ICWE2015 Rapid Mashup Challenge: Extended Techniques for Flexible Modeling and Execution of Data Mashups

Pascal Hirmer, Bernhard Mitschang

Institute of Parallel and Distributed Systems,
Universität Stuttgart,
pascal.hirmer@ipvs.uni-stuttgart.de
http://www.ipvs.uni-stuttgart.de

Abstract. Today, a multitude of highly-connected applications and information systems hold, consume and produce huge amounts of heterogeneous data. The overall amount of data is even expected to dramatically increase in the future. In order to conduct, e.g., data analysis, visualizations or other value-adding scenarios, it is necessary to integrate specific, relevant parts of data into a common source. Due to oftentimes changing environments and dynamic requests, this integration has to support ad-hoc and flexible data processing capabilities. Furthermore, an iterative and explorative trial-and-error integration based on different data sources has to be possible. To cope with these requirements, several data mashup platforms have been developed in the past. However, existing solutions oftentimes have limitations regarding the mentioned requirements. For this challenge, we present an approach that tackles them (i) by introducing a domain-specific model to enable usage by non-IT experts, (ii) by the introduction of transformation patterns to enable decoupling from implementation details, and (iii) by a high degree of flexibility and extensibility that enables the integration of heterogeneous data as well as dynamic (un-)tethering of data sources. An existing prototypical implementation of the approach serves as basis for this challenge.

Keywords: Data Mashups, Domain-specific, Patterns, Data Flow

1 Goals

In previous work [4], we compared and tested several data mashup solutions regarding different aspects. We found out that there are several limitations regarding (i) usability by non IT-experts, (ii) coping with different requirements of various scenarios, i.e., universality of the implementation, and (iii) handling of heterogeneous – especially unstructured – data. Furthermore, the handling of *"live data"*, e.g., sensor data streams, is only supported by tailor-made, non-generic solutions. We tackle these limitations by an approach that (i) enables usage by non IT-experts by exclusively using means specific to their domain, (ii) uses patterns to abstract from implementation details, which enables using a

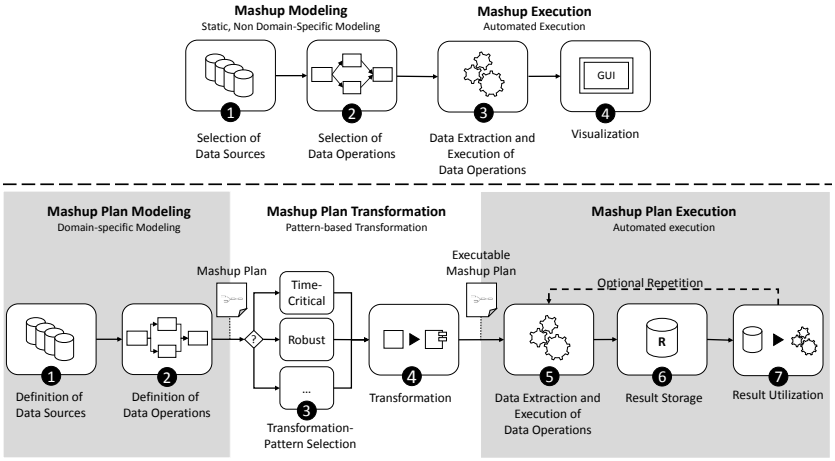

Fig. 1. Traditional Data Mashup (top) and our Extended Approach (bottom)

tailor-made implementation specific to a respective scenario, (iii) facilitates the integration of various kinds of data through a generic, extensible approach, and (iv) enables dynamic (un-)tethering of data sources. To prove the feasibility of our approach, we created a prototype that implements the introduced concepts. This prototype will be used as basis for this challenge.

2 Proposed Solution

In contrast to *"traditional"* data mashups as shown on the top of Figure 1, our extended approach (depicted at the bottom of Figure 1) offers an increased flexibility for modeling and executing data mashups. In the first two steps of our extended approach, a domain-expert defines the data sources as well as the data operations to be executed for the mashup, using a domain-specific graph model called *Mashup Plan*. As usual in graphical data mashup modeling, this model contains data sources and data operations as nodes and the data flow between them as edges. Technical details about the data sources are abstracted using so called business artifacts [1] [6] that are specific to the user's domain. Using mapping approaches [8], they can be bound to various kinds of underlying data structures ranging from traditional relational databases to unstructured text. Furthermore, data operations can be defined in the Mashup Plan using modeling patterns (based on [7]) to abstract from technical details. In step 3, *transformation patterns* are selected to choose an implementation suitable for the respective scenario (e.g., time critical or robust). The pattern selection affects the execution language and engine that is chosen to execute the data mashup. In step 4, the Mashup Plan is transformed into an executable model depending on the transformation pattern. In step 5, this executable model is executed by a suitable engine. Using the results of previous work [3], we are able to deploy

and install the necessary components for the execution automatically in a cloud environment. After execution, the integrated result is stored into a data store (step 6) and is available, e.g., for analysis or visualization (step 7).

