Google Inc.
76 Ninth Avenue
4th Floor
New York, NY 10011

Google

Main 212 565-0000
Fax 212 565-0001
www.google.com

September 30, 2007

To Whom It May Concern:

This is an evaluation of the Ph.D. thesis titled "Addressing On-Demand Assembly and Adaptation Using a Runtime Intentional Versioning Engine" by Mgr. Jaroslav Gergič.

The rise of cellular phones, PDA's, and other internet-enabled devices made the Internet accessible from many different hardware and software platforms. Most of these platforms differ in the features and capabilities available for rendering the web content. To ensure the best user experience the web content needs to be tailored to each platform. This becomes an exponentially difficult task as the number of supported platforms increases. The work of Mgr. Jaroslav Gergič addresses the above task by the introduction of a run-time framework for content adaptation and versioning named Versatile.

The initial chapter of the thesis introduces the reader to the problem domain. The task of content adaptation is illustrated on a set of case studies, and the current state-of-the-art technologies are discussed in the related work and standards sections. Based on that discussion the author sets five goals his framework aims to fulfill: (1) consolidation of platform-specific metadata, (2) canonicalization of that metadata, (3) reduction of the gap between the abstract-level metadata and the platform-specific metadata, (4) lowering the level of expertise required to use the framework, and (5) enforcement of modularization and concept separation in the framework.

The presentation of results starts with the discussion of the goals and how these goals can be addressed in the framework's design, followed by an evaluation of several possible approaches to the implementation. The core of the thesis is formed by a description of the Versatile framework. The framework provides a means for describing device capabilities and application provisions via semantically rich, hierarchical properties. These capabilities and provisions are evaluated and paired in the run-time using a matching algorithm. An important part of the matching algorithm is a novel scoring function. The scoring function regulates the degree of approximation between the device resources and application provisions. The degree of this approximate match is modulated by a parameter called the scoring factor. The presentation is concluded by an evaluation of the results and their comparison with existing frameworks and standards.

The thesis presents a novel solution to an important problem in the area of adaptive run-time application versioning, and the results of the thesis are a step forward in the field. On the negative side, the thesis seems to be written hastily with a lack of attention to detail. Some important portions of the work have been omitted or abbreviated in the thesis. Therefore, I have the following questions to the author:

1) The scoring function on p. 74 describes a general solution to measuring an approximation match between two vectors of floating-point numbers, modulated

by a scoring factor. Unfortunately very little motivation, guidance, and evaluation are given regarding the selection of the function's parameters. Why is the scoring factor of 0.99 considered the best default scoring factor? How does the scoring function perform on a wider range of scoring factors? What is the range of property values that are involved in the function? And what is the exact definition of the distance function Δ? The property values and their ordering are also important to the outcome of the scoring function. How does the outcome of the scoring function change when the property ordering and the property values change? Are there any guidelines regarding which properties should be listed first in the property vector and what values they should be assigned?

2) The approximation matching algorithm is described on p. 72. In Step 2 of that algorithm the concept of a resource value being a generalization of a property value is used. The definition of generalization is not given in the thesis. What is the definition of generalization? In Step 4 the lexicographical ordering is applied on entries with the same values. It is not completely clear why and how this ordering is applied on the scoring function and whether a different ordering can yield better results. Why is the ordering necessary in the algorithm? Are there any other, possibly better options than the lexicographical ordering? In the last paragraph on p. 72 it is mentioned that the structures 'preference bag' and 'preference chain' result in modifications of the algorithm, but the description of their impact is vague and the modified algorithms is not given. What is the exact algorithm after all structures of the framework are incorporated in it?

3) The thesis contains several motivation examples and case studies. A comparison of the Versatile framework with other available standards and frameworks is also given in the thesis. But there is no complete example or case study of a system built in the Versatile framework. Can the author present a case study of such a system?

Provided that all the above questions are answered to full satisfaction of the dissertation defense committee, and taking into account the impact and novelty of the presented research, I recommend that the thesis is accepted as satisfactory. The work constitutes an evidence of Mgr. Jaroslav Gergič's capability to conduct independent research and contribute novel solutions to the research community.

RNDr. Miloslav Bešta, Ph.D.
Google Inc.
76 9th Avenue, 4th Floor
New York, NY 10011
United States
E-mail: milos@google.com
Phone: +1-212-565-8862
Fax: +1-212-565-0001