Barcelona: A Design and Runtime Environment for Declarative Artifact-Centric BPM

Fenno (Terry) Heath, III[1], David Boaz[2], Manmohan Gupta[3], Roman Vaculín[1], Yutian Sun[4,⋆], Richard Hull[1], and Lior Limonad[2]

[1] IBM T.J. Watson Research Center, USA
{theath,vaculin,hull}@us.ibm.com
[2] IBM Haifa Research Lab, Israel
{davidbo,lliori}@il.ibm.com
[3] IBM Global Business Services, India
manmohan.gupta@in.ibm.com
[4] University of California, Santa Barbara, USA
sun@cs.ucsb.edu

A promising approach to managing business operations is based on business artifacts, a.k.a. business entities (with lifecycles) [8, 6]. These are key conceptual entities that are central to guiding the operations of a business, and whose content changes as they move through those operations. A business artifact type is modeled using (a) an information model, which is intended to hold all business-relevant data about entities of this type, and (b) a lifecycle model, which is intended to hold the possible ways that an entity of this type might progress through the business. In 2010 a declarative style of business artifact lifecycles, called Guard-Stage-Milestone (GSM), was introduced [4, 5]. GSM has since been adopted [7] to form the conceptual basis of the OMG Case Management Model and Notation (CMMN) standard [1]. The Barcelona component of the recently open-sourced [2] ArtiFact system supports both design-time and run-time environments for GSM. Both of these will be illustrated in the proposed demo.

The GSM approach will be illustrated in the demo using a simplified *OrderToCash* scenario. Figure 1 shows a screen shot from Barcelona that provides a view of the design editor for this scenario. The focus is on a single artifact type, called `CustomerOrder`. The information model, which is essentially a record with scalar and nested relation fields, is not illustrated in the figure, but is accessible from the tree on the left side. The example focuses on the steps of *Drafting* a (customized) product based on an incoming order; *Submitting* the draft for approval; and then *Processing* the order. These activities are captured as top-level stages in the GSM schema for this example, which is illustrated in the right-hand portion of Figure 1. Stages may contain a single task (as is the case with *Drafting* and *Submitting*), and may be nested (as is the case with *Processing* and inside that, with *Preparing*). Stages may be executed once or repeatedly, and may execute in parallel.

Launching of a stage execution is controlled by rules-based *guards*, designated using blue diamonds on the right edge of the stage. These may be triggered by an external event (in which case a yellow lightening bolt is included), or by internal events.

⋆ This author supported in part by NSF grant IIS-0812578.

S. Basu et al. (Eds.): ICSOC 2013, LNCS 8274, pp. 705–709, 2013.

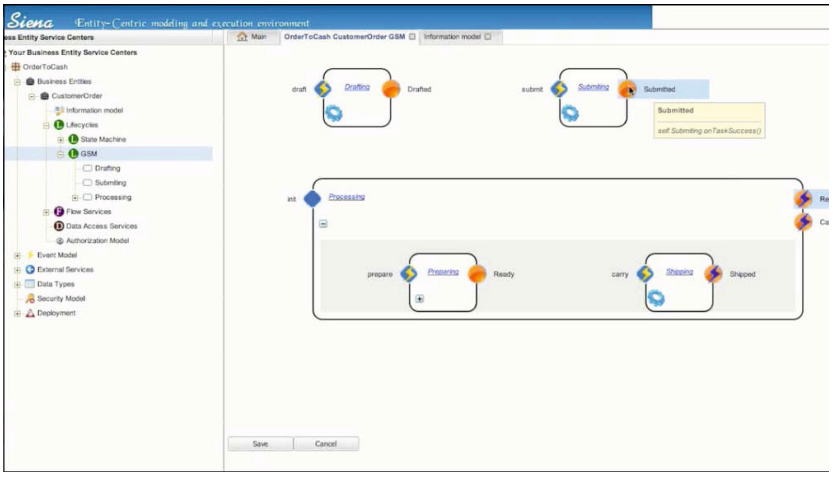

Fig. 1. Illustration of Barcelona graphical schema editor

Completion of a stage execution is controlled by rules-based *milestones*, indicated by orange disks on the right-hand boundaries of stages. These, too, may be triggered by external or internal events.

