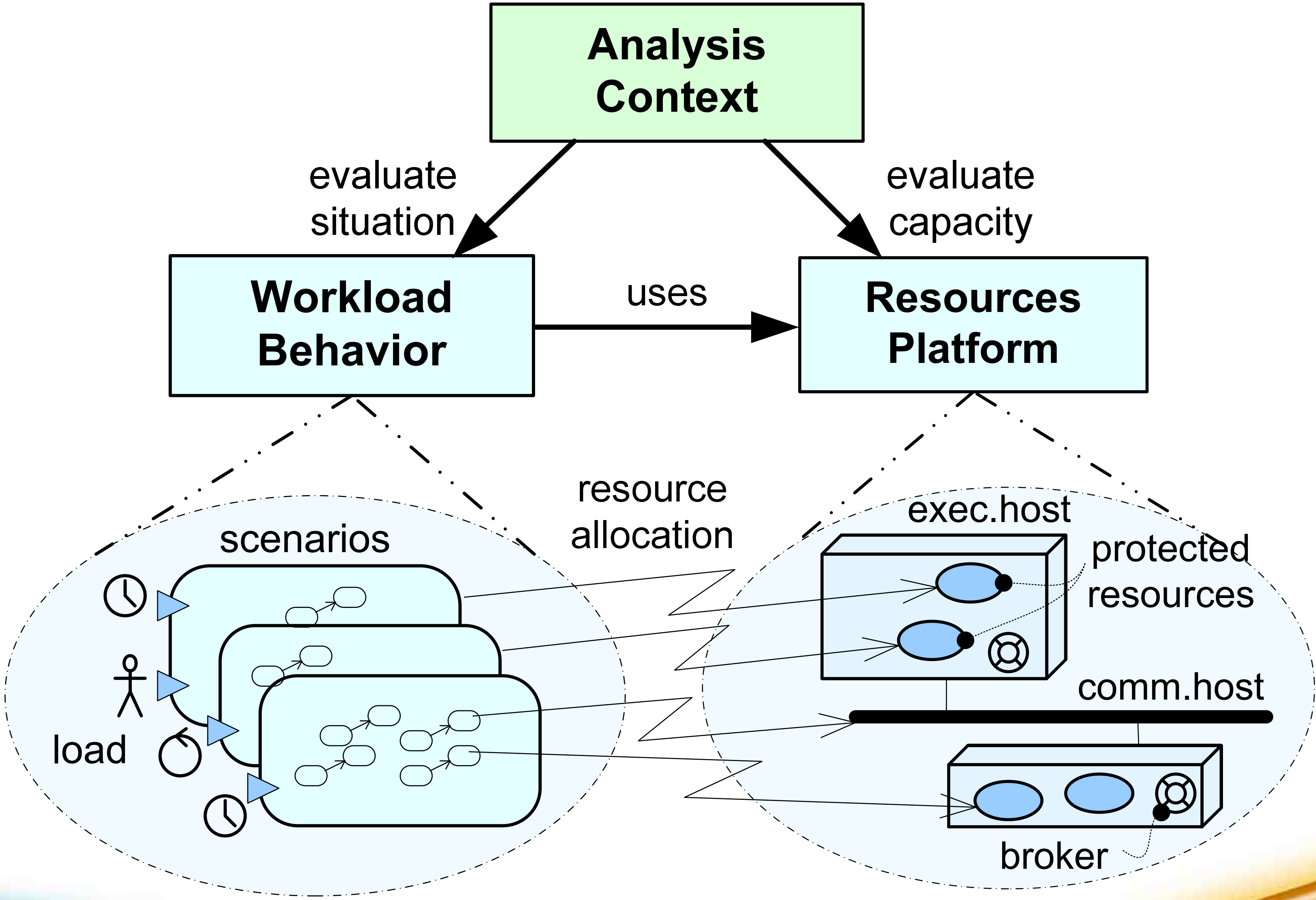
- **Modeling and Analysis of Real-Time and Embedded systems**
- **OMG standard: version 1.0 standardized in july 2009**
- **Structured in sub-profiles covering RTE systems development aspects**
- **Model libraries of RTE specific types**
- **Textual language for value specifications (VSL)**



