Mashups: Behavior in Context(s)

Emilian Pascalau

Hasso Plattner Institute at University of Potsdam,
Germany,
emilian.pascalau@hpi.uni-potsdam.de

Abstract. The World Wide Web (WWW) has left behind the dot-com bubble and changed into something new. The move to the Internet as a platform and the shift from transaction-based Web pages to interaction-based ones opened the way to a whole new environment. Future Internet is a generative environment that fosters innovation, through an advance of technologies and a shift in how people perceive, use and interact with it. Nowadays the abilities to create new information have far exceeded the abilities to manage it. There exist a huge amount of data and potential that is still unused and undiscovered.
Mashups are a new paradigm emerged from Web 2.0 that tries to empower users with some sort of freedom to tackle this huge amount of potential provided nowadays by the web. Current approaches however are still restrictive in many ways e.g. are data oriented, and platform dependent. Hence this paper introduces a new perspective on mashups. Here a mashup is seen as a plan that a user and an engine need to follow in order to achieve a desired goal. As such a mashup comprises contextual information and the necessary behavior related to the context(s) described, in order to fulfill desired goal. Subsequently a mashup is defined here as behavior in context(s).

1 Introduction

Web 2.0 [12] is *no longer a bleeding edge but rather a leading edge now* [15] and has become integral part of life and business. Participation is one aspect which pushes forward Web 2.0 [3]. In the last five years Web 2.0 technologies (i.e. social networking sites, blogs, wikis) have spread widely among consumers. Sites such as Facebook attract more that 100 million visitors a month[3]. There is a shift from processes towards users. Users want their problems and their requirements to be taken into account; they want to be part of the conversation. Continuously changing business models do not fit anymore the old and stiff approaches. Processes must be in accordance with the reality, and reality means people. It means that processes and system behavior have to be in accordance with what users require and with their needs. This issue is also underlined by process mining [16] approaches that look at event logs and see that the processes that actually get executed are different compared to the original blueprints. Companies need to change to what customers/users actually do.

Harnessing collective intelligence, wisdom of the crowds, easy consumption, web as innovation platform, context are requirements that need to be tackled to allow people to use their imagination without too much restriction in order to fulfill by themselves their goals. As argued in [13] the Web is more than just data, is about knowledge, context, behavior and most important is about people.

This paper proposes a new definition for the mashup concept from a user's perspective. Thus a mashup is behavior in context(s). This particular perspective is a high level one addressing mashups at a conceptual level opposed to current approaches that are mostly application and technology oriented (see for instance Yahoo Widgets[1]). The framework discussed here allows users to model a mashup as a map containing context(s) and behavior description required to fulfill a specific goal. In consequence this approach implies business rules, business processes, business concepts and vocabularies that describe the businesses and users' goal themselves rather than a possible IT system that might support it. The framework is formalized using the Unified Modeling Language (UML).

The reason for such a framework is manifold. To name just a few: (1) users are able to define their own applications in order to fulfill their needs; (2) because the framework unifies several paradigms (behavior, context etc) reasoning can be performed in an unified over all these; (3) models described can be extended, modified and maintained in an unified way as well; (4) companies can learn their customers' needs as these mashups expose behavior as well as the context(s) in which exposed behavior is performed; (5) these mashups can act very easily as prototypes for possible future major implementations; (6) statistics can be provided about the usage of mashups and about the system(s) involved in relationship with social networking platforms and so forth.

2 Conferences Calendar Example

The Conferences Calendar example has been first introduced in [13]. Such a calendar is user specific since for example some users might be interested in web related conferences, others in semantic web, others in rules or business processes conferences. Contextual information is mashed up to fulfill a user goal; hence specific information about conferences is stored in a calendar context. At least two services are required: one that deals with conferences and one that offers a calendar. From a technological point of view these services might not be compatible with each other e.g. might not use the same definition language.