The Barcelona engine can support GSM schemas with multiple artifact types, and can support large numbers of artifact instances. In typical applications the instances are created programmatically, and progress through their lifecycles through responses to automatic and manual steps. However, to illustrate the run-time operation of Barcelona, and to inspect the status of artifact instances, we use the run-time editor, illustrated in Figure 2. That figure shows a single instance of the `CustomerOrder` artifact type, at some point in its progression through the business. The central area in the screen shows the data currently held in the information model of this order. Both scalar and nexted values are supported. On the right hand side is a schematic representation of the status of stage executions for this artifact instance. This indicates that both the *Drafting* and *Submitting* stage have executed once, and that the *Processing* stage is currently executing, with substage *Preparing* and inside that subsubstage *Collecting* all executing, and that one occurrence of *ResearchingOrdering* has completed inside. This representation supports hierarchy, to reflect the nesting of stages. If a stage is executed multiple times, then it is listed multiple times in this listing.

The gray buttons at the upper right of the screen are used to simulate the arrival of external events, so that the user can manually progress the artifact instance through its lifecycle, for testing purposes.

The high-level architecture of Barcelona is shown in Figure 3. The following summarizes the major features of the respective components.

Execution engine: Provides support for (1) management of business artifact instances; instances stored in the relational database; (2) management of artifact lifecycles & interactions (by means of service invocations or event subscriptions); (3) interaction

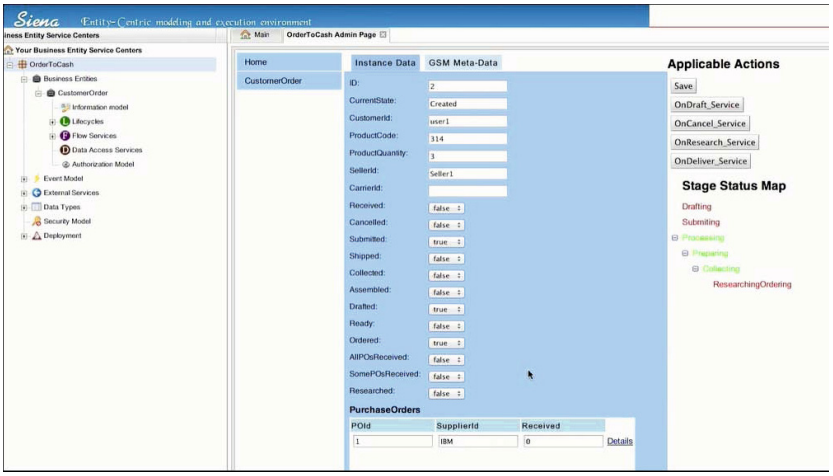

Fig. 2. Illustration of Barcelona runtime

with external environment via REST and WSDL service APIs; (4) management of access control; (5) basic support for execution monitoring.

Solution Designer Editor: Provides functions for easy authoring of artifact-centric business processes, including (1) design of information models (nested relational structures); (2) design of artifact lifecycles, supporting both GSM and Finite-State-Machine lifecycle models (in GSM supporting two condition languages, one based on JEXL, and the other one based on extended OCL); (3) design of data services: serve for providing access to arbitrary data queries over information model of artifact instances; (4) design of flow services, which are complex flows based on flowcharts.

Default Runtime GUI: Supports execution of artifact-centric business processes including (1) step through BOM executions; (2) inspection of the artifact instances states (information model, lifecycle model). The tool is intended for testing, debugging and support of rapid prototyping. In contrast, the Solution Builder application [10] incorporates a special-purpose UI working on top of the Barcelona runtime engine.