Public website:

www.omgmarte.org

MARTE Concepts to Perform Analysis



MARTE4Optimum

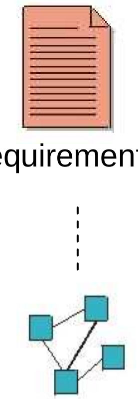
- **MARTE subset for Optimum methodology (out of MARTE 158 stereotypes)**
- **Restriction of stereotypes applicability (w.r.t UML base elements)**
  - Reduces complexity of methodological rules validation
  - Reduces complexity of automation support

MARTE4Optimum stereotype	Covered analysis concept	MARTE4Optimum UML extensions
Alloc::Allocate	Resource allocation	Abstraction
Alloc::Allocated	Resource allocation	CallAction, Property
GRM::SchedulableResource	Platform abstraction: task	Property
GQAM::GaPlatformResources	Platform abstraction: container	Class
GQAM::GaWorkloadBehavior	Workload behavior	Activity
GQAM::GaWorkloadEvent	Load: event with arrival pattern	AcceptEventAction
SAM::SaAnalysisContext	Analysis context	Activity
SAM:: SaEndToEndFlow	Scenario with deadline	ActivityPartition
SAM::SaExecHost	Platform abstraction: exec host	Property
SAM::SaSharedResource	Platform abstraction: shared res.	Property
SAM::SaStep	Workload behavior: action	CallAction

Agenda

- Outline
- The Modeling Language
- **The Optimum Methodology for Early Stage Schedulability Analysis**
- Conclusions and Future Work

