**Purpose:**
- Compositional verification.
- Reuse of components.

**Component interfaces:**
- A component interface defines boundary of the interaction between the component and its environment.
- Syntax given by a set of input and output ports.
- Semantics given by a set of traces (behavior, history of events and values on data ports).

**Component implementations:**
- Implementation of a basic component: behavioral model (state machine).
- Implementation of a composite component: composition of subcomponents’ implementations (composite implementations).

**Component environments:**
- Environment of the system component: behavioral model (state machine).
- Environment of a subcomponent: composition of sibling subcomponents’ implementations and parent’s environment (composite environment).

**Contracts:**
- Assumption/guarantee pair \( (A, G) \) of properties.
- Assumption must be satisfied by the component environment \( (Env \models A) \).
- Guarantee must be satisfied by the component implementation \( (Imp \models A \rightarrow G) \).

**Contracts refinement:**
- Condition on the contracts of a components and its subcomponents.
- Ensure that:
  1. Composite implementations are correct.
  2. Composite environments are correct.

**Contract-based design:**
- Draw system architecture.
- Specify components’ contracts.
- Check if the contracts refinement in the system architecture is correct.

**Compositional verification:**
- Specify behavioral models of basic components.
- Check if they satisfy the contract of the basic components.
- System contract is guaranteed by the contract refinement!

**Reuse of components:**
- Consider existing component with verified contract.
- Check if this refines the component’s contract in the system architecture.
- Correctness of reuse guaranteed by the contract refinement!

**The OCRA tool:**
- OCRA=Othello Contract Refinement Analysis
- Developed within SafeCer
- Contracts’ assertions specified in Othello
  - Language for embedded systems properties.
- Built on top of NuSMV3 for infinite-state model checking.
- Integrated with CASE tools:
  - CHESS
    - Developed by Intecs.
    - For SysML and UML modeling.
  - AutoFocus3
    - Developed by Fortiss.
    - For synchronous system architectures.

**SafeCer**
Safety Certification of Software-Intensive Systems with Reusable Components

**OCRA: A Tool for Checking the Refinement of Temporal Contracts**

By Alessandro Cimatti, Michele Dorigatti, Stefano Tonetta

[Diagram of system architecture and components]

https://es.fbk.eu/tools/ocra