Typesafe Eventful Programming with DCR Graphs - Extended abstract*

João Costa Seco Søren Debois Thomas Hildebrandt
Tijs Slaats

Abstract

Online and distributed software systems, in particular web and mobile applications, are designed around data, computations and processes that orchestrate computations, user interactions and third-party services, to meet a specification, usually drawn from their business oriented counterparts. The result is that business processes are usually encoded into the applications' code, via a number of unwritten conventions and entity (data) status values, based on which (imperative) code is designed to orchestrate and compute all the necessary outputs.

This software engineering problem is magnified by the fact that contemporary business processes have parts that interact or form sub-processes of other processes. For example, the process for assessing funding applications has parts that overlap with the process for running payout of successful applications. Both overlap the overall process for executing the call-assess-payout-report lifecycle of a funding agency.

In this talk, we introduce the ongoing work on an extension of DCR graphs [1], presented as single reactive language [2] capable of expressing distributed systems, while embedding data transformation computations REBLS15deep in the process definition layer. Process events, that include input and output events, are designed to encapsulate control flow, computations, and the state of the application. We introduce a mechanism that makes event constraints conditional on the state of the application in a natural, and more expressive way. The state of an application includes input from the environment, computed data, and also meta information about the status of the existing (process) event instances, queried in a structured way. For instance, in the example above, an automatic output event to communicate the acceptance of an application is triggered when a minimum number of 3 reviewers (working in a sub-process) signals the acceptance of the project.

The orchestration of underlying computations, usually expressed using imperative code, is naturally defined in this approach by direct dependencies between data items, as in a spreadsheet, and the process constraints in the event

*Authors listed alphabetically

graph. We propose a validation procedure on valued DCR graphs that ensures that the data transformations are sound. Moreover, besides laying a natural specification abstraction, both graph views, the one of event constraints and the one of data dependencies between event expressions, lay ground worth exploring in data privacy and provenance situations as a way of slicing and controlling data usage.

Technically, our contributions are (1) the extension of DCR graphs to aggregate data-carrying sub-processes, with declarative specification of execution constraints through (2) xpath-like queries on aggregate process state; and (3) type composition primitives for this extended notation, inspired by typed reactive programming.

References

[1] Søren Debois, Thomas T. Hildebrandt, and Tijs Slaats. Safety, liveness and run-time refinement for modular process-aware information systems with dynamic sub processes. In *FM 2015: Formal Methods - 20th International Symposium, Oslo, Norway, June 24-26, 2015, Proceedings*, pages 143–160, 2015.

[2] Miguel Domingues and João Costa Seco. Type Safe Evolution of Live Systems. In *Workshop on Reactive and Event-based Languages Systems (REBLS'15)*, 2015.