Wireless's Need for the Semantic Web

The wireless environment differs from the wired environment by more than just the lack of wires. The aspect of mobility adds new constraints and different characteristics. The wireless devices themselves add physical constrains regarding input and output. The arduous methods of input dictate that wireless users enter a minimum amount of information. The small displays prevent users from browsing information, and suggest the need for more precise searches.

Based on aspects of mobility, wireless applications must also maintain the following characteristics: 1) simplicity 2) dynamism and 3) awareness. While the input/output constraints are device related, these three characteristics address how the mobile users themselves normally interface with the mobile devices.

In a general context, wireless users follow the 5-step RDCAP methodology to complete a task:

1. **R**ecognition of general task
2. **D**ecomposition of general tasks into sub-tasks and questions
3. **C**ompletion of sub-tasks and questions (recursively completed through RDCAP process)
4. **A**ggregation of results
5. **P**rovision of the solution

The RDCAP methodology realizes that wireless users search for information for more than just the information itself. The wireless users need the information to solve a more general task. The specific questions and services are aggregated to provide a **solution** for the general task. To address the physical and mobile characteristics of wireless devices, applications must provide solutions that follow the RDCAP methodology.

At this point, current Internet technologies prevent developers from producing applications that can provide the desired solutions because the Internet technologies require hard-coded links from solutions to services. This approach makes the solutions either 1) too general

or 2) too specific. A "general solution" can provide all users a limited solution that only addresses part of the general task, and a "specific solution" addresses the entire general task for only a few users. Neither solution type effectively addresses the wireless environment.

The semantic web solves this dilemma by enabling developers to establish a general framework in which the specific links to sites are determined at runtime. Developers specify the desired **types** of sites and services, but not the specific sites and services. At runtime, an agent determines the most appropriate services based on user preferences, input, location, etc, and the agent delivers the user a personalized aggregation of services.

For example, consider a travel solution for a user who wants to travel from Pittsburgh to Sao Paulo. The user needs ground transport from his home to the Pittsburgh International Airport, a flight from Pittsburgh to Sao Paulo, a hotel in Sao Paulo, and transportation from Sao Paulo's airport to the user's hotel. To provide a complete solution without the semantic web, developers would need to hardcode links to ground transportation for every city in the world. This method is doomed to fail.

Instead, the Semantic Web solution specifies the framework needs flight, hotel, and transportation services (the meaning of these terms is clarified by the use of ontologies). At runtime, an agent determines the most appropriate services for this framework based on the solution environment. In the travel example, the agent finds and uses services that provide ground transportation in Sao Paulo without foreknowledge of the services.

In this manner, the semantic web wireless application personalizes each solution returned to the user while insulating the user from unnecessary complexities. Without the need to hard-code solutions, developers can create solution frameworks that provide the specificity needed by users and the generality needed for actual implementation. This framework addresses the entire

RDCAP process and provides a solution for the user's general task that minimizes user input and maximizes simplicity.