Data-Centric Systems and Applications

Florian Daniel Maristella Matera

Mashups

Concepts, Models and Architectures

Chapter 5 Mashup Components

Figures





Fig. 5.1 The most basic component model consists of either one operation that can be invoked or of a piece of UI that can be rendered, or of both.



Fig. 5.2 Conceptual model of a web service consisting of a set of four different types of message-based operations.



Fig. 5.3 Model of a RESTful web service delivering representations of and manipulating resources in response to standard HTTP requests.



Fig. 5.4 Model of a typical JavaScript API. The gray shaded entity is not part of the actual component model; it tells which artifacts the developer must deploy, in order to be able to use the library. Pure logic components do not have a user interface.

```
<rss version="2.0">
<channel>
  <title>Liftoff News</title>
 <link>http://liftoff.msfc.nasa.gov/</link>
 <description>Liftoff to Space Exploration.</description>
  . . .
  <item>
   <title>Star City</title>
   <link>http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp</link>
    <description>
     How do Americans get ready to work with Russians aboard the
     International Space Station? They take a crash course in culture,
     language and protocol at Russia's
     <a href="http://howe.iki.rssi.ru/GCTC/qctc_e.htm">Star City</a>.
    </description>
   <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
   <guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
```

Fi_{ (ht



Fig. 5.6 The model of an RSS feed is simplified version of the model we presented for RESTful web services (see Figure 5.3).



Fig. 5.7 The model of an Atom feed including the features specified in the Atom Publishing Protocol is that of a RESTful web service with schema and media type restrictions.

Fig. 5.8 Screen shot of the Dapper content extraction tool in action: interactive extraction of data from the New York Times web site and publication as RSS feed.

```
<div class="vcard">
  <div class="fn n">
    <span class="family-name">Daniel</span>
    <span class="given-name">Daniel</span>
    <span class="given-name">Daniel</span>
    <span class="given-name">Daniel</span>
    <span class="given-name">Florian</span>
    <div>
    <div class="given-name">Florian</div>
    <span class="org">University of Trento</div>
    <span class="org">University of Trento</div>
    <span class="street-address">Via Sommarive 5</span>,
    <span class="postal-code">38123</span>
    <span class="locality">Povo (TN)</span>,
    <span class="locality">Povo (TN)</span>,
    <span class="country-name">Italy</span>
</div>
```

Fig. 5.9 An excerpt of HTML markup annotated with the hCard micro-format for annotating contact details. hCard uses class names to identify its elements.

Fig. 5.10 The Linking Open Data cloud diagram by Richard Cyganiak and Anja Jentzsch (http://lod-cloud.net/) visualizing the interrelations among the datasets that have been published in Linked Data format.

Fig. 5.11 Model of a Java portlet UI component according to JSR 168 [1].

Fig. 5.12 Extended model of a Java portlet according to JSR 286 [143] with a the possibility to share session and render parameters, to launch events, and to process events via the processEvent operation.

Fig. 5.13 Model of a W3C widget. In white the actual component model; in gray the artifacts of the component.

Fig. 5.14 Model of a W3C widget with inter-widget communication extension [277].

Fig. 5.15 The mashArt UI component model for UI components extracted from web applications annotated with the mashArt Event Annotation [96].

Fig. 5.16 Model of a real-time multimedia streaming component.

Fig. 5.17 Simplified model of a telco streaming component for audio/video conferencing with a shared session and multiple participants (streaming channels).

Fig. 5.18 Summary of the component characteristics that impact on the way components are to be used and, hence, on how they can be integrated with each other.