Save-IDE

A Tool for Design, Analysis and Implementation of Component-Based Embedded Systems

Séverine Sentilles, Anders Pettersson, Dag Nyström, Thomas Nolte, Paul Pettersson, Ivica Crnkovic

PROGRESS

A national Swedish Strategic Research Centre

MÄLARDALENS HÖGSKOLA

MRTC
MÄLARDALEN REAL-TIME RESEARCH CENTRE

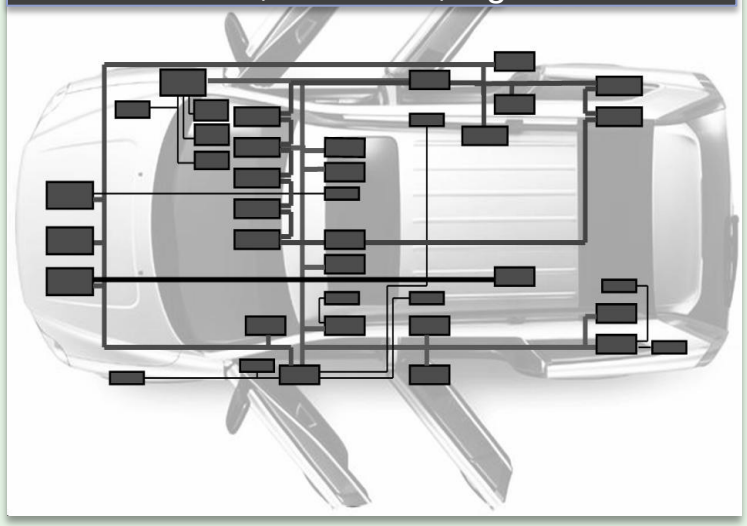
Stiftelsen för Strategisk Forskning

Project Overview

- Development:
  - Started in 2007 within the Save project
  - Now, continues within the Progress project
- Research project
  - 10 part-time developers (MsC and PhD students)
  - 5 researchers
- Open source:
  - http://sourceforge.net/projects/save-ide/
  - Eclipse plugins
- Aims:
  - Provides a platform to integrate research results
  - Illustrates the technology developed within the SAVE project
  - Evaluates the feasibility/advantages/limitations of the SAVE approach

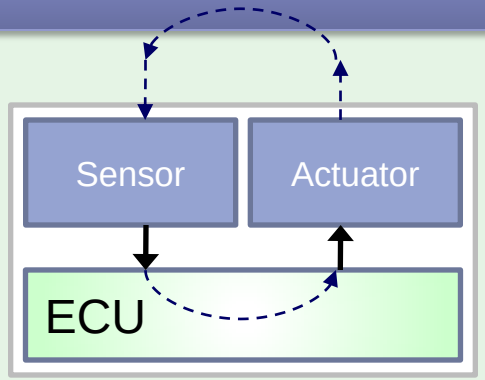
Vehicular Embedded Systems

Cruise control, Airbag, Turning light
Climate control, Infotainment,
Windows lift, Door lock, Light control