For scientists in the filed of IT the DbWorld Mailing List is the well known place where they can search for an IT conference. A series of information are provided here, but most important are the subject, deadline and the web page of the event published. From a technical perspective DBWorld does not provide an API to allow programmatically access and interrogation of the service. In consequence with respect to current mashups approaches this service is useless. On the other hand Google Calendar is one of the most known Google Apps

[1] `http://manual.widgets.yahoo.com`

services and the service that provides the calendar context for the use case. The information of interest for a calendar is the title of the event, the date and description of an event. This information is found also in a Google Calendar. Opposed to the DbWorld service, Google provides for this service beside the regular web page representation also an API to access the contents.

The usual way to achieve this goal, of having conferences stored in Google Calendar by their deadline is manually: (1) the user is required to maintain two open tabs in the browser; (2) even though there might be several entries that comply with a search term, the user must deal with the events one by one as DBWorld does not provide built in search functionality; (3) the user has to move between the open tabs several times, in order to store only one event in the calendar, since just one piece of information can be copied and pasted at a time (e.g. the title of the event). An important aspect is that both services as well as users interact with each other.

3 The Framework

As argued the framework discussed here it is defined from a user perspective and is formalized using the de facto standard modeling language UML.

Next subsections discuss the main concepts of the framework required to define the *mashup* concept.

3.1 Concept

Ackoff [1] defines an abstract system as one all of whose elements are concepts. Because the framework has to deal with a high degree of generality it has at the core the abstract notion of *concept*. A concept is a cognitive unit of meaning - an abstract idea or a mental symbol.

Languages, number systems are examples of abstract systems. Numbers are concepts but the symbols that represent them, numerals are physical things [1]. Humans deal both with conceptualizations as well as with physical things. However the reasoning process involves only conceptualizations.

Although many of the ontological approaches (see for instance OWL [6]) use as the upper level entity the *thing* notion for the framework discussed here the notion of *concept* is at the top.

The OMG specification for Semantics of Business Vocabulary and Business Rules Specification (SBVR) [10] uses as top entity the *concept* notion. Definition 1 is the SBVR definition for the concept notion.

Definition 1. ***Concept.*** *Unit of knowledge created by a unique combination of characteristics.*

There are two ways to recognize entities. Basically in software engineering when dealing with typed languages, entities are recognized by their types (class name). The other way around is based on a set of characteristics. Take for instance a *car*. Stating the concept's name *car* someone will be able to tell you

the characteristics of a car, that it has for wheels, that it has an engine etc. Nevertheless stating the characteristics of the concept that it has 4 wheels and an engine, the answer will be a *car*. The reasoning architecture [13] that uses this framework tackles both approaches.

3.2 Context

The notion of context is of interest in cognitive psychology, in linguistics and computer sciences. In the field of computer sciences notions of context have appeared in several areas such as artificial intelligence, machine learning, data bases and software development. In some of these areas the notion of context appears in the form of views, aspects, information for concept classification, means to partition knowledge in manageable sets, or as an abstraction mechanism to partition information into possibly overlapping parts [2].

Dey in [5] defines context as *any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.* Similarly for Coutaz et al. [4] the context is a *structured and unified view of the world in which the system operates.*

Context is under permanent change, is episodic, personal and hence subjective interpretations and experiences of the communicative context [17],[5]. Analyti et al. discuss in [2] a general framework to harness the notion of context in conceptual modeling. A full mathematical apparatus has been defined to tackle issues such as containment and relationships between contexts. According to them "context in an information base can be seen as a higher-order conceptual entity that groups together other conceptual entities on which we want to focus" [2]. More precisely a context is a set of objects within which each object is associated with a set of names.

For the framework discussed here the notion of context adheres to the mathematical apparatus defined in [2], however here a context is a set of concepts not objects. Nonetheless our definition is compliant with Analyti et al. definition.

Basically a context is a set of concepts (concepts according to Definition 1).

Definition 2. ***Context.*** *A context consists of a context identifier and a set of concepts identifiers.*

Recalling the simple mechanism that has been discussed in subsection 3.1, a context is identified by recursively identifying all the constituent concepts.

