07051 Abstracts Collection
Programming Paradigms for the Web: Web Programming and Web Services
— Dagstuhl Seminar —

Richard Hull[1], Peter Thiemann[2] and Philip Wadler[3]

[1] Bell Labs. Murry Hill, US
hull@research.bell-labs.com
[2] Univ. Freiburg, DE
thiemann@informatik.uni-freiburg.de
[3] Univ. of Edinburgh, UK
wadler@inf.ed.ac.uk

Abstract. From 28.01. to 02.02.2007, the Dagstuhl Seminar 07051 "Programming Paradigms for the Web: Web Programming and Web Services" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Web programming, programming concepts, program analysis, type systems, scripting languages, XML processing and querying

07051 Executive Summary – Programming Paradigms for the Web: Web Programming and Web Services

The world-wide web raises a variety of new programming challenges. To name a few: programming at the level of the web browser, data-centric approaches, and attempts to automatically discover and compose web services. This seminar brought together researchers from the web programming and web services communities and strove to engage them in communication with each other.

The seminar was held in an unusual style, in a mixture of short presentations and in-depth discussion sessions in small groups. This style enabled the participants to identify and discuss burning questions in small birds-of-a-feather sessions as well as in large plenary sessions. It required active participation of all attendees.

Keywords: Web programming, web services, programming paradigms, analysis and verification, implementation techniques and optimizations

Joint work of: Hull, Richard; Thiemann, Peter; Wadler, Philip

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2007/1125

07051 Working Group Outcomes – Programming Paradigms for the Web: Web Programming and Web Services

Participants in the seminar broke into groups on "Patterns and Paradigms" for web programming, "Web Services," "Data on the Web," "Software Engineering" and "Security." Here we give the raw notes recorded during these sessions.

Keywords: Web programming, web services, programming paradigms, analysis and verification, implementation techniques and optimizations

Joint work of: Hull, Richard; Thiemann, Peter; Wadler, Philip

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2007/1127

Calculus and algebra for distributed data management

Serge Abiteboul (INRIA Futurs - Orsay, F)

The sharing of content by communities of users (e.g., scientists) in a P2P context remains cumbersome. We argue that main reasons for this is the lack of calculus and algebra for distributed data management.

We present the ActiveXML language that extends the XML language with features to handle distribution. More precisely, ActiveXML documents are XML documents with a special syntax for specifying the embedding of Web service calls, e.g. XML queries such as XQueries. We also present ActiveXML algebra that extends ActiveXML notably with explicit control of data exchanges. ActiveXML algebra allows describing query plans, and exchanging them between peers.

Keywords: Distributed data management, XML, Web, Web service, calculus, algebra, monitoring, stream.

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2007/1126

Web, environment and types

Christophe Fouqueré (Université Paris-Nord, F)

Prerequisites for defining languages for web programming include modularity, safety, characterization of user or client/server interactions, description of services or site evolution and internalization of external operations (e.g. a click).

We present a new typed language based on a functional core and web objects (called xobjects) as first-class objects in order to satisfy such prerequisites.

These xobjects are parameterized and given by a set of reactions (aka services or reactions for requests) together with an XML structure possibly including expressions to be evaluated or xobject parameters.

Xobjects may be composed thanks to parameterization to obtain a tree structure called environment and we show in which extent operations on xobjects may induce a modification of such an environment. Furthermore, parameterization facilitates the definition of a lazy operational semantics.

Finally, the typing system allows to check 'correctness' and 'completeness' of a program: services or forms explicitly offered and defined reactions should be in one-to-one correspondence.

Flapjax

Shriram Krishnamurthi (Brown Univ. - Providence, USA)

Web and graphical applications lend themselves to natural expression in an event-driven, dataflow style. This style is appropriate for linking the components of the applications, while the contents of the components can still be expressed well in a traditional programming language. I describe Flapjax, a language built atop JavaScript to explore these ideas. Besides designing the language we have studied implementation techniques, principles for interfacing it to traditional libraries, applications such as scriptable debugging, program transformations to improve performance, and integration into programming environments.

Full Paper:
http://www.flapjax-lang.org/

Burning Question

Niels Lohmann (HU Berlin, D)

The SOA is often represented by a triangle, consisting of service requestor, service provider and service broker. The actions of the participants are usually "publish", "find" and "bind".

My burning question: What should be published and how can it be found?

Keywords: SOA, find, publish, matching, public view

Web Programming Supported by Introspective Model-Driven Development

Florian Matthes (TU München, D)

We present a new approach to model-driven development, which we call introspective model-driven development (IMDD).

The goal of IMDD is to improve the consistency between a system and its domain-specific models throughout the system life cycle. In our presentation, we focus on the following models relevant for web programming: A UML data model describing the persistent data managed by the system, a web interaction model describing user / service interaction patterns and bindings to HTML / XML templates and a configuration model capturing deployment information (e.g. network / database configuration).

The system is assumed to be implemented in an object-oriented, strongly-typed programming language with introspection and annotation support (in our case Java 5+). The textual and graphical models are visualized using an integrated development platform (in our case Eclipse and GEF).

The models are extracted from the source code in a systematic way through introspection based on a common compact metamodel. The central idea of IMDD is to limit the introspection to those parts of the application which are based on domain-specific frameworks. For example, the UML data model is extracted based on the use of an object-relational mapping framework, and the interaction model is extracted based on the use of a web visualization framework. There are two kinds of introspective frameworks Ũ introspective blackbox and introspective whitebox frameworks.

