

# Declarative RDF Models for Feature-based Targeting of Content to Multiple Devices

Vipul Kashyap

Telcordia Technologies, 445 South St, Morristown, NJ 07960, USA  
kashyap@research.telcordia.com

Leon Shklar

Information Architects, 70 Hudson St., Hoboken, NJ 07030, USA  
leon@ia.com

## ABSTRACT

We discuss an approach for targeted transformations of content to open sets of devices. The key element of this approach is the RDF Model that represents devices as sets of generic features and represents content resources as sets of components. The other key element is the RDF server utilizing this Model to support transformations across different devices and presentation channels. We illustrate our approach with an example that involves aggregating content and transforming it for display on wireless devices.

## Keywords

RDF, CC/PP, WML, XML

## 1. INTRODUCTION

The [Resource Description Framework \(RDF\)](#) is a metadata standard that was designed by the World Wide Web Consortium ([W3C](#)) to enable Web applications that depend on machine-understandable metadata, and to support interoperability between such applications. It targets a number of important areas that include dynamic syndication and personalization, mobile devices, resource discovery, intelligent agents, content rating, intellectual property rights, and privacy preferences. RDF-based syndication models can be naturally extended from targeting different content channels to enabling the exploding variety of wireless devices.

Emerging commercial products support multiple devices by building libraries of device-specific [XSLT](#) stylesheets to transcode XML content. Such stylesheets may be fairly efficient when compiled into Java bytecode (Sun distributes the [XSLT](#) compiler). The problem is in maintaining stylesheet libraries for the growing variety of new and evolving devices. The solution is to change the level of granularity of transformations and design them for individual features rather than devices. Using RDF models to implement device and user agent profiles, it is possible to design RDF servers that construct stylesheets by adapting and combining feature-based components, making it unnecessary to build and maintain ever-expanding libraries of complex device-specific stylesheets.

W3C, in coordination with the [Wireless Access Protocol \(WAP\) Forum](#) is developing the Composite Capabilities/ Preference Profiles (CC/PP) specification as the standard for setting device and user agent preferences. The upcoming RDF-based specification would allow defining a device by its screen size, keyboard (if any), display characteristics, etc. Next-generation RDF-based servers that target multiple devices would model device and user agent profiles, combine profile information with connection bandwidth, and use it to compose XSLT stylesheets from feature-specific components. An efficient server would optimize stylesheet construction by caching components and intermediate composites. For example, caching device-specific stylesheets that are constructed based on device profiles. Device-specific stylesheets may be combined with stylesheet components that are determined by the operating system, user agent software, connection bandwidth, and content components. We will illustrate this approach with an example

that focuses on specification and interpretation of device models. Issues related to caching will be discussed in future publications.

## 2. EXAMPLE

Our example is prototyped in the context of a wireless portal application that aggregates content from various Internet resources, and presents the content over a variety of devices. In this example, we present various pieces of information that are captured in the RDF Model and demonstrate how a RDF Server can interpret the model to display content on a wireless device. The techniques used in the example are general and can be applied to presenting content on the variety of non-wireless devices as well.

### 2.1 The RDF Model

The RDF Model in Figure 1 implements an employee resource portal. The content is aggregated from multiple sources and transformed according to its presentation targets. In our example, the resource is addressed using the following URI: <http://www.zzzzz.com/rdf/empinfo?name=John>.

#### 2.1.1 Content Properties

The content node represents the aggregation of personal information and email. Each component node, in turn, references its sources of content. In the case of John's information, they are specified by `ldap://zzzzz.com/?name=John` for LDAP and `imap://zzzzz.com/?email=john@zzzzz.com` for IMAP. The *has-markup* edge emanating from the content node references the XML DTD that imposes constraints on the aggregated content.

#### 2.1.2 Presentation Properties

The WML DTD node represents the presentation target and is associated with the presentation node via the *has-markup* edge. The role of the `Intermediate` DTD node is to enable composing the final transformation. The final transformation is composed of the generic WML transformation and the one that gets computed based on features of the target device. The generic transformation is specified using the *transform* edge from the XML DTD node to the `Intermediate` DTD node. The triple (*XML DTD*, *transform*, *Intermediate DTD*) is reified into a node representing this transformation. The *uses* edge emanating from the resulting node defines the generic XML-WML Transform. The sample transformation is static, it is defined at <http://www.zzzzz.com/transforms/XMLWML.xsl>.