The notion of context supports a series of features as they have been defined in [2]: (1) concept sharing or overlapping contexts; (2) context-dependent concept names; (3) context dependent references; (4) context sharing; (5) context-dependent reachability; (6) synonyms, homonyms, anonyms.

Beside these features the notion of context is enhanced also with attribution, generalization and classification.

3.3 Behavior

For this particular approach behavior comprises rules and processes. It is described by users in relationship with related context(s).

When dealing with human like behavior a single system (mind) produces all aspects of behavior [8]. *It is one mind that minds them all* [8]. Even if such a system has parts, modules, components or whatever they mesh together to produce behavior.

The mind is the control unit that guides the behaving organism in its complex interactions with the dynamic real world [8]. Both the behaving organism as well as the environment behaves through time with a series of interactions between them. [8] continues by stating that these transactions or interactions *are embedded in a sequence such that each becomes part of the context within which further action follow.*

Newell underlines a set of requirements that behavior must comply with [8]: (1) it has to be flexible as a function of the environment; (2) flexible in such a sense that it can deal with goals; (3) real time; (4) according with the context.

Behavior of an entity is the set of events, actions and messages that that entity produces. Behavior is conditioned by the context and it is expressed either as rules or as processes. Hence an event is any observable occurrence of a phenomena. An action as stated in [7] is represented with the keyword `do` and is represented as *function from a time and an action, to the time at the end of the action.* Acokff defines behavior in [1] in terms of system as a system change which initiates other events implying that behavior consists of events *whose consequences are of interest* [1].

According to the PRR [9] specification a production rule is a statement that specifies the execution of one or more actions in the case that its conditions are satisfied. An Event Condition Action (ECA) rule is a production rule triggered by an event. Thus the form of an ECA rule is: `on [events] if [conditions] then [action-list]`.

While Weske argues in [18] that a *"business process consists of a set of activities that are performed in coordination in an organizational and technical environment"*, Ackoff on the other hand defines a process as a sequence of behavior that constitutes a system and has a goal producing function [1].

Hence behavior is goal oriented, is context(s) related and is expressed as rules and processes.

3.4 Mashup

This section unifies and puts together previously discussed aspects in order to define the mashup concept. In consequence a mashup is a map which describes the context(s) and related behavior that a user needs to do in order to achieve a desired goal. Such a mashup is defined from a user perspective. Coutaz et al.' [4] view of *context-as-process* is related to the idea I discuss here. However their approach is not from a user perspective but rather from an IT system perspective. Nonetheless at least two issues are addressed by having context

related to behavior. First *context-as-process* view allows for greater flexibility than *context-as-state* as utility and usability are derived from information exchange and interaction [4], [14]. Secondly there is no mismatch risk between system's interaction model and the mental model that a user might have about the system[4]. This approach uses actually as interaction model for the system the one that a user defines as a mashup. Moreover as discussed in [14] context provides meaning to processes. For example one could deal with a sell/buy process, a very generic one. But whenever contextual information is added, the *meaning* of a process could be totaly different, as there is a big difference between selling tomatoes and selling e.g. chemical products. To support even more this idea SBVR specification [10] states that a body of shared meaning that a community has is represented in concepts, fact types (relationships between concepts) and business rules (constraints on concepts and fact types).

Definition 3 (mashup). ***Mashup.*** *A mashup is a set of contexts and behavior. Behavior consists of rules and processes.*

Figures 1, 2, 3, 4 formalize the framework. These models comply with the definitions previously discussed.

