

# Comments on “Modelling Concurrent Process Coordination in Workflow Specifications” by A.P. Barros and A.H.M. ter Hofstede

Jan L.G. Dietz

*Technical University Delft, The Netherlands*

## Discussant’s summary of the paper

The topic of the paper is the conceptual modelling of workflows, and the formalisation of these models. A workflow is understood to be a process or task structure.

Four basic constructs, i.e. coordinating mechanisms are distinguished in a task structure:

- *Trigger*. The activation or initiation of the execution of a task by some other task, is called triggering. A trigger is an instance of triggering.
- *Decision*. A decision is a construct that takes a trigger as input and dispatches it in one of a number of output directions, according to some condition being fulfilled. The output directions seem to be mutually exclusive.
- *Synchroniser*. A synchroniser takes a number of triggers as input and produces a number of triggers as output. On the input side it seems to function as an AND gate.
- *Buffer*. A task may produce items (data, values) that are put in a buffer, from which some other task consumes items.

The authors propose to extend these constructs by two new ones: abstract messaging and aborts. Three abstract messaging configurations are distinguished:

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35500-9\\_30](https://doi.org/10.1007/978-0-387-35500-9_30)

- *Waiting request.* A subtask, called the sender, of some task A, sends a request to a subtask, called the acceptor, of some task B, and suspends its execution until another subtask of B, called the returner, sends a response back to the sender.
- *Non-waiting request.* A subtask, called the sender, of some task A, sends a request to a subtask, called the acceptor, of some task B. After some time, another subtask of B, called the returner, sends a response back to another subtask of A, called the receiver.
- *Transfer.* A subtask, called the sender, of some task A, sends a message to a subtask, called the receiver, of some task B.

An *abort* is a special type of trigger. The effect of generating an abort is that the task to which it belongs is terminated. A secondary effect may be that a special task, called the abort handler, is executed.

### Discussant's comments

In the list of comments hereafter, the comments are marked with one of three marks: "+", "-", "?". Their meaning is respectively: positive or strong point, negative or weak point, and question or unclear point.

- + The authors demonstrate that they are well able to provide elegant formal representations of process or task structures.
- ? It is yet uncertain whether these formal specifications can easily be applied for simulating the dynamics of a task structure, as well as for direct execution.
- It is not so easy to grasp what the authors exactly had in mind when proposing the extensions. Comments from the reviewers regarding the confusing explanation in section 2 were not taken into account.
- ? Why didn't they apply more well-known graphical formalisms, like the Petri net, to explain task structures?
- Although the models are indeed conceptual in the sense that they abstract from realization issues, they fail to be conceptual in the sense of being purely about work flow or business processes. This holds particularly for the 'informative' communications, like the buffer and the transfer(?).