3 Level of Maturity

We currently implemented a prototype of the presented approach and used it in two different use case scenarios. In the first scenario, sensor data is integrated and processed to compute high-level situations in smart environments. For this implementation, we slightly adopted our prototype to support the processing of sensor data. The detailed results are described in [2]. The second use case implements a data mashup for exception escalation in advanced manufacturing environments, which is described in [5]. At the moment, the prototype is tailor-made for these use cases and does not offer a generic solution. Furthermore, we are limited to two patterns as well as a few data sources and data operations.

4 Feature Checklist

- **Mashup Type:** Data mashups
- **Component Type:** Data components
- **Runtime Location:** Both Client and Server
- **Integration Logic:** Orchestrated integration
- **Instantiation Lifecycle:** Stateless
- **Targeted End-User:** Non Programmers
- **Automation Degree:** Semi-automation
- **Liveness Level:** Level 3
- **Interaction Technique:** Visual Language (Iconic)
- **Online User Community:** None

5 Demo

We implemented a web-based, graphical modeling service based on Java, JavaScript and the JavaScript library AlloyUI[1], which is hosted using the platform-as-a-service provider IBM BlueMix[2] (cf. Figure 2 on the top). As data sources, we support structured MySQL[3] databases, unstructured text input feeds and sensor data streams. As data operations, we currently support data combination (e.g., join, aggregate) and data selection (e.g., filter). By using extendable repositories, additional data sources and data operations can be added easily. Furthermore, the transformation patterns *"Robust Mashup"* or *"Time-Critical Mashup"* can be selected at transformation level. After modeling, the transformation and execution can be invoked through the modeler's UI. The modeling and transformation results are shown in Figure 2. In our demo, we will model an integration scenario using our graphical editor and we will execute the mashup with different patterns.

[1] http://www.alloyui.com/ [2] http://www.bluemix.net [3] http://www.mysql.de/

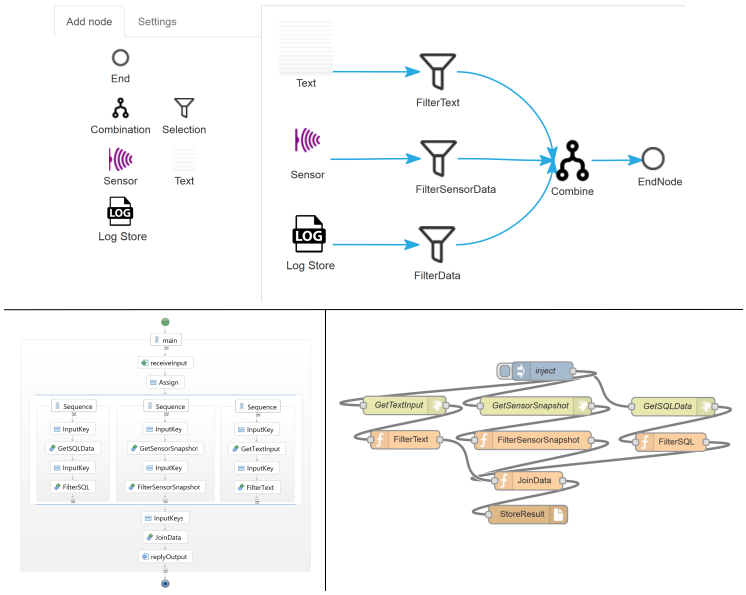

Fig. 2. Screenshot of the Mashup Plan Modeling (top) and two Transformation Pattern Implementations (bottom left: robust/BPEL; bottom right: time-critical/Node-RED)

References

1. Cohn, D., et al.: Business artifacts: A Data-centric Approach to Modeling Business Operations and Processes. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering (2009)
2. Hirmer, P., Wieland, M., Schwarz, H., Mitschang, B., Breitenbücher, U., Leymann, F.: SitRS - A Situation Recognition Service based on Modeling and Executing Situation Templates. In: In Proceedings of the 9th Symposium and Summer School On Service-Oriented Computing (SUMMERSOC15) (2015)
3. Hirmer, P., Breitenbücher, U., Binz, T., Leymann, F.: Automatic Topology Completion of TOSCA-based Cloud Applications. In: Proceedings des CloudCycle14 Workshops auf der 44. Jahrestagung der Gesellschaft für Informatik e.V. (GI) (2014)
4. Hirmer, P., Reimann, P., Wieland, M., Mitschang, B.: Extended Techniques for Flexible Modeling and Execution of Data Mashups (April 2015), submitted
5. Kassner, L., Mitschang, B.: MaXCept–Decision Support in Exception Handling through Unstructured Data Integration in the Production Context. In Proceedings of the 48th Hawaii International Conference on System Sciences (2015)
6. Künzle, V., et al.: PHILharmonicFlows: Towards a Framework for Object-aware Process Management. Journal of Software Maintenance and Evolution: Research and Practice (2011)
7. Reimann, P., et al.: A Pattern Approach to Conquer the Data Complexity in Simulation Workflow Design. In: Proceedings of the 22nd International Conference on Cooperative Information Systems (CoopIS 2014), Amantea, Italy (2014)
8. Sun, Y., et al.: Modeling Data for Business Processes. In: Proceedings of the 30th IEEE International Conference on Data Engineering (ICDE), Chicago, USA (2014)