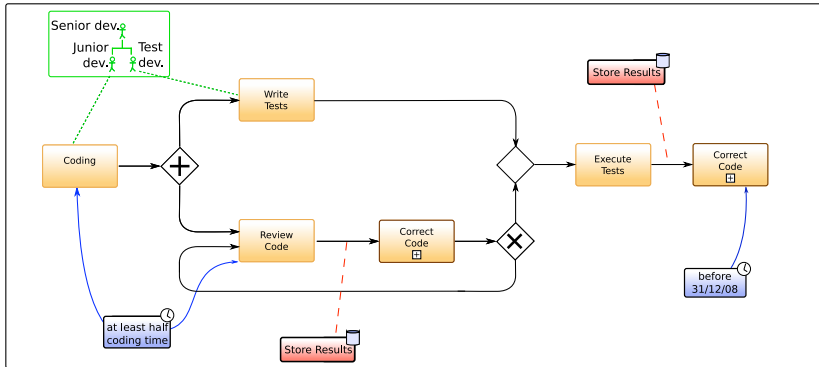


Functional Workflow Concerns



- **Control:**
What tasks have to be performed (business-wise), and in what order.
- **Time:**
Temporal restrictions on the process execution.
- **Resources:**
What resources are available to perform each task.
- **Data:**
How data relevant to the process has to be managed.

Requirements

Independent Implementations

Concerns have to be implemented separately so they can be designed, evolved and maintained independently. Each concern should have a specific language suitable to describe its models. Each concern can have different technology requirements.

Relations between Concerns

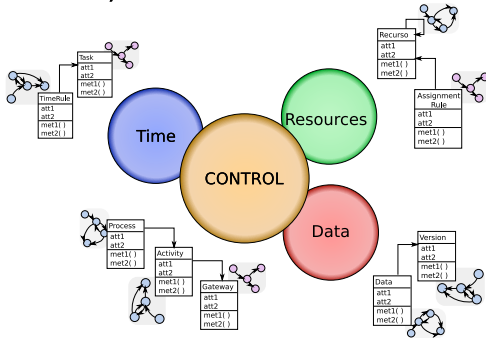
Relations between participating concerns are specific to each application or process. Usually, the control concern is related to the other ones; however, it is also possible to have relations between other concerns. Since there should not be hard dependencies between concerns, reuse of concern implementations should be common and easy.

Composition / Weaving

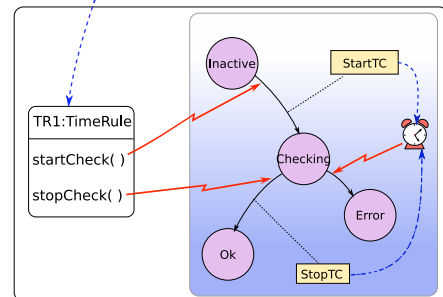
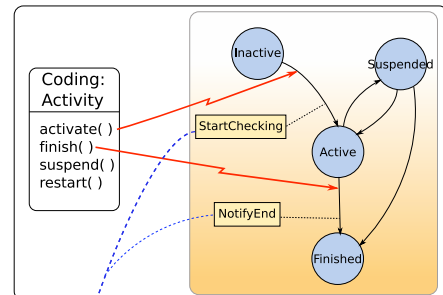
When executing, concerns should behave in a coherent way. This has to be achieved through weaving of concerns' elements. To be successful, weaving requires fine-grained composition and coordination hooks. The granularity of the hooks depends on the implementation of the concerns.

Implementation using Executable Models

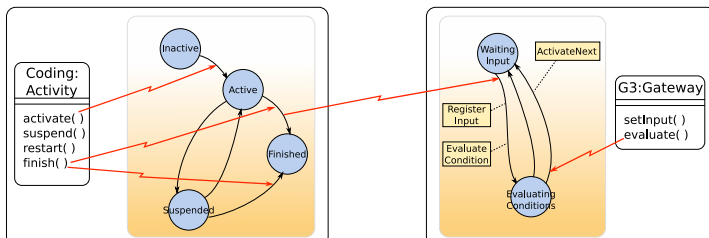
A One concern, One executable model



C Synchronization of elements and state machines from different concerns



B Executable models built with 'open objects'



Open Objects are the basic construction element for our executable models. They are composed by an 'entity' and a state machine.

Consistency between an entity and its state machine, or between two open objects, is achieved with a mechanism based on events and method calls.

State machines have 'actions': operations associated to transitions which are executed when transitions are taken. Actions can call methods of any entity.

A weaving language is used to describe relations between elements from different concerns. These relations are concretized with actions, which are capable of invoking methods of entities even if they belong to different concerns.

In the example above, there is an activity (control concern) that must observe a time rule (time concern): when 'Coding' is activated, the action 'StartChecking' forces the time rule 'TR1' to start checking the duration.