Additional Components: There are repositories for (1) Artifact Schemas, (2) Artifact Instances, and (3) External Service specifications (which link to external REST and WSDL services)

As noted above, Barcelona forms one component of the ArtiFact system. The other components are Siena (which supports Finite-State-Machine lifecycles for artifacts), and the ACSI Interoperation Hub (which supports interoperation between enterprises). The ArtiFact system originated with the Siena system, developed at IBM Research starting in 2006. The Barcelona component was added in the early 2010s, with several extensions in the past 2 years. A substantial portion of the Barcelona component, and all of the Interopration Hub, were developed as part of the EU-funded Artifact-Centric Services Interoperation (ACSI) project [9]. The ACSI project studied the artifact-centric

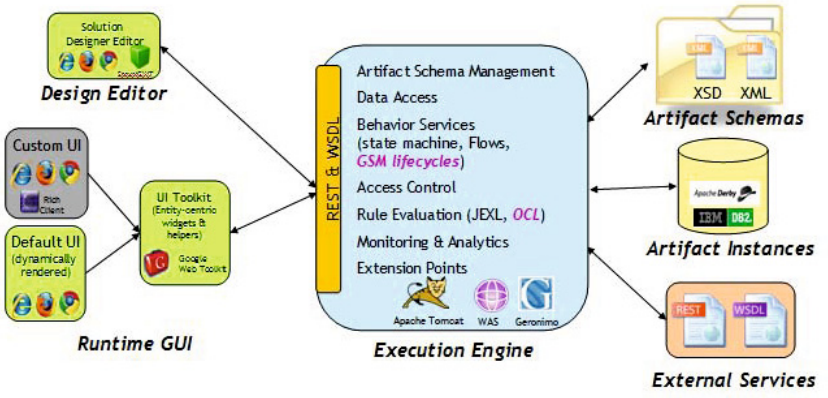

Fig. 3. High-level Barcelona architecture

approach from a variety of perspectives, including conceptual modeling, verification, process mining, systems, services interoperation, access controls, and practical applications; GSM was a starting point for most of these investigations.

The GSM model was used as a basis for the recently released OMG CMMN standard. While there are significant differences between GSM and CMMN, the Barcelona component of the ArtiFact system can be used as a lightweight, open-source tool for studying the basic approach underlying modern case management.

A preliminary version of Barcelona was demonstrated at the BPM 2011 conference. The most important advances since that time include (a) the development of a graphical design editor, (b) the development of the OCL-based condition language (which brings the power of full first-order logic), and (c) the implementation of the operational semantics as described in [3]. (That semantics has certain desirable properties, including conformance to intuitive principles and equivalence to a fixpoint characterization.)

A demo video is available at this URL: `http://goo.gl/YAvxd`.

References

1. Boaz, D., Limonad, L., Gupta, M.: BizArtifact: Artifact-centric Business Process Management (June 2013), `http://sourceforge.net/projects/bizartifact/`
2. Damaggio, E., Hull, R., Vaculín, R.: On the equivalence of incremental and fixpoint semantics for business artifacts with guard-stage-milestone lifecycles
3. Hull, R., et al.: Introducing the guard-stage-milestone approach for specifying business entity lifecycles. In: Bravetti, M., Bultan, T. (eds.) WS-FM 2010. LNCS, vol. 6551, pp. 1–24. Springer, Heidelberg (2011)
4. Hull, R., et al.: Business artifacts with guard-stage-milestone lifecycles: Managing artifact interactions with conditions and events. In: ACM Intl. Conf. on Distributed Event-Based Systems, DEBS (2011)
5. Kumaran, S., Nandi, P., Terry Heath III, F.F., Bhaskaran, K., Das, R.: Adoc-oriented programming. In: SAINT, pp. 334–343 (2003)

6. Marin, M., Hull, R., Vaculín, R.: Data centric BPM and the emerging case management standard: A short survey. In: La Rosa, M., Soffer, P. (eds.) BPM 2012 Workshops. LNBIP, vol. 132, pp. 24–30. Springer, Heidelberg (2013)
7. Nigam, A., Caswell, N.S.: Business Artifacts: An Approach to Operational Specification. IBM Systems Journal 42(3) (2003)
8. Vaculín, R., Hull, R., Heath, T., Cochran, C., Nigam, A., Sukaviriya, P.: Declarative business artifact centric modeling of decision and knowledge intensive business processes. In: The Fifteenth IEEE International Enterprise Computing Conference (EDOC 2011), pp. 151–160. IEEE Computer Society (2011)