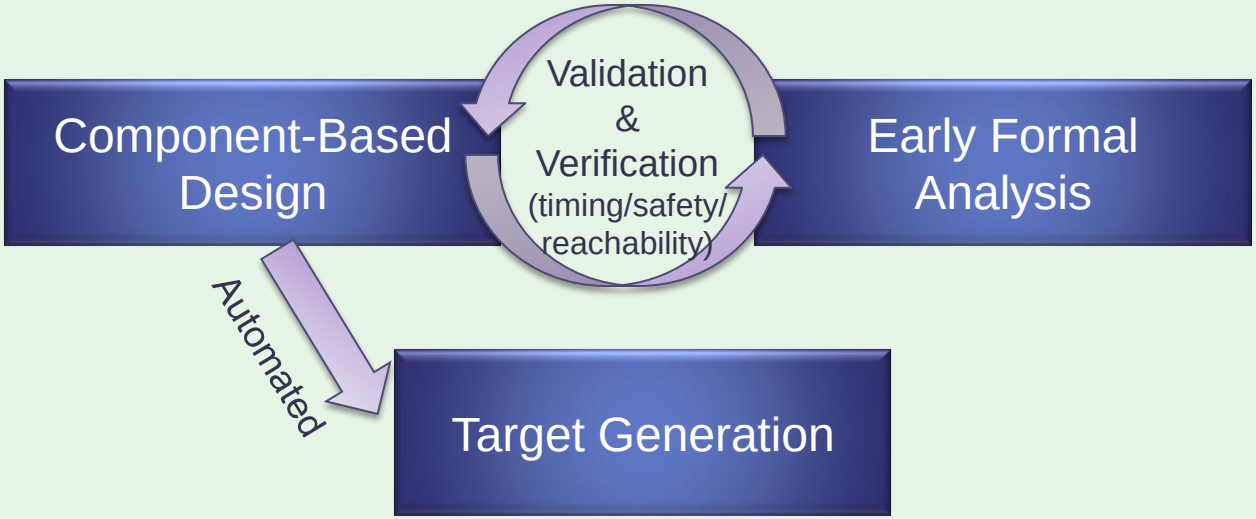
Provided by
Embedded systems

Environment – Vehicle mechanics



The SAVE Approach

- Coupling Component-Based Development with:
  - Formal Analysis Techniques
  - Model-Based Transformations
  - Automated Code Generation
- Early validation and verification of the system



Component-Based Development

- CBD goal:

  Increase efficiency in software development by:

  - Reusing already existing solution encapsulated in well-defined entities (components)
  - Building systems by composition of those entities (functional composition and extra-functional properties of the components)

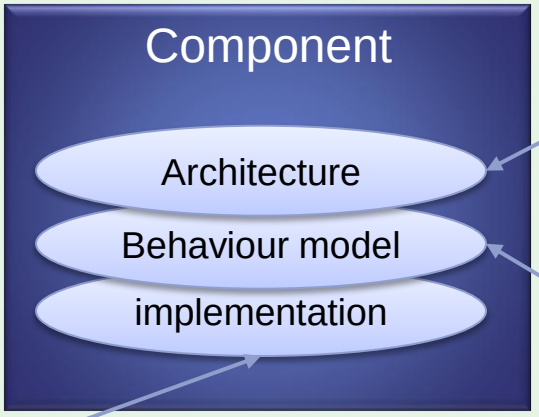
- Possible advantages brought by CBD:
  - Management of the complexity
  - Short time-to-market
  - Lower maintenance costs
  - Reusability

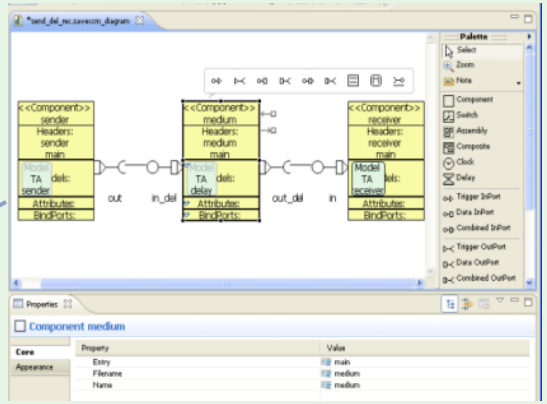
- However, CBD need some adaptation to support the specifics of vehicular ES

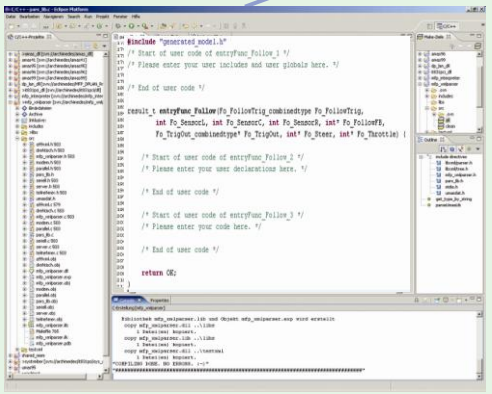
SaveCCM

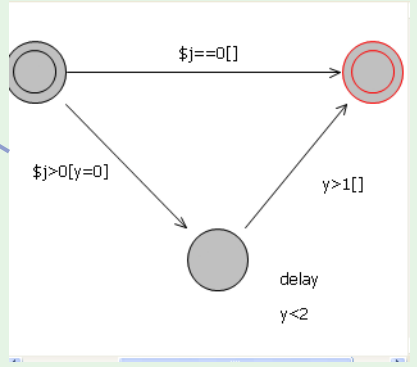
- Design-time component model, with:
  - One communication paradigm: Pipe & Filter
    - Separation of Data- and Control flow
    - Supports periodic (clock) and event-driven activities(external inputs)
  - Limited number of architectural elements
    - Component
    - Composite
    - Assembly
    - Switch
    - Clock
    - Delay
- Restrictive semantics:
  - Read-Execute-Write