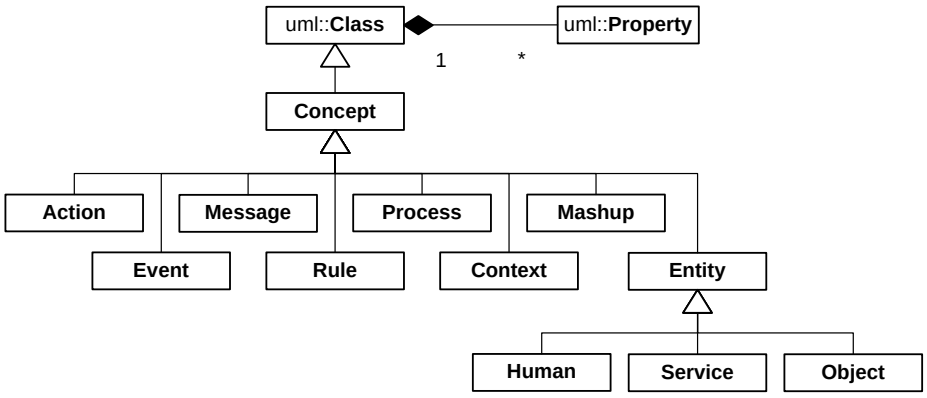

Fig. 1. Concepts

Based on Definition 1 a concept is a unit of knowledge created by a unique combination of characteristics. Thus as depicted in Figure 1 every element of the framework is a concept. In addition although not visually represented a `Concept` is also a `Concept`. In this way the reasoning process can involve any of the concepts defined in a unified way.

Figure 2 depicts the general framework. Hence the `Mashup` concept contains 1 or more `Contexts`. Further a `Mashup` can contain `Processes`, `Rules` or a combination of those two. A `Context` is basically a collection of `Concepts`. In addition a `Context` could have subcontexts. A `Context` refers to an `Entity`.

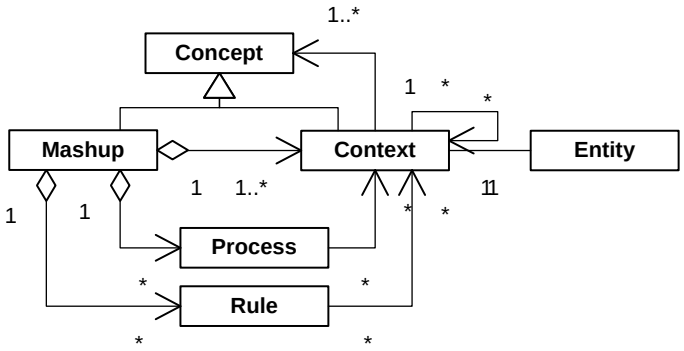

Fig. 2. Mashup Concept

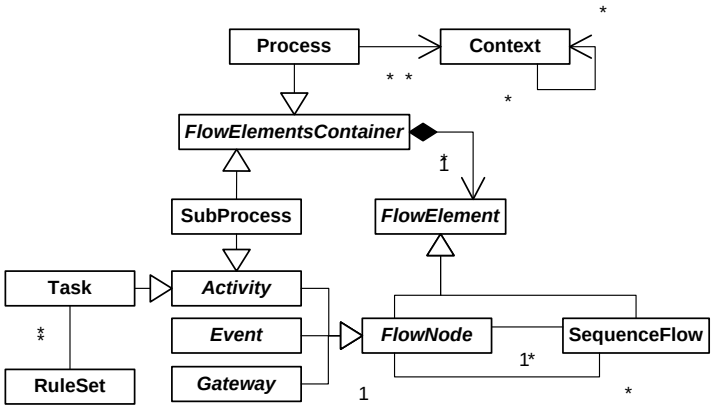

Fig. 3. Business Process, based on BPMN 2.0 specification [11]

The Process Concept is further expanded in Figure 3. The definition is based on the BPMN 2.0 specification. Thus a process is a `FlowContainer` and contains `FlowElements` and `SequenceFlows`. Furthermore a `FlowElement` is either an `Activity`, a `Gateway` or an `Event`. An `Activity` is subclassed by a `SubProcess`, meaning that a `Process` might have subprocesses, and by a `Task`. A `Task` is an atomic `Activity`. However this model introduces the following relationships, which were not previously contained by the BPMN 2.0 specification: the execution of a Task can mean the execution of RuleSets; `Processes` are related to `Contexts`. This particular relationship can provide as discussed in [14] meaning to processes.

