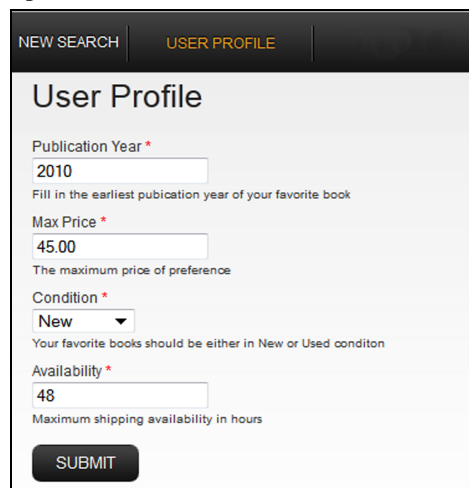


licating data into another syntax. However, in a mashup situation such as this, data flow into the application from many disparate sources. Following Linked Data principles [1], we therefore maintain the context of book data and their various offers, by assigning resolvable identifiers to them. These data can then be contributed to the LOD cloud and made available to other applications in RDF format.

- *Reasoner integration:* Most OWL 2 reasoners (like, Pellet, FaCT++ and Hermit) are traditionally deployed directly in-memory and interaction is performed by means of a java-based API. Although a PHP-to-Java bridge is available³, there are many reasons why one may want to keep reasoning services logically and/or physically separated [7]. Among them, the need for interoperability and independence from the actual programming language are of particular importance for integration with a CMS. In our implementation, we use OWLlink [8] as the reasoner communication protocol of choice and its implementation, the OWLlink API [9] that helps us deploy a true 3-tier architecture. OWLlink offers a consistent way of transmitting data to and receiving responses from the most popular Semantic Web reasoners, in a REST-like manner and over HTTP. Potential communication overhead that may be introduced with this approach can be alleviated by freeing up resources as a consequence of delegating computationally hard reasoning tasks to another tier [7]. Moreover, Drupal offers us generic function implementations that can be used to wrap and construct HTTP requests.

4 A Semantic Mashup over Drupal

When a user visits our app for the first time, he has to register by filling a form with his username and e-mail. An administrator then enables the account and a password is sent to the user at the specified mail address.



The image shows a web form titled "User Profile" with a dark header containing "NEW SEARCH" and "USER PROFILE" in yellow. The form fields are as follows:

- Publication Year ***: Text input with value "2010". Below it, a note says "Fill in the earliest publication year of your favorite book".
- Max Price ***: Text input with value "45.00". Below it, a note says "The maximum price of preference".
- Condition ***: A dropdown menu with "New" selected. Below it, a note says "Your favorite books should be either in New or Used condition".
- Availability ***: Text input with value "48". Below it, a note says "Maximum shipping availability in hours".

A "SUBMIT" button is located at the bottom of the form.

Fig. 2. Collecting user preferences.

³ <http://php-java-bridge.sourceforge.net/pjb/>