Optimum Methodology: Inputs

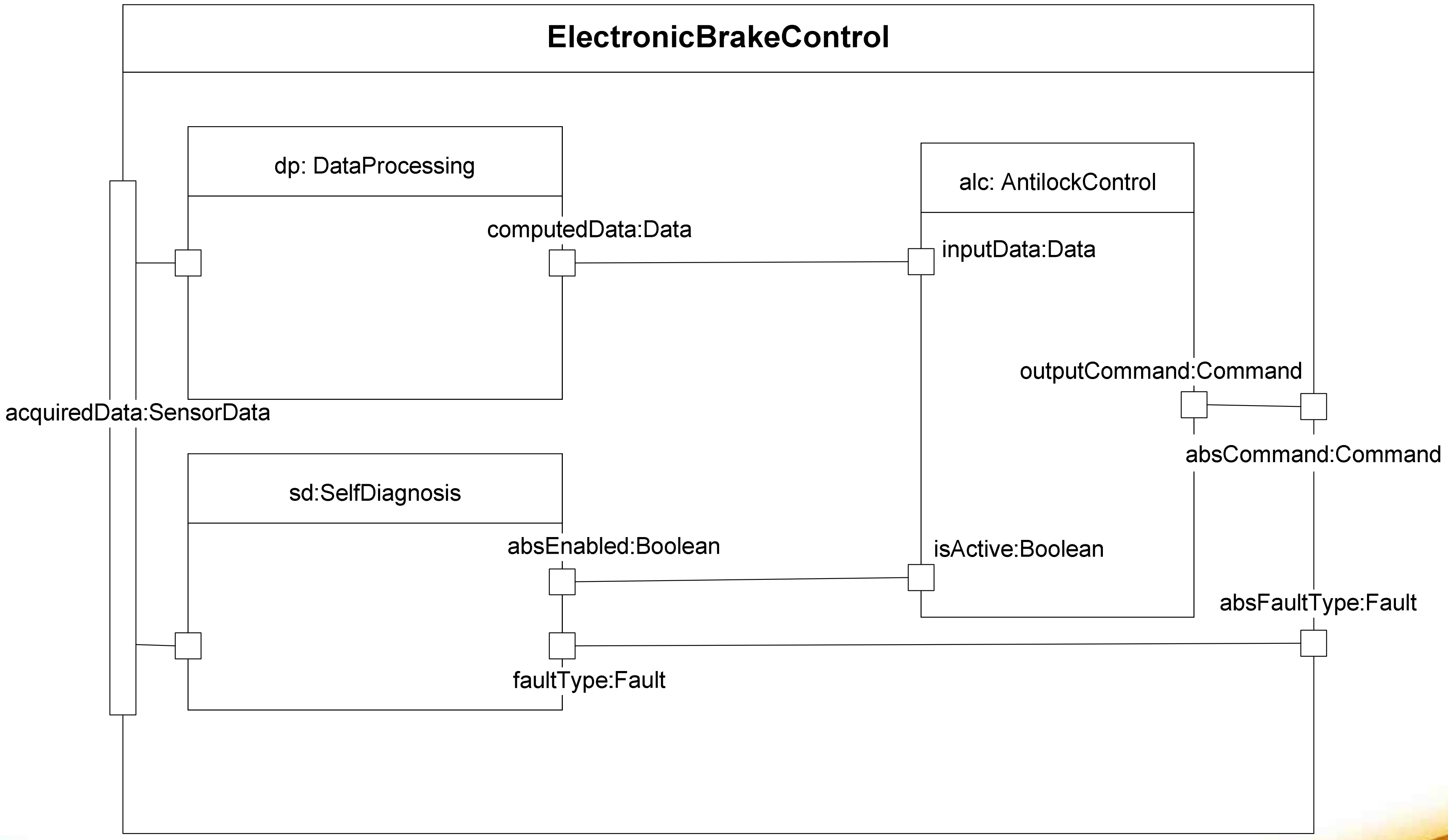


Requirements

Functional Model

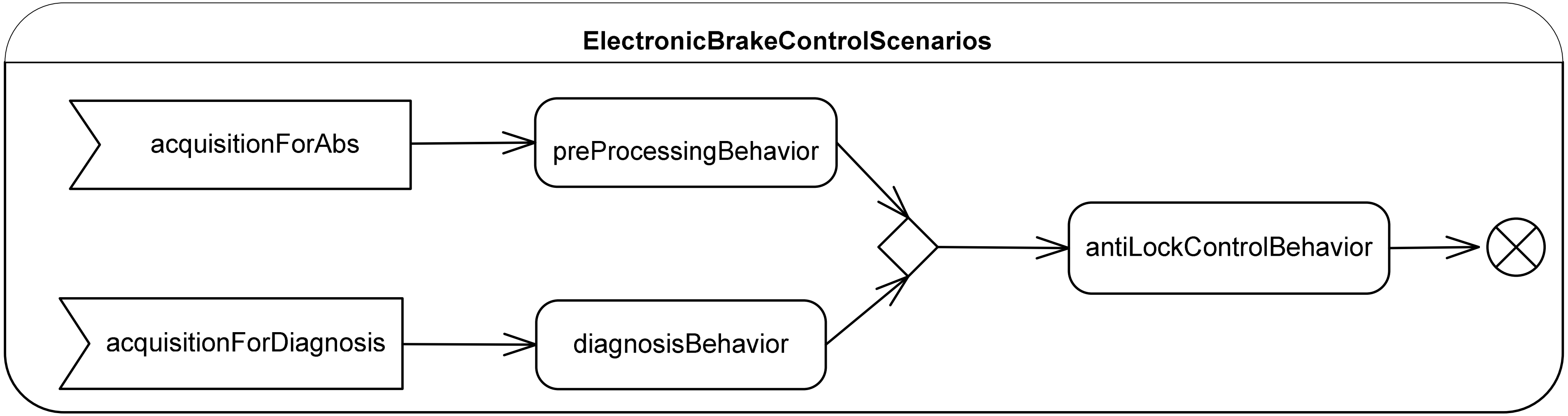
An Automotive Example

System-level functions structure



An Automotive Example