The model depicted in Figure 4 is compliant with the OMG PRR specification [9] and is the basic model for a Rule. A Rule similar to a Process is related to a Context. It can be triggered by an Event, in the case of an Event Condition Action (ECA) Rule. It can be conditioned by a set of conditions. Conditions concern Concepts. Actions are the result of rule execution. With respect to execution beside the regular rule conflict resolution mechanism the engine using

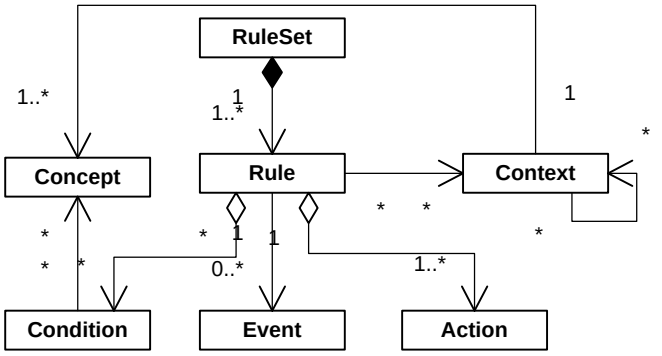

Fig. 4. Rules

this framework uses processes to order the execution of actions in relationship with events.

4 Using the Framework

Recall the example discussed in Section 2. Several contexts are involved in this particular mashup: the DBWorld context, the Google Calendar context and the calendconf context. Figure 5 depicts all the involved contexts mashed together. According to Definition 2 a context contains a set of concepts. In addition as argued in Section 3.2 identifying a context means identifying all the constituent concepts. Let's take for instance the Google Calendar context. This one refers to the Google Calendar Entity. This entity is uniquely identified by its URL. The context contains a `Create Event` button. While this concept in relationship with the entity is enough for one user, someone else could use a different set of concepts to identify the same context. Furthermore `Create Event` button is identified by a set of characteristics. The most evident one is the name: `Create Event`. Figure 6 depicts an excerpt of the framework instantiation.

I was arguing that behavior is in a strong relationship with the context. The most simple example: to be able to create a calendar event in Google Calendar a user needs to click on the `Create Event` button. A more complex one (see Figure 7) is the process of searching for a particular conference in DbWorld. From DbWorld the subject, deadline and web page are the concepts of interest. With respect to behavior in this context, one user could be interested both in the subject and deadline when searching for a conference. On the other hand another one could be interested only in the subject.

While the framework allows reasoning over all the constitutes elements similar to human cognition and as such empowering users with the ability to define behavior in fine details an user friendly modeling platform for the non technical users is desired. Widgets based, pipes based platforms have proven to be easy to use. Similar to those approaches a visual modeling platform for the framework

