Gelee: Cooperative Lifecycle Management for (Composite) Artifacts

Marcos Báez, Cristhian Parra, Fabio Casati, Maurizio Marchese,
Florian Daniel, Kasia di Meo, Sivia Zobele, Carlo Menapace, Beatrice Valeri

University of Trento, Italy
{baez,parra,casati,marchese,daniel}@disi.unitn.it,
{katarzyna.dimeo, silvia.zobele, carlo.menapace, beatrice.valeri}@studenti.unitn.it

Abstract. In this demonstration we introduce *Gelee*, our online platform for the hosted specification and cooperative execution of lifecycles of artifacts of any kind. With Gelee we aim at filling two lacks we identify in current cooperative software systems when it comes to unstructured, artifact-based works (e.g., the writing of a project deliverable): the lack of state and the complete lack of automated actions. Lifecycles allow us to model the state of any object, and if we focus on online resources (e.g., a Google Doc) then we can also automate some lifecycle actions. If we apply Gelee to composite artifacts, e.g., a set of web services, lifecycles provide for the human-driven orchestration of services.

Keywords: Lifecycle Management, Artifacts, Online Resources, Gelee

1 Introduction

Historically, the spectrum of cooperative software has been divided into two macro-areas: *process-centric* systems (e.g., workflow management or service orchestration systems) and *document-centric* systems (e.g., groupware or sub-versioning systems). The former typically suffer from a too rigid imposition of the process logic, not allowing users to easily adapt or change a running instance; as a consequence, such systems do not suit unstructured, creative works without predefined process. Ad-hoc or adaptive workflow management systems or case handling systems only partially introduced flexibility into process-centric systems. Document-centric systems, on the other hand, typically come without any explicit notion of state for the work being assisted by the system (the state is represented by the data in the system) and, hence, there is no automated coordination of the work or support for automated actions.

We argue that everything has a *lifecycle*, a real-world object (e.g., a car) the same way as a creative work (e.g., the writing of a deliverable). If modeled in terms of phases and transitions, the lifecycle of an artifact allows us to capture some notion of state of the artifact. While in general we cannot automate the progression of a lifecycle for a given object, the people working on the artifact know how it changes during its life. So we rely on humans to progress lifecycles. In projects where multiple artifacts are manipulated, this already grants the project coordinator visibility into the state of each artifact (e.g., to fill a progression report), a feature that is only scarcely supported by any project management tool on the market (if at all). We then specifi-

cally focus on online resources, which typically come with an API (a web service) that allows the enactment of actions on the resource. By binding a lifecycle to specific resource (e.g., a Google Doc), Gelee allows for the automation of the API's actions by extending the lifecycle model with resource-specific actions (e.g., the translation into PDF), thus alleviating the work of human actors. Composite artifacts (e.g., the writing of a paper and its submission to a conference) can be obtained by combining atomic artifacts, and lifecycles can be used to coordinate the interaction with their APIs, practically yielding a human-orchestrated service composition.

In this demo we show Gelee at work, and we show that it indeed is an answer to many situations that cannot be adequately managed with existing cooperative software. The demo introduces the Gelee online platform, the lifecycle editor, the execution environment, and the monitoring tool. Gelee itself implements a SOA and allows one to plug in new services through a dedicated registry. The platform includes a SOA middleware for resource management, with on top the lifecycle management applications.

2 Demonstration flow

In this demonstration we will show the Gelee prototype at work. This prototype implements the concepts in [1] providing artifact lifecycle modeling, progression, and monitoring. The goal of this demonstration is to introduce the user to the Gelee features and underlying concepts in the following flow:

1. First, we put Gelee into context to explain what the key contributions and the novel features of the tool are.
2. Then we show the Gelee system at work, starting from the Gelee *workspace*.
3. From the workspace we move to the *modeling environment* to describe the modeling features with an example of a deliverable lifecycle.
4. We follow then this example to describe the *execution environment*, in which we bind the actual deliverable (e.g., in Google Docs) to the lifecycle defined in the modeling environment. In particular, we show how we operate on the resource by executing and configuring lifecycle actions.
5. After introducing both environments, we briefly show the *monitoring widgets* and how they can be included into web dashboards (e.g. iGoogle).
6. Finally, we summarize the demonstration and we mention our ongoing and future work.

A short video describing the above demonstration flow is available at the following address: http://project.liquidpub.org/gelee/docs/gelee-demo.wmv.

3 References

[1] Marcos Báez, Fabio Casati, Maurizio Marchese. Universal Resource Lifecycle Management. ICDE 2009, pp. 1741-1748.