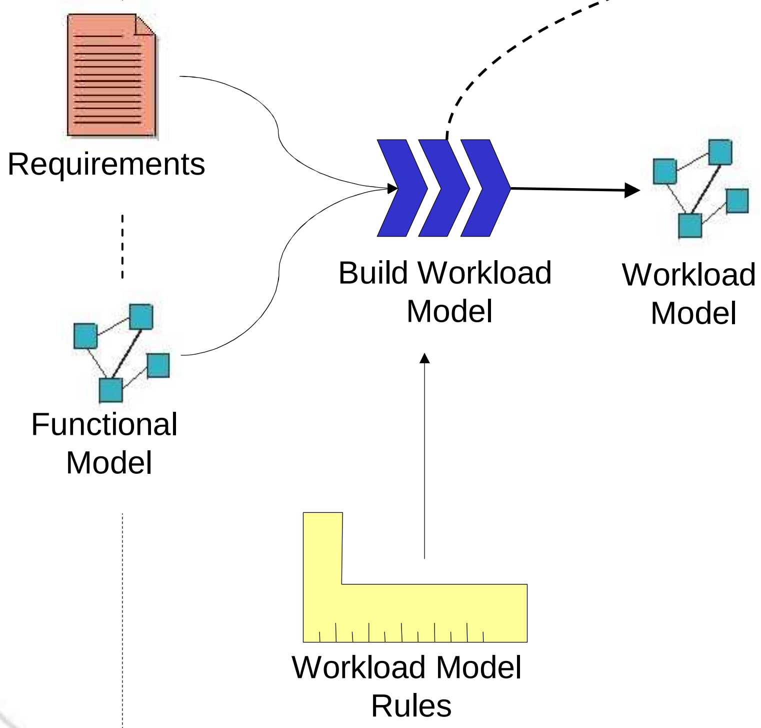
System end-to-end scenarios



Textual requirements

- The acquisition for ABS occurs every 60ms
- The acquisition for diagnosis can occur every 100ms at minimum
- ...

