Score formalism : User need example

Project : an intermedia project example to base on Score formalism

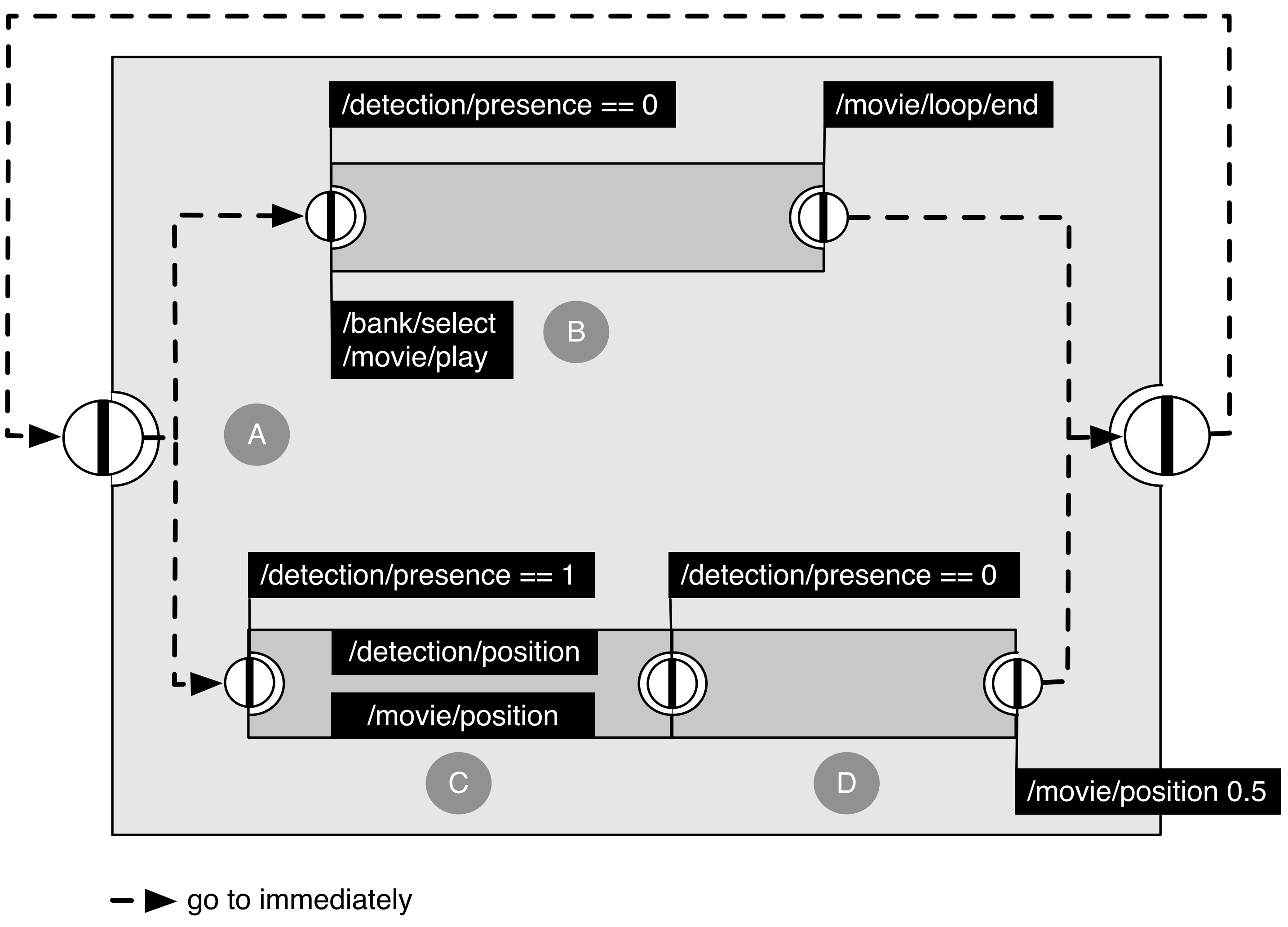
A face of a priest is displayed on a TV. While nobody walks in front of the TV, several movies show how bored the priest is (looking at the sky, at his watch, …). But when someone is walking in front of the priest (and when a bored priest face movie ends), the priest starts to look at the person fixedly. His eyes follow her to get attention. Then, when the person is gone, the priest is bored again. This example is based on the project of Abtin Sarabi, student at the ISDAT School.

Ressources : all hardware and software services the project use

There are a movie player, a movie bank and a camera detection system that can returns someone's presence in front of the TV and that can track the position of the person. The state of those resources can be addressed, memorized, recalled or listened using protocol solutions.

Scenario : formalization of the temporal and logical operations that the project involves

Here the project is translated using the Score formalism. Consider it uses the same protocol solutions as above to memorized resources states at edition and drive the resources at execution.



A — two conditions on **/detection/presence** state to select either **B** (if equal to 0) or **C** (if equal to 1).

B — select randomly a new movie using **/bank/select** message and play it using **/movie/play.** Then wait the **/movie/loop/end** return before to loop on **A.**

C — map the **/detection/position** return on the **/movie/position** parameter. The mapper is active until **/detection/presence** goes to 0. Then it moves on **D**.

D — drive the **/movie/position** parameter to reach 0.5 in a precise amount of time before to loop on **A**.

Lexicon

Services : what a device offers to be used : parameters, messages, returns.
For example, it could respectively be an audio track volume, the play trigger button and the end of the play.

Namespace : all services name organized into a tree structure.
For example :

```
player
      play
      volume
      end
```

State : service value

▬ **Actions** : operation made on state :
- set : update state
- get : ask state
- listen : observe state

Condition : verification made on the state.
For example : "/player/gain > 12" => true or false.

◐ **Event** : a point in the time that can be triggered at a precise date or if a condition is true.
Some actions can either be necessary to trigger the event or be done when the event happen.

▨ **Process** : operations made in the time between two events :
- wait : a amount of time that can be either static, ranged or free
- automation : set services value according curves or trajectory
- mapping : listen a service value to scale it according another service value to set
- scenario : contain actions related to events and processes