The Component View

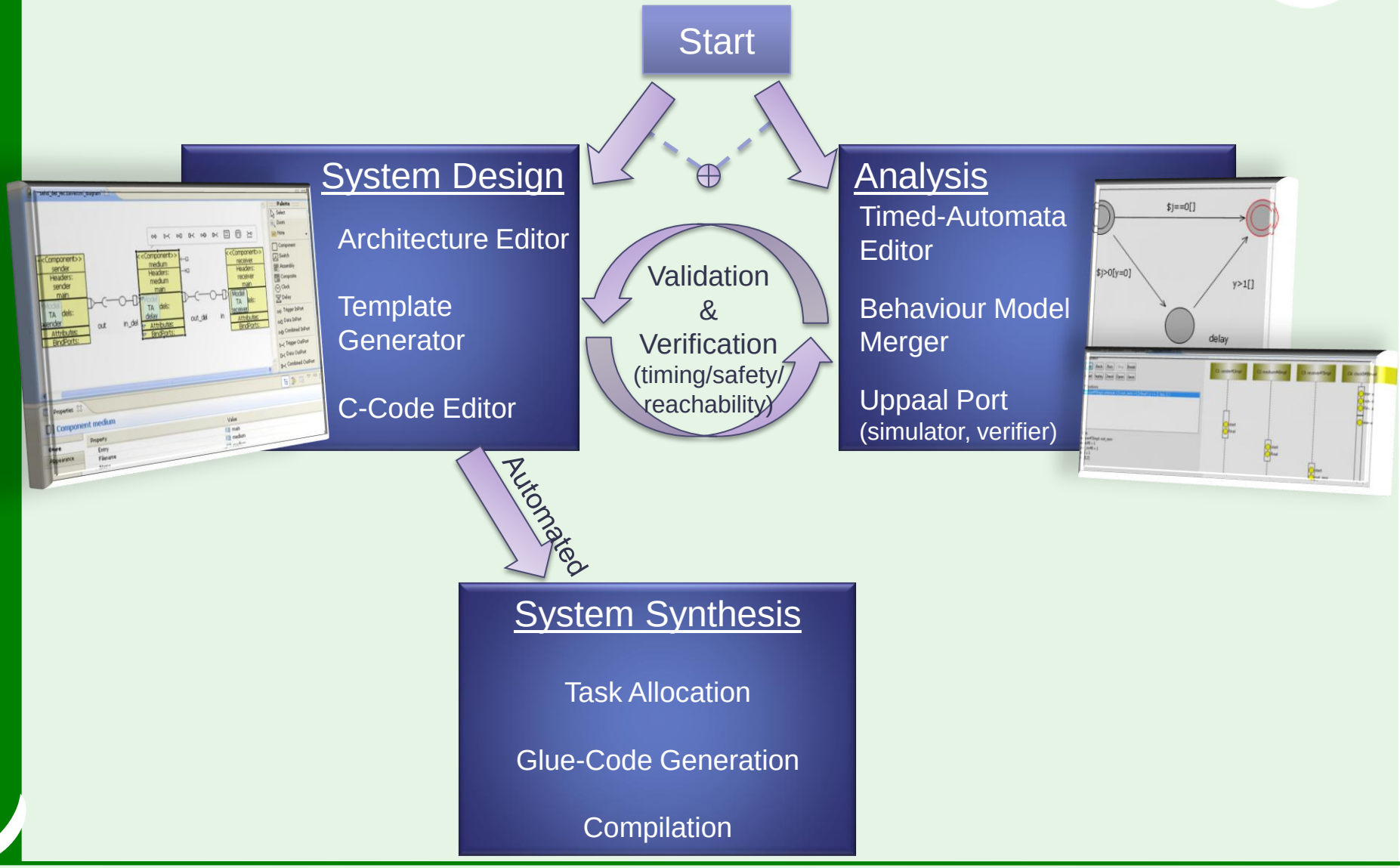








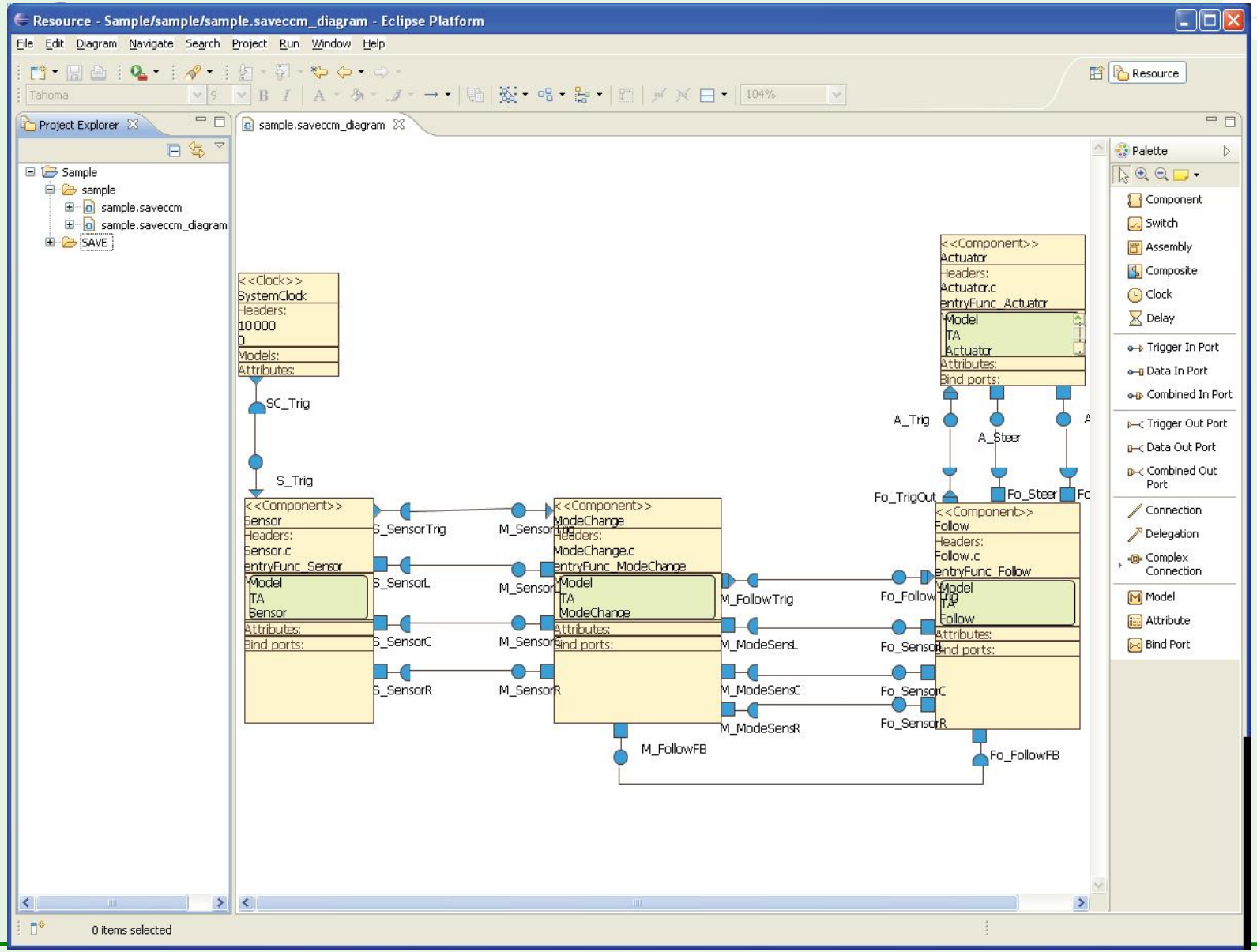
Overview of the Save-IDE



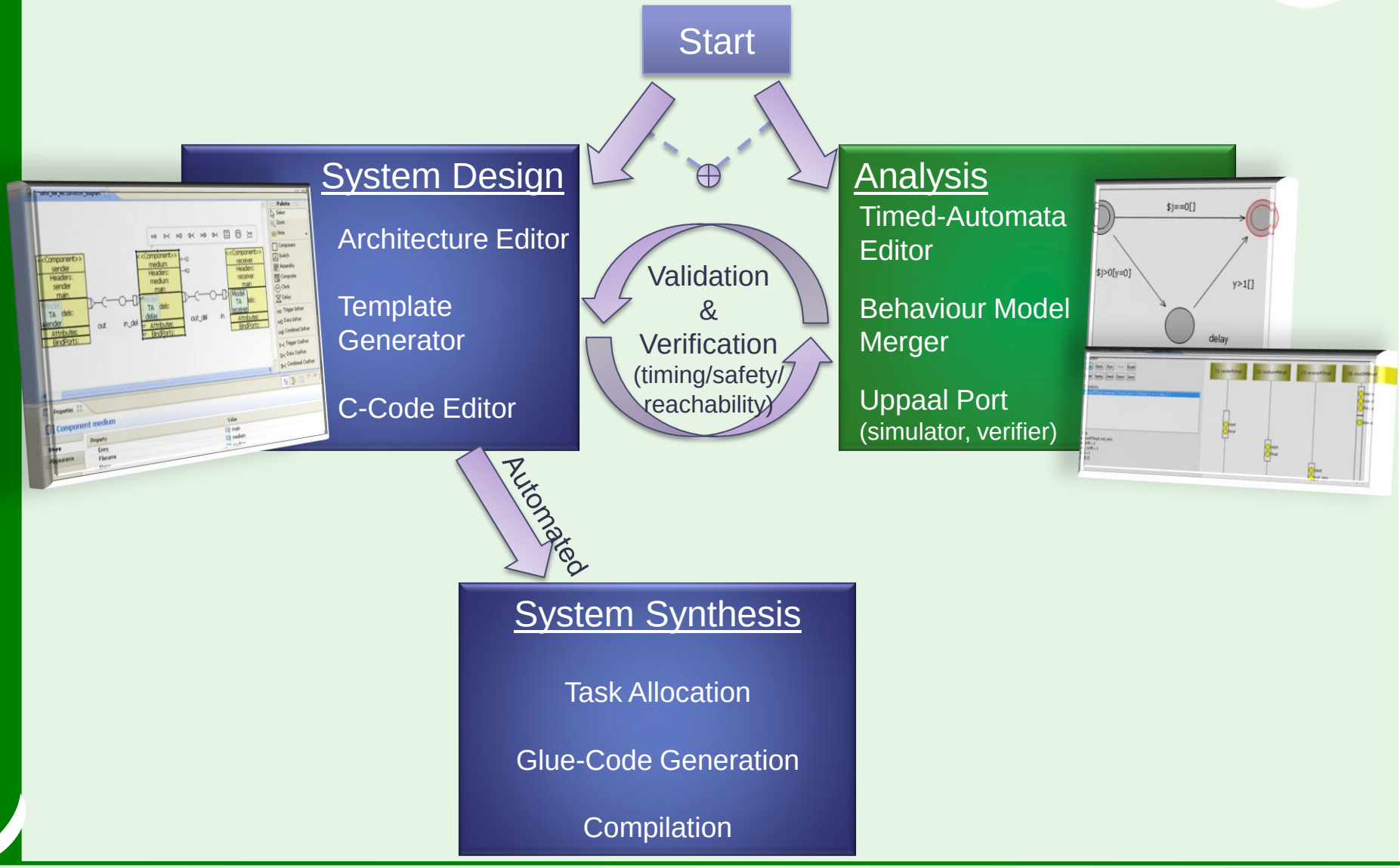
The Architectural Editor

- Design a system compliant with SaveCCM
- 2 views for each element
  - External view
    - Name, type, ports, included models
  - Internal view
    - Inner elements and their connections (composites, assemblies, switches)
    - Implementation (“primitive” component)
- Recursive
  - Elements inside composite elements have also these 2 views

