COMPONENT-BASED RUNTIME ENVIRONMENT FOR INTERNET APPLICATIONS

Mark Wallis, Frans Henskens, Michael Hannaford
Distributed Computing Research Group, University of Newcastle
Callaghan, NSW, 2308, Australia

ABSTRACT
The modern day Internet has evolved over the past 35 years from a single-purpose research network to a massively connected network of peers hosting a major percentage of world-wide communications. This evolutionary process has created a network that is performing tasks which were never in its original design. While emerging overlays such as peer-to-peer networking have spawned in an attempt to cope with the restrictive original design, core-level issues hamper further development. As the Internet grows into an increasingly connected and dynamic network these design flaws will continue to cause problems.

It is proposed to investigate the use of distributed computing and software runtime environments to host the next generation of Internet applications, in much the same way as the current crop of Internet browsers host thin web applications. This research will define a runtime environment that allows developers to create distributed, component-based applications for which the system manages cross-platform issues such as data persistence, and component-interaction issues such as concurrency. A detailed application programming interface (API) specification and software prototype will be developed to demonstrate how such a runtime environment can be used to address the concept of using the Internet as a platform for large-scale applications, while resolving known issues with such existing designs as Cloud Computing and Software-as-a-service (SAAS).

KEYWORDS
Component-based, internet, software-engineering

1. INTRODUCTION

The modern day Internet has evolved over the past 35 years [Leiner:2009] from a single-purpose research network to a massively connected peer-to-peer network of corporations and individuals. The original ARPANet [Marill:1966] was built to host primarily client-server communications between static research sites. The evolution of the modern day Internet beyond its original ARPANet roots has been classified largely by the ways in which information is presented, stored and altered by the user [OReilly:2005]. This classification, while mostly retrospective, has provided a way of conceptualising the evolution of how users and service providers interact on the Internet.

To be able to treat the Internet itself as a true platform for application delivery a number of changes must be made to the way users access the Internet. Lessons can be learnt from such areas as Operating Systems and Component Based Design that will allow for enhancements to the tools used by clients to interact with online services. This enhanced ‘Web Platform’ will allow greater flexibility in the way users interact with online applications. It will also provide a virtual environment that will allow larger and more complex Internet applications to execute while still integrating tightly with existing online web services.

This paper provides an introduction to our research, describes the research approach and presents a bibliography of work already published by our group as part of this research.
2. RESEARCH QUESTION

This research proposes a new design for deploying applications on the Internet. By designing applications in a component-based nature we can treat the Web itself as a distributed platform for execution. Take for example a simple personal calendar application. Such an application can be composed of a user interface component and a data storage component. By expanding the concept of the cloud (as shown in Figure 1) to encompass all computing resources available to the user the data storage component can execute either within a classic cloud environment (i.e. a remote server farm) or on a local server managed by the user themselves. An abstracted execution environment allows this data storage component to migrate to a remote server farm at a later date if the user wished to offload the storage management. Likewise, the user-interface component can migrate between resources managed by the user. With the appropriate runtime abstraction the user-interface is able to migrate from the user's desktop to their mobile device and back again, without losing state.

By promoting the concept of using the Web itself as a platform, the following high-level benefits can be expected:

- Local data storage of sensitive data while retaining web application access to the data for processing.
- Component level application patching that will allow application developers to push out updated code in a distributed manner, retaining many of the deployment benefits found with existing web applications.
- Distributed execution will allow unused resources in a user's local environment to be consumed by processor-intensive and data-intensive components.
- Component mobility will allow users to move components from system to system as required.
- User interaction to the application is abstracted as a graphical user interface (GUI) component and is deployed into the system in the same manner as other application components.
- Existing technology such as web browsers can be wrapped in component shells and executed within the new web platform environment providing a large amount of reuse of existing technology.
3. RESEARCH OBJECTIVE

The outcome of this research will be a component-based architecture that allows developers to design Internet-aware applications out of reusable components. These components will be able to be independently executed in a distributed fashion across any Internet connected host which the user has procured resources - either by owning the host in question or renting resources on a 3rd party server. The GUI components that the user interacts with will also be able to support distributed concepts such as migration. This will allow users to move the interface for an Internet-aware application from one host to another, perhaps as they move from a desktop workstation to a mobile device. The component framework and runtime environment design will be inherently cross-platform and language agnostic.