Build the Workload Model

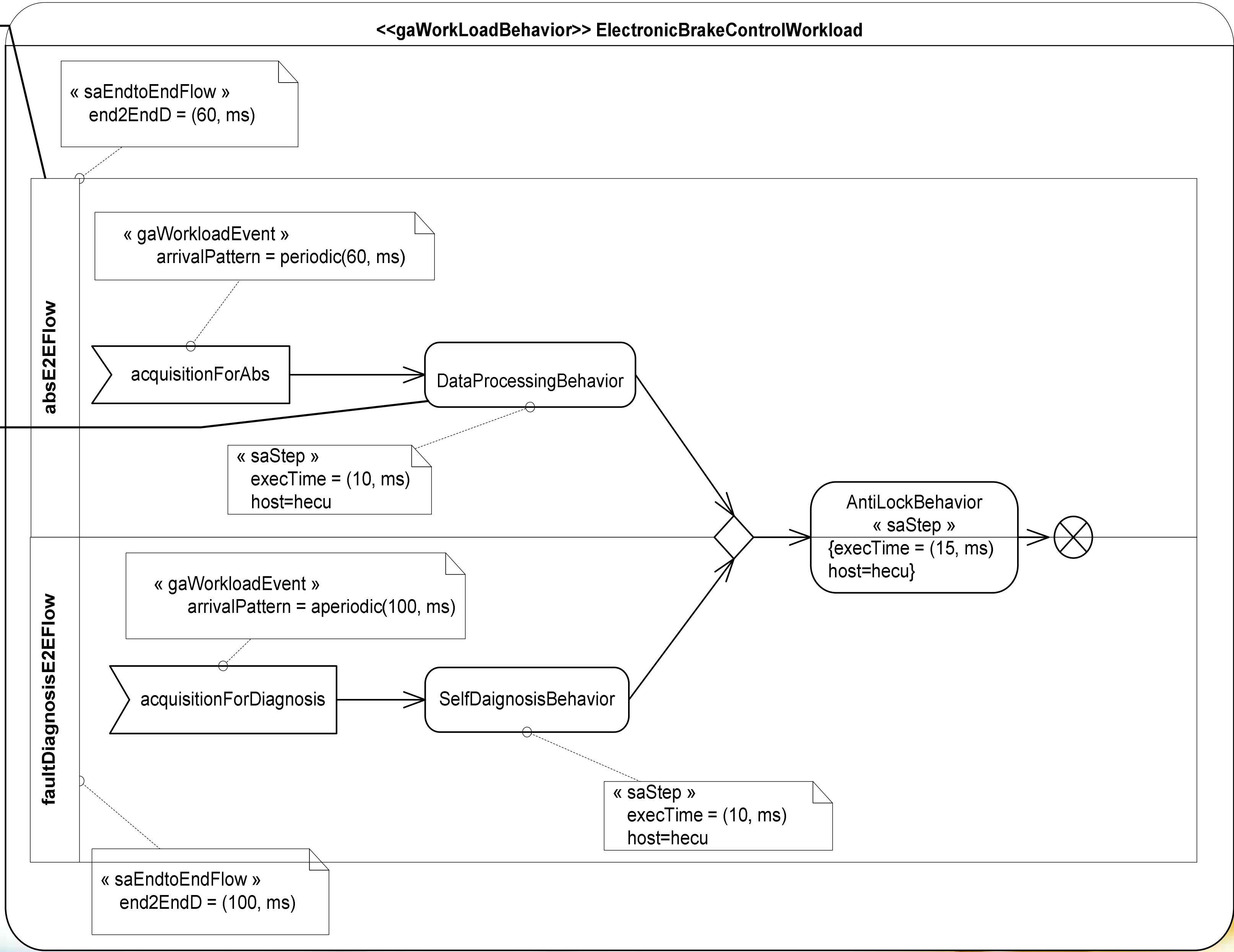


- **Workload behavior specification**
  - End-to-end flows identification
  - End-to-end deadlines
  - Event arrival pattern specification
  - Time budgets (w.r.t execution host)