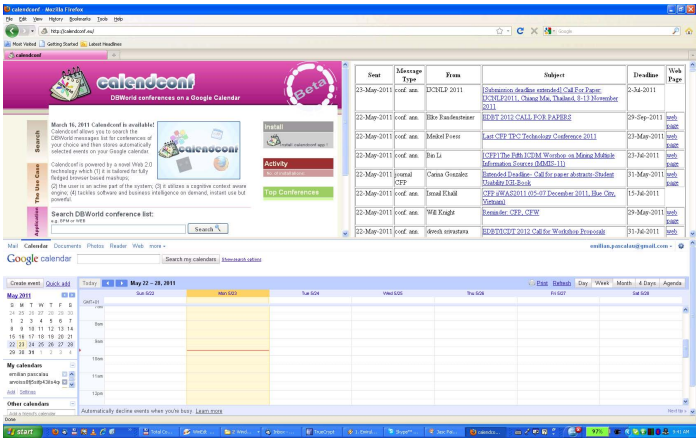

Fig. 5. Calendconf Mashup

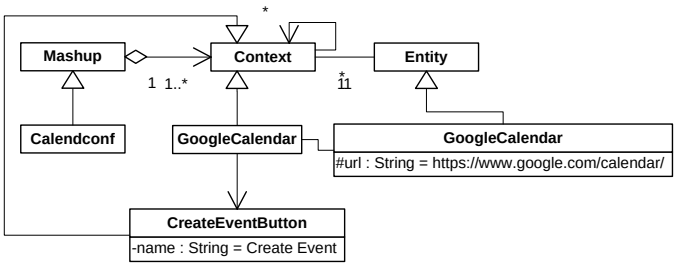

Fig. 6. Framework Instantiation - an excerpt

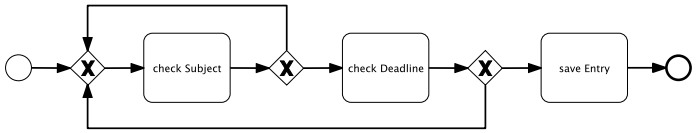

Fig. 7. Search for a Conference Process

discussed here is under development. Currently mashups are defined declaratively using JSON notation. Nonetheless the running version of the example discussed here can be accessed at `http://calendconf.eu`.

5 Conclusions

This paper discussed a new perspective for the mashup concept. While the Web 2.0 mashup paradigm has been mostly currently addressed from a technical perspective and strongly application oriented, the framework formalized here concerns a high level perspective and defines a mashup as behavior in context(s).

Further improvements of the framework concerns reuse of mashups with an emphasis on inheritance. As argued in [14] this is not a straight forward process as here behavior is defined using UML, hence as static constructs. In consequence UML class inheritance can not be used as it is but special types of inheritance mechanisms are required.

References

1. R.L. Ackoff. Towards a System of System Concepts. *Management Science*, 17(11), 1971.
2. Anastasia Analyti, Manos Theodorakis, Nicolas Spyratos, and Panos Constantopoulos. Contextualization as an independent abstraction mechanism for conceptual modeling . *Information Systems*, 32(1):24–60, 2007.
3. Michael Chul, Andy Miller, and Roger P. Roberts. Six Ways to make Web 2.0 work. *Business Technology, The McKinsey Quaterly*, pages 1–6, February 2009.
4. Joelle Coutaz, James L. Crowley, Simon Dobson, and David Garlan. Context is key. *Communications of the ACM*, 48(3):49–53, 2005.
5. Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. Technical Report GIT-GVU-99-22, Georgia Institute of Technology, 1999.
6. W3C OWL Working Group. OWL 2 Web Ontology Language. http://www.w3.org/TR/owl2-overview/, 2009.
7. Patrick J. Hayes. A Catalog of Temporal Theories. Technical Report UIUC-BI-AI-96-01, University of Illinois, 1996.
8. Allen Newell. *Unified Theories of Cognition*. Harvard University Press, 1994.
9. OMG. Production Rule Representation (PRR), Beta 1. Technical report, OMG, November 2007.
10. OMG. Semantics of Business Vocabulary and Business Rules Specification. http://www.omg.org/spec/SBVR/, January 2008.
11. OMG. Business Process Model and Notation (BPMN). FTF Beta 1 for Version 2.0. http://www.omg.org/spec/BPMN/2.0, August 2009.
12. T. O'Reilly. What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. *Communications and Startegies*, 1st quarter(65):17, 2007.
13. Emilian Pascalau. Towards TomTom like systems for the web: a novel architecture for browser-based mashups. In *Proceedings of the 2nd International Workshop on Business intelligencE and the WEB (BEWEB11)*, pages 44–47. ACM New York, NY, USA, 2011.
14. Emilian Pascalau and Clemens Rath. Managing Business Process Variants at eBay. In Jan Mendling and Mathias Weske, editors, *Proceedings of the 2nd International Workshop on BPMN, BPMN2010*. Springer, 2010.
15. The Economist Intelligence Unit. Serious business. Web 2.0 goes corporate. *The Economist*, pages 1–20, 2007.
16. W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003. `http://is.tm.tue.nl/staff/wvdaalst/publications/p206.pdf`.
17. Teun A van Dijk. Cognitive Context Models and Discourse. *Proceedings and Debates of the 102d Congress*, 137(84):189–226, June 1991.
18. M. Weske. *Business Process Management: Concepts, Languages, Architectures* . Springer-Verlag Berlin Heidelberg, 2007.