The Web Platform will in itself be treated much like a Virtual Machine (VM) in which these applications execute. The applications will not be able to gain access to any hardware or data resources without going through the proper functions calls into the VM. This virtualisation concept will benefit future research greatly due to the parallels it draws with the server-side of Cloud Computing concepts.

The components that comprise an application can be distributed over many hosts - including local hosts such as workstations and servers on the user’s home network and remote hosts such as those in 3rd party data centers. Each host that supports these components contains an instance of the runtime environment. Executing the runtime environment, which is essentially middleware, joins that host to the application network. When combining the various hosts that execute components for a user together a personal cloud is formed. How these personal clouds interact with each other to produce and consume services will mirror how Web Service technology is currently deployed. Components themselves in this design may be designated data storage components. This allows applications to distribute their data storage transparently across multiple locations, providing additional levels of redundancy and security not easily available to the existing web application paradigm. For instance, a web application may be deployed onto this platform that deals with sensitive data. The storage of this sensitive data would be the responsibility of a data storage component that executes only on hosts in the users home network.

4. RESEARCH APPROACH

To prove the concepts presented during this research a software prototype will be developed. Existing Open Source Web Browser technology such as the Mozilla engine will be leveraged to provide a base for the new software development. This will allow rapid prototyping of the enhanced concepts while retaining backwards compatibility with existing Web 1.0 and 2.0 technologies.

To prove that the enhanced Web Platform meets the needs of the next generation of Web applications, a sample application will be developed. This sample application will allow the testing of the Web Platform prototype while also providing a useful set of data which can be used to validate the research. A feature-comparable application will also be developed under Web 2.0 technologies. This will allow for metrics to be generated and compared proving the effectiveness of the design.

The final stage of research will cover the complete documentation of the research process, including the requirements, design, API and prototype documentation. This will result in a Thesis which will be submitted for official review.

5. EXISTING PUBLICATIONS

The following internationally peer-reviewed publications have already been produced by this research. I have been either primary author or co-author in all of the following publications.

- "A System for Robust Peer-to-Peer Communication with Dynamic Protocol Selection" [Wallis:2007]: This paper resolved the issue of server-to-client connectivity in the personal network where NAT and PAT devices limited a user’s connectivity to the Internet. The concepts and technologies introduced where provide a basis for ensuring that all nodes in the extended web platform will be able to maintain connectivity despite their location in the network.
- "Transaction Support for Interactive Web Applications" [Paul:2008]: This research explored the idea of extending transaction management across the web server/web browser divide. This concept is a stepping stone for expanding the application runtime environment to exist across both the content providers and the end user’s network space.
- "Web Browser Transactionality" [Wallis:2009]: This publication involved extensions to the 2008 paper above that provided enhanced stability to a transaction system which involved entities that may disappear from the network without notice. Extended transaction support across unstable networks is a key issue behind this research.
- "A publish/subscribe model for personal data on the Internet” [Wallis:2010]: This research is a stepping stone for resolving multiple issues related to the storage and access of personal data on the internet. It describes enhancements which eventually will form key components of the extended web platform.
- "A Distributed Content Storage Model for Web Applications" [Wallis:2010b]: This research expanded on our public/subscribe model for data storage, essentially offering a solution to distributed data storage of all online data. We formally defined the DSS (Distributed Storage Service) and provided results showing the performance benefits over classic Web 2.0 style storage models.
- “Expanding the Cloud: A component-based architecture to application deployment on the Internet” [Wallis:2010c]: This early-research presented the idea of expanded the concept of Cloud Computing to encompass client-side resources in addition to server farm resources. This new paradigm provides the basis for resource usage in a distributed component model.

6. CONCLUSION

The final outcome of this research will be a distributed component-based design for engineering Web-enabled applications that expand beyond the current confines of the web browser. Issues with existing Web 2.0 and Cloud Computing technologies will be addressed by re-thinking the concept of how applications are built. This will bypass issues caused by the gradual evolution of the Internet and provide a firm foundation for future advancements.

REFERENCES