- **Platform resources abstraction**
  - Execution hosts

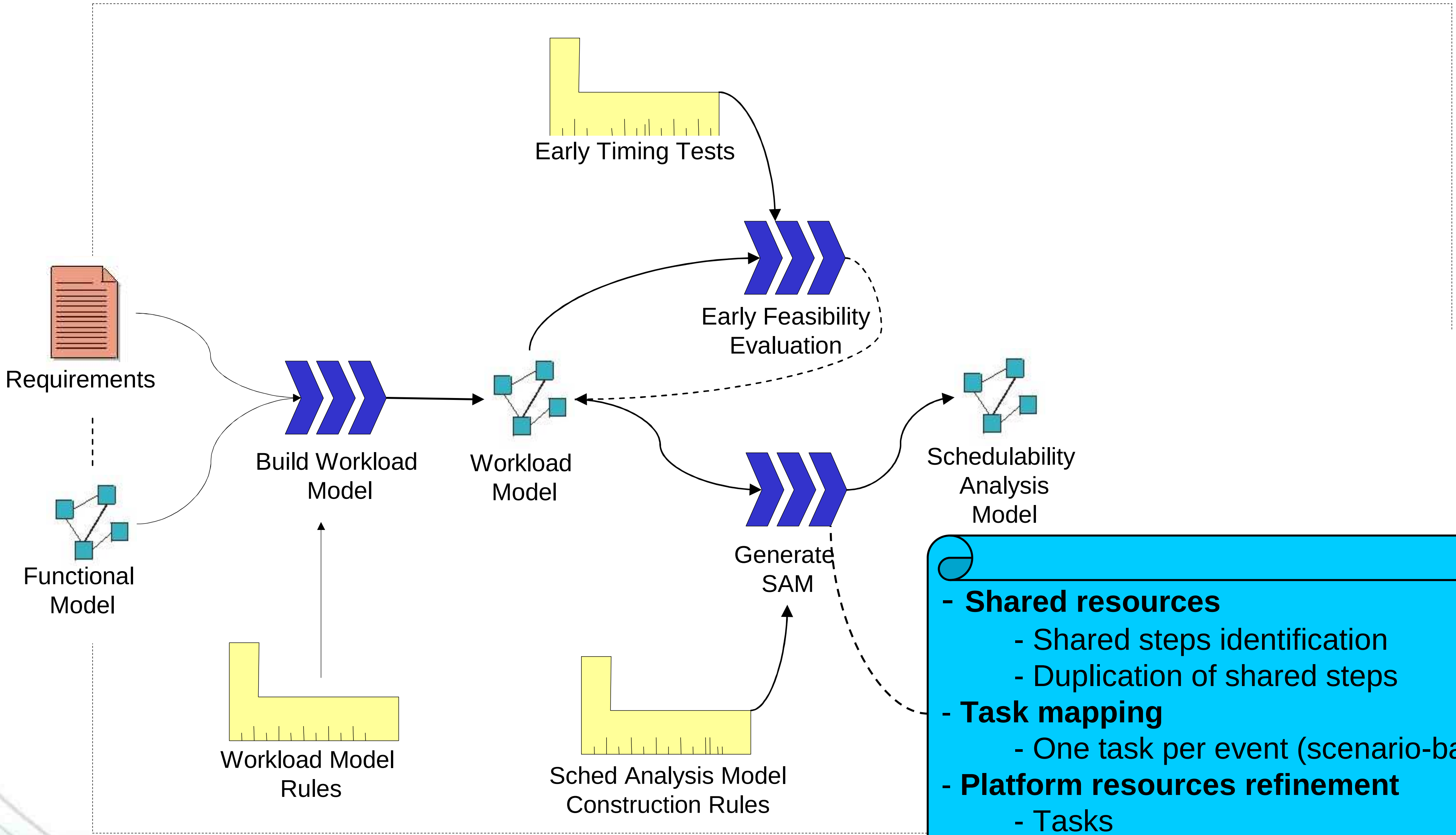
Build the Workload Model

End-to-end flow:
- Activity partition
- Deadline
- Event arrival pattern



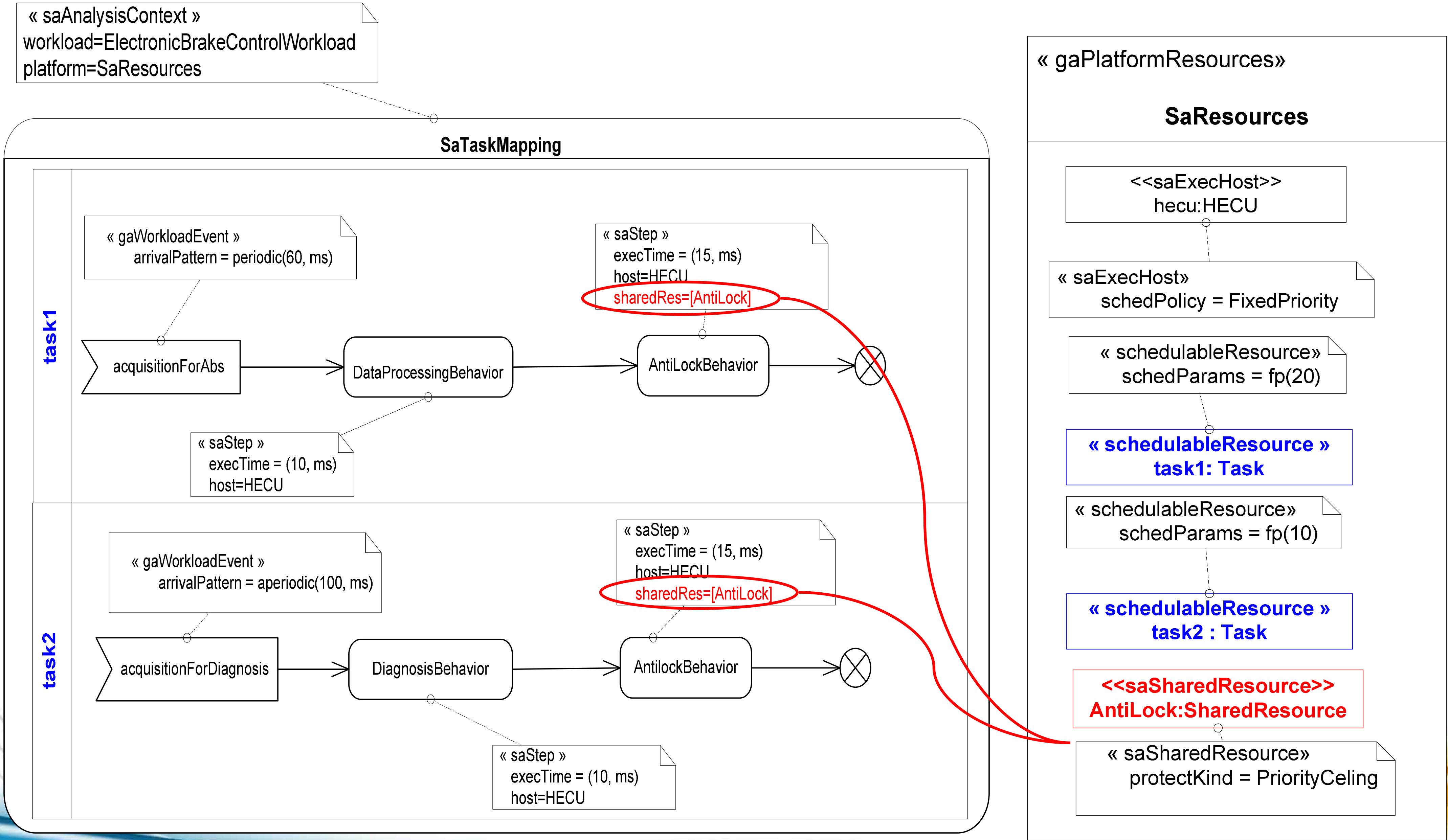
- Allocation on exec host
- Time budget

Build the Schedulability Analysis Model

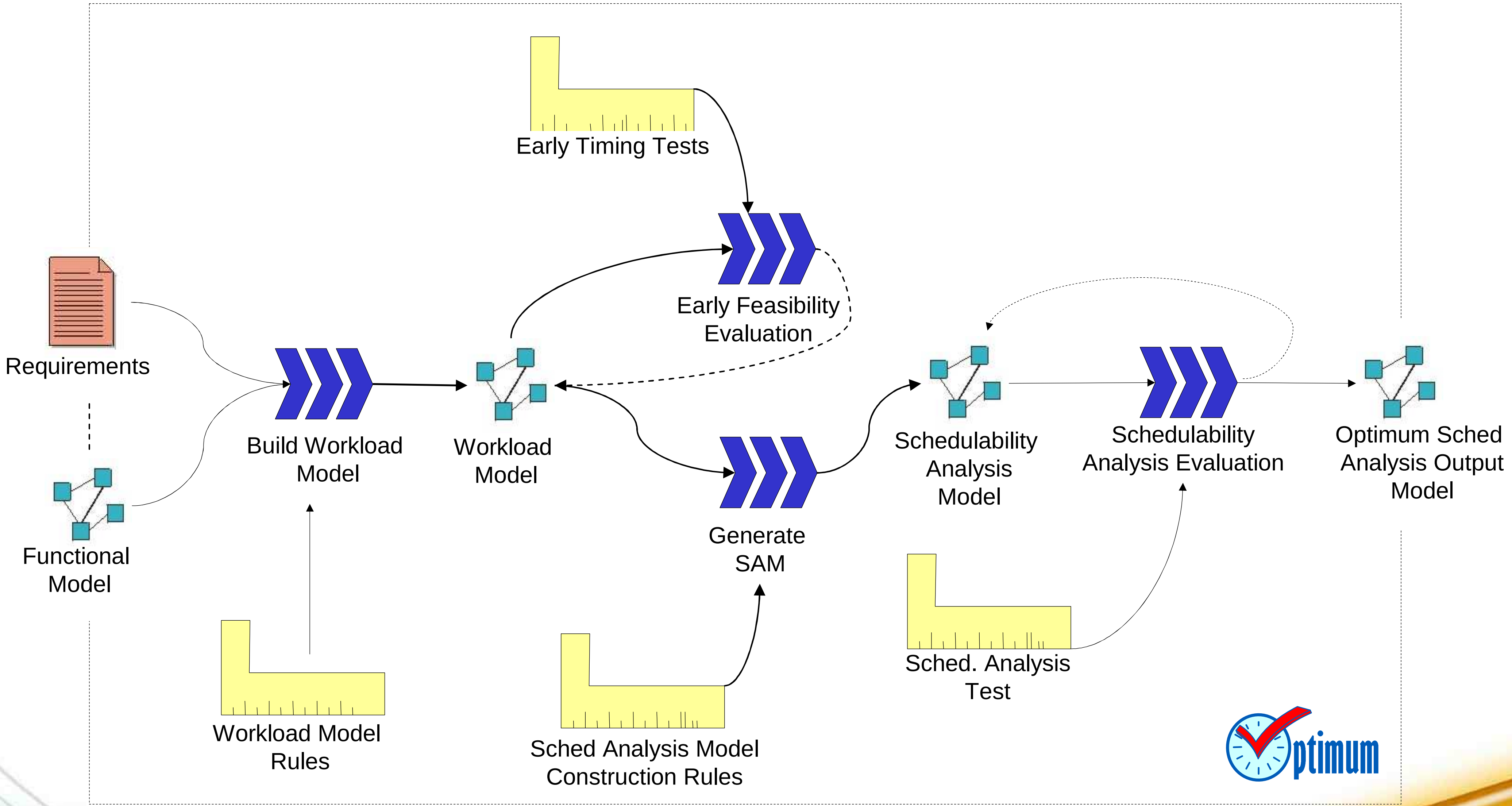


- **Shared resources**
  - Shared steps identification
  - Duplication of shared steps
- **Task mapping**
  - One task per event (scenario-based)
- **Platform resources refinement**
  - Tasks
  - Shared resources

Build the Schedulability Analysis Model



Schedulability Analysis Evaluation