The Architectural Editor



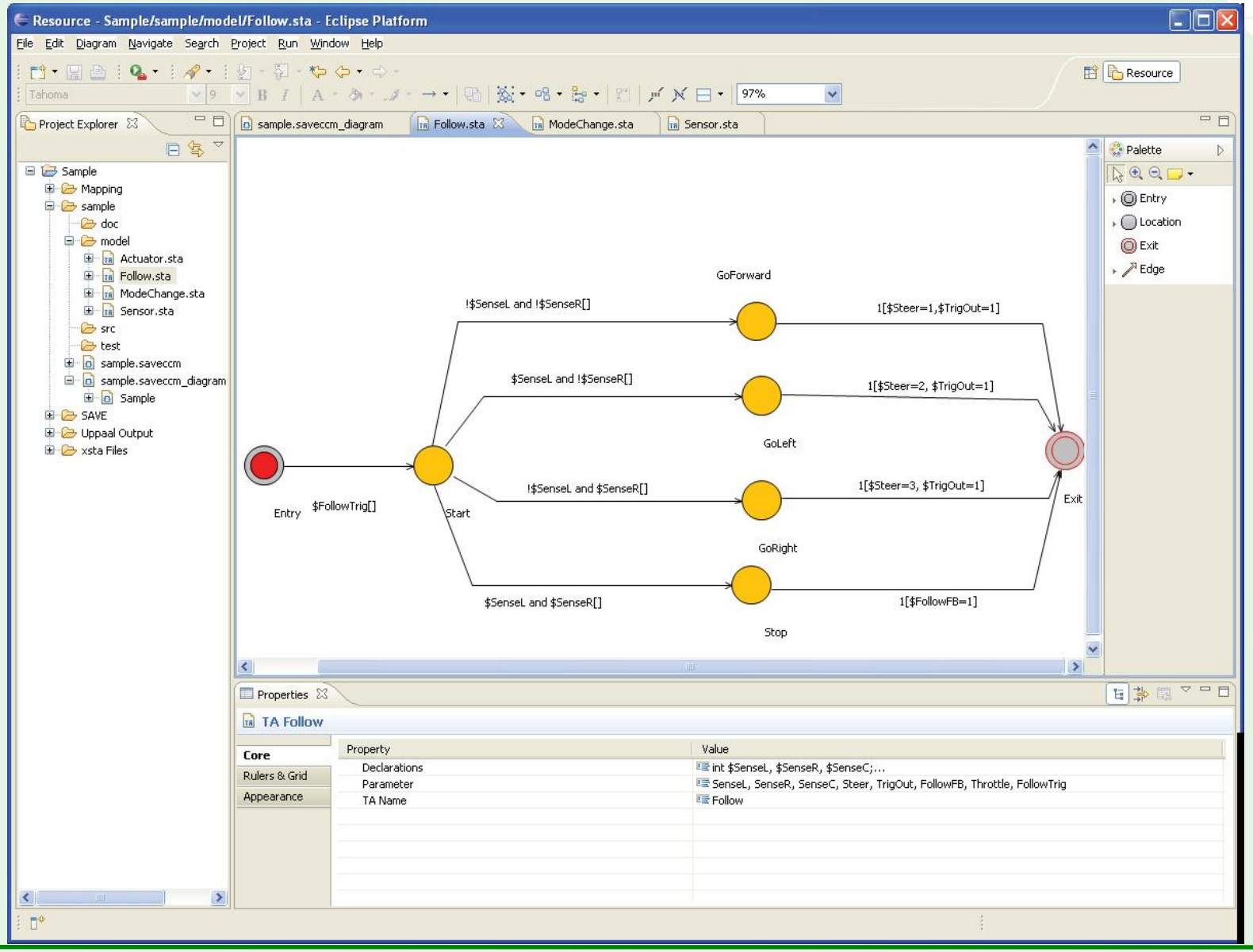
Overview of the Save-IDE



Analysis

- Consists of 4 tools
  - Timed Automata Editor
  - Behaviour Model Merger
  - Simulator
  - Model checker
- Aim:
  - Provide analysis of the system under development in early phases of the development process (prior any implementation)
- Properties that can be checked
  - Deadlock-free
  - Response time
  - Reachability
  - Liveness
  - Safety
  - …