The final transformation between the `Intermediate` DTD node and the WML DTD node is represented as another *transform* edge. The triple (*Intermediate DTD*, *transform*, *WML DTD*) is reified, and the *uses* edge from this resulting node identifies the `xyzTransform` node. The associated stylesheet is computed based on the device profile `xyzProfile` referenced by the *computed-from* edge. Features of `xyzProfile` are specified by the *type*, *CPU* and *screen-size* edges. It should be noted that the device features may only become known at request time, and hence the `xyzTransform` may have to be computed dynamically. The resulting transformation applied by

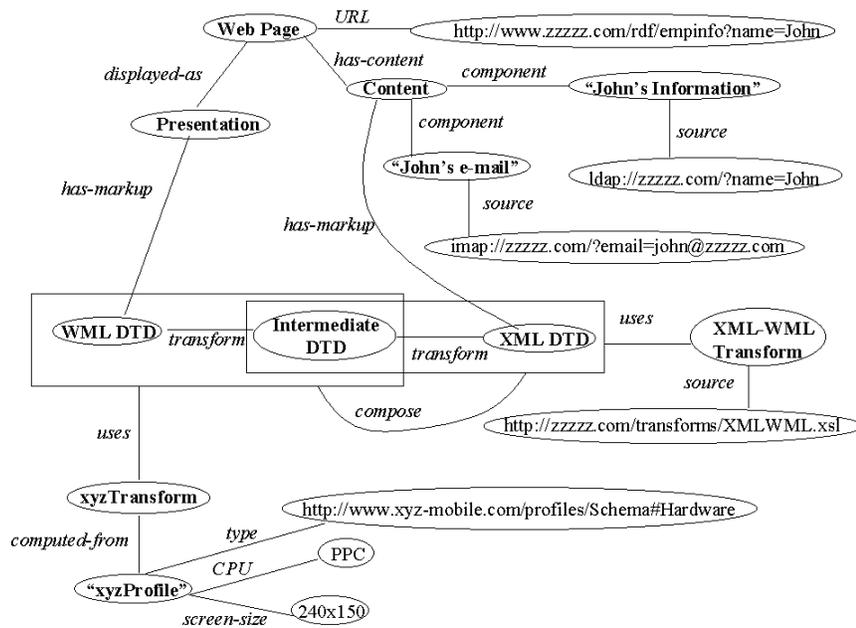


Figure 1. Sample RDF Model

the server is the composition of the static and feature-based transformations; it transforms aggregated XML format to WML format according to the device specifications.

## 2.2 The RDF Server

Our RDF Server interprets the RDF Model to generate and present content to wireless devices. The server traverses the graph to access the content and presentation nodes, and uses the retrieved properties to invoke appropriate parameterized `actors` that first aggregate content, and then compute and apply transformations that target proper wireless devices. Control issues related to the order in which the RDF model is traversed and the actors are invoked are handled by the server logic and are not represented in the model. A sample sequence of steps followed by the server is as follows:

- **Interpreting the content node:** The server identifies the various components, viz., John's e-mail and John's information. *Access actors* are invoked at the respective nodes to access the URLs and retrieve the content. The semantics of the component relationship helps the server to retrieve the aggregation specification (e.g., a xml template) and assign the template actor to interpret it.
- **Interpreting the presentation node:** The server first invokes *transformation computation actors* to compute the transformations. Since the XML-WML transformation is obtained from a source file, the appropriate *access actor* is invoked to retrieve the transformation. The `xyzTransform` is computed from the feature specification. The server invokes the *transformation computation actor* to traverse the *computed-from* edge and compute `xyzTransform`. The semantics of the *compose* edge instructs the server to locate the *composition actor* that would compose the transformation that finally gets applied by invoking the *transformation actor*.

The RDF server has an open library of functional actors. These actors are invoked based on the properties specified by the nodes and the semantics of the edges in the RDF model.

## 3. CONCLUSIONS

The component-based approach to constructing device and user agent profiles further promotes automating targeted transformations of XML content. A new and unknown device may be analyzed and mapped to a known device with the closest set of features. The resulting specification, similar to the one above, gets stored in the device database (or databases) and serves as input to the transformation computation actor that is responsible for composing generic XML-WML transformations with device and feature-specific transformations.

RDF is emerging as the foundation for next-generation frameworks that enable automated construction of Java applications. Such applications are composed of networks of metadata objects implementing RDF models and Java classes and methods that use metadata objects as processing context. RDF Servers navigate these RDF model-based networks and invoke actors implemented by the Java classes and methods. Using RDF for wireless applications opens up opportunities for building next-generation mobile services that don't depend on costly maintenance to keep up with new, evolving, and personalized devices. In other words, properly designed applications that are built in the era of cell phones and palm devices would still work for future microwave ovens that connect to the Internet to download cooking instructions.

## 4. REFERENCES

- [1] Composite Capabilities/Preferences Profiles, Work in Progress, W3C. <http://www.w3.org/Mobile/CCPP>
- [2] Resource Description Framework: Model and Syntax Specification, W3C, 1999. <http://www.w3.org/TR/REC-rdf-syntax>
- [3] Resource Description Framework Schema Specification, W3C, 2000. <http://www.w3.org/TR/rdf-schema>
- [4] User Agent Profile Specification, WAP Forum, 2000. [http://www1.wapforum.org/tech/documents/WAP-174\\_100-UAPProf-20000621-a.pdf](http://www1.wapforum.org/tech/documents/WAP-174_100-UAPProf-20000621-a.pdf)