Evaluation and Results

Schedulability analysis input model

task	e_i	T_i	C_i	P_i	B_i	D_i
task1	acquisitionForAbs	60	25	20	15	60
task2	acquisitionForDiagnosis	100	25	10	0	100

Schedulability analysis output results (back annotated)

task	*response time*	*isSched*
task1	40	true
task2	50	true

Back annotated schedulability analysis models provides guidance to build a schedulable design model

- Schedulable task mapping
- Shared resources identified

Agenda

- Outline
- The Modeling Language
- The Optimum Methodology for Early Stage Schedulability Analysis
- **Conclusions and Future Work**

Conclusions

- **MARTE-based methodology for schedulability guided design of RTS**
  - Reduces design cycle
  - Eliminates unfeasible design at early stage
  - Gives correct by construction real-time design patterns

- **Tooling support**
  - Optimum framework is integrated in Papyrus UML modeling tool
  - Provides wizards for capturing real-time properties in a simple way
  - Automatic construction of the schedulability analysis model
  - Bridges to external schedulability analysis tools (MAST and Rt-Druid)
  - Provides basic schedulability analysis tests (RMA for fixed priority)

- **Future work**
  - Enrich the set of task mapping construction
  - Enrich internal schedulability analysis algorithms
  - Real-time component models generation