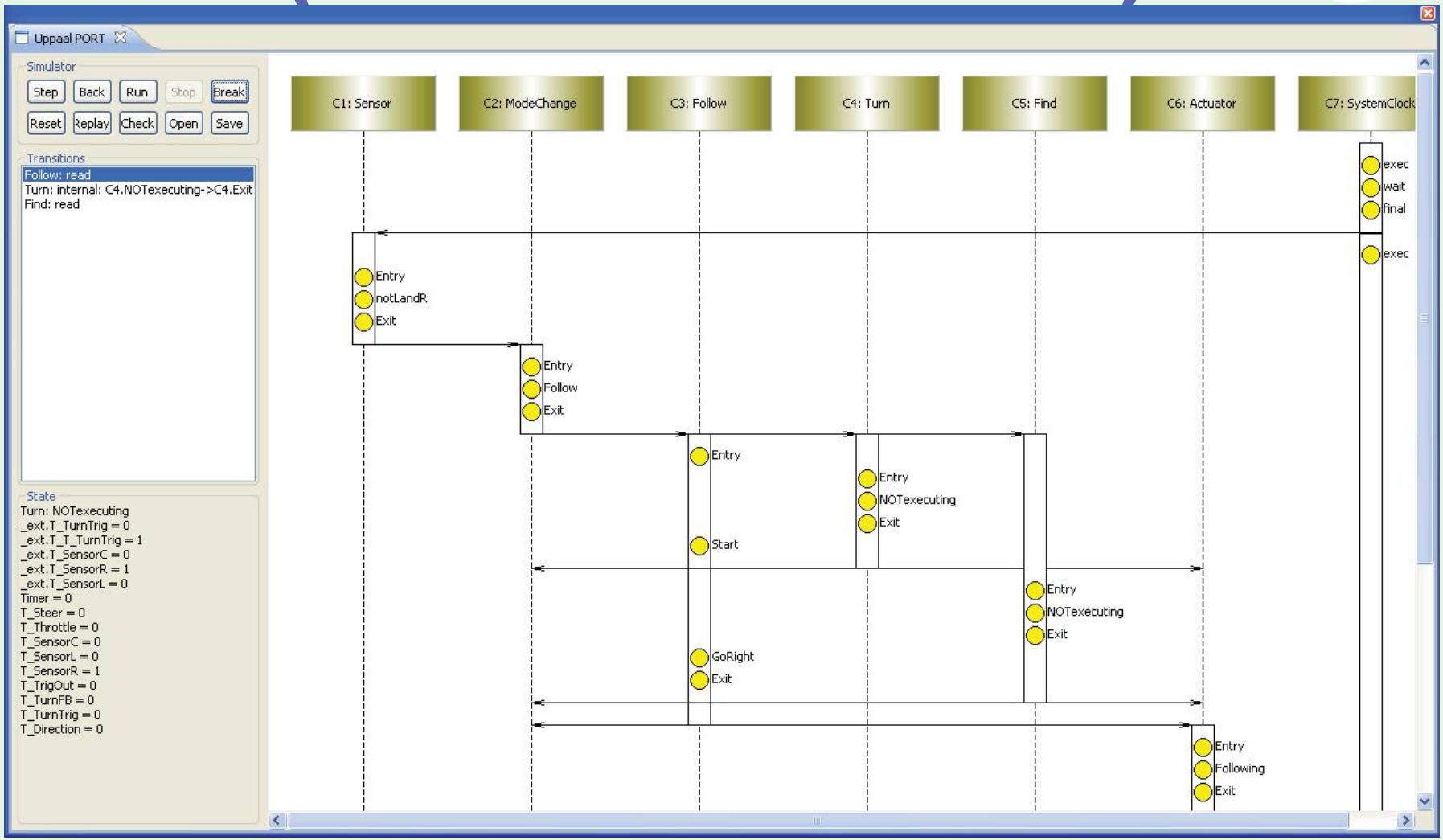
Timed Automata Editor



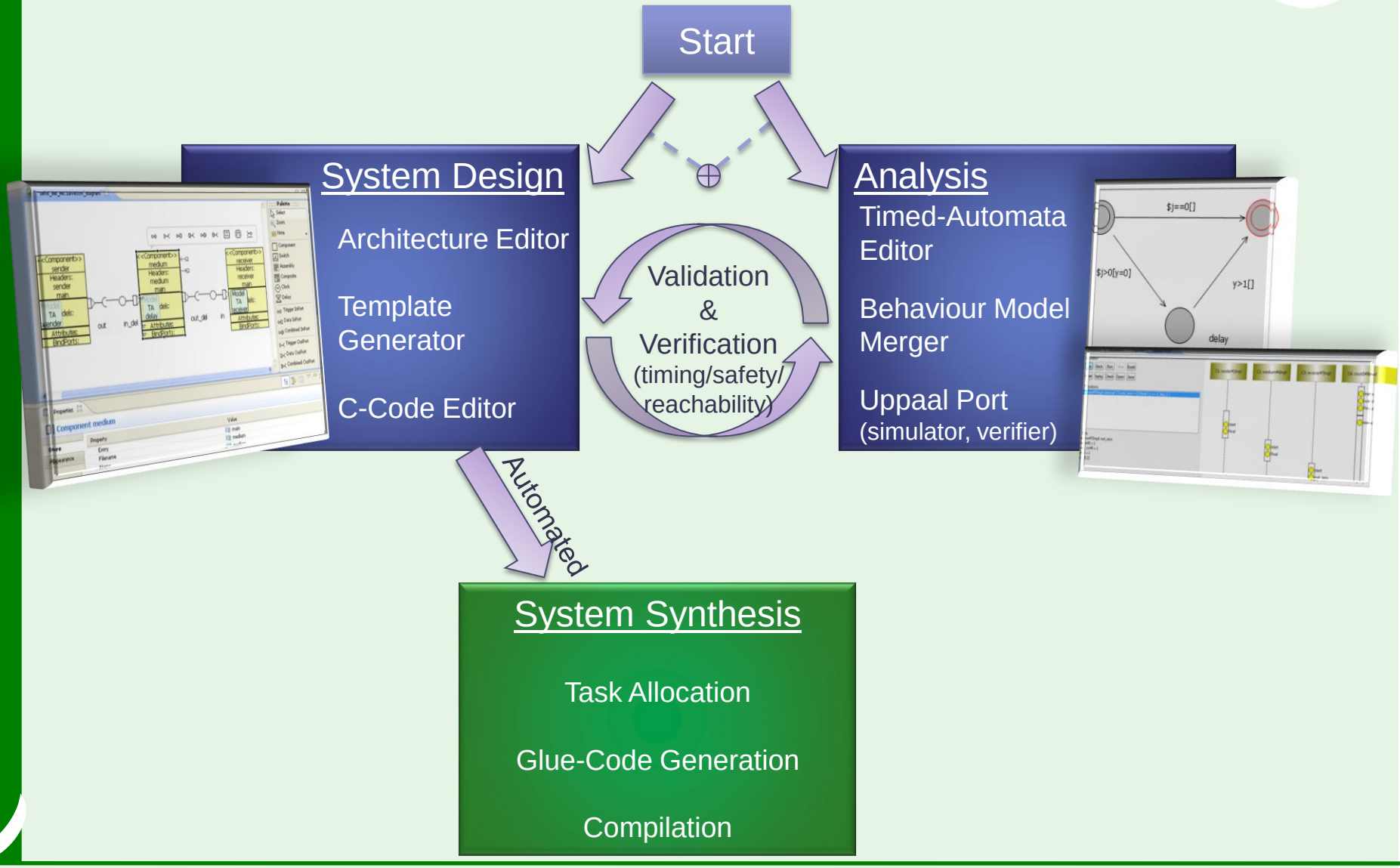
Uppaal PORT

- Component-based simulation and verification of real-time and embedded systems
- Based on UPPAAL (a timed automata model-checker)
- Extended with partial order reduction techniques
  - Exploits the structure and semantics of SaveCCM model to improve the model-checking performance
- Consists of 2 features
  - A graphical simulator
    - Allow to explore the dynamic behaviour of the system in the early phase of the development (prior any implementation)
  - A formal verifier
    - Check formal requirements specified as Timed CTL

Uppaal PORT (simulator + verifier)



Overview of the Save-IDE



Synthesis

- Set of automated generation tools which transform the SaveCCM-model into execution model (task model)
  - Constructs of a set of trees based on the control flow
  - 1 tree = 1 task
  - Generates the glue code
- Independent of the runtime environment
  - Uses the SaveOS
    - Abstraction layer between the actual run-time environment and application
    - Requires minimal computing, memory resources

Lessons Learned & Future Work

- Lessons learned:
  - Approach is well-suited for low-level control systems
  - But too restrictive
  - Components must be more than design-time units only
    - When to decide that a component model is a type?
    - Mix both bottom-up and top-down process
- Future work:
  - Integrate a new component model (ProCom)
  - Integrates a new language for modeling and describing resources (REMES)

Thank you

Save-IDE

A Tool for Design, Analysis and Implementation of Component-Based Embedded Systems

Downloadable from:
http://sourceforge.net/projects/save-ide/

PROGRESS
A national Swedish Strategic Research Centre

MÄLARDALENS HÖGSKOLA

MRTC
MÄLARDALEN REAL-TIME RESEARCH CENTRE

Stiftelsen för Strategisk Forskning