For both of these frameworks, the framework designer annotates those parts (methods) of the framework that later contribute to the model. While this process is domain-independent, the transformation of this (raw) model to a user-friendly view (which may also support code modifications through controllers) is domain-specific.

We demonstrate the practical applicability of our approach using the example of a substantial web application developed since 1999 by a small-sized German software vendor. Using Eclipse refactoring support, a single programmer was able to migrate the entire application in 6 months.

Joint work of: Matthes, Florian; Büchner, Thomas

Interaction-Safe State for the Web

Jay McCarthy (Brown Univ. - Providence, USA)

Recent research has demonstrated that continuations provide a clean basis to describe interactive Web programs. This account, however, provides only a limited description of state, which is essential to Web applications. This state is affected by the numerous control operators (known as navigation buttons) in Web browsers, which make Web applications behave in unexpected and even erroneous ways.

We describe these subtleties as discovered in the context of working Web applications. Based on this analysis we present linguistic extensions that accurately capture state in the context of the Web, presenting a novel form of dynamic scope. We support this investigation with a formal semantics and a discussion of applications. The results of this paper have already been successfully applied to working applications.

Full Paper:
http://scheme2006.cs.uchicago.edu/03-mccarthy.pdf

See also: McCarthy, Jay and Krishnamurthi, Shriram, "Interaction-Safe State for the Web" in Scheme and Functional Programming, 2006

Compiling Cryptographic Protocols for Deployment on the Web

Jay McCarthy (Brown Univ. - Providence, USA)

Cryptographic protocols are useful for trust engineering in Web transactions. The Cryptographic Protocol Programming Language (CPPL) provides a model wherein trust management annotations are attached to protocol actions, and are used to constrain the behavior of a protocol participant to be compatible with its own trust policy.

The first implementation of CPPL generated stand-alone, single-session servers, making it unsuitable for deploying protocols on the Web. We describe a new compiler that uses a constraint-based analysis to produce multi-session server programs. The resulting programs run without persistent TCP connections for deployment on traditional Web servers. Most importantly, the compiler preserves existing proofs about the protocols. We present an enhanced version of the CPPL language, discuss the generation and use of constraints, show their use in the compiler, formalize the preservation of properties, present subtleties, and outline implementation details.

Joint work of: McCarthy, Jay; Guttman, Joshua D.; Ramsdell, John D.; Krishnamurthi, Shriram

Web Programming with Modal Logic

Tom Murphy (CMU - Pittsburgh, USA)

I present ML5, an in-development programming language for web applications. The language's type system is endowed with a notion of location, allowing for type safe distributed manipulation of local resources.

Keywords: Distributed programming languages modal logic ml5 types

Static Analysis for Java Servlets and JSP

Anders Møller (BRICS - Aarhus, DK)

Developers of Web applications that use, for example, Java Servlets and JSP, often face the following problem: How can we guarantee that the dynamically generated output of the programs is always well-formed and valid XML?

The talk presents a program analysis that attacks this problem. The approach builds on a collection of program analysis techniques developed earlier in the JWIG and XACT projects, combined with Knuth's classical work on balanced context-free grammars.

Together, this provides the necessary foundation concerning reasoning about output streams and application control flow in the Web application code.

(This is joint work with Christian Kirkegaard. It was presented at the 13th International Static Analysis Symposium (SAS'06).)

Full Paper:
http://www.brics.dk/~amoeller/papers/servlets/

Reasoning about Behaviour on the Semantic Web

Barry Norton (The Open University - Milton Keynes, GB)

Semantic Web aims to encode machine-comprehensible knowledge, rather than merely human-comprehensible text via: ontologies, which represent shared conceptualisations; reasoning, which becomes satisfaction, entailment etc., in an underlying logic for these ontologies.

Semantic Web services encodes knowledge about services in ontological form. In the Web Services Modelling Ontology (WSMO): Capabilities represent the functional description of a service; Interfaces represent the behavioural description of a service. Within an interface: Orchestrations describe how a service may be made up as a composite behaviour involving other services; Choreographies describe the client interface to a service.

Most reasoning about service descriptions, however, concerns only the functional description. So: How can we reason over behaviour using Semantic Web technologies?

Keywords: Semantic web, Web services, Service-oriented architectures

Web Programming with XJ

Mukund Raghavachari (IBM TJ Watson Research Center, USA)

The development of web-based applications object-oriented languages such as Java is tedious and error-prone. A drawback of current frameworks is that the Java language and compiler do not understand XML or the Web Service standards natively — they operate on an object mapping of XML and WSDL specifications (for example).

As a result, programmers must deal with the complexities caused by the Object-XML impedance mismatch that often arises XJ, a language extension to the Java programming language simplifies the development of efficient XML-based applications easily using XJ. We will, through a hands-on demonstration, describe new features in XJ that simplify the development of Web Service applications. In XJ, a programmer can refer to WSDL port types as if they were Java interfaces. Specifically, a programmer may import WSDL specifications, declare classes to implement WSDL port types, instantiate concrete ports, or invoke operations on variables declared to have a WSDL port type. The compiler ensures statically that XML data passed/received as arguments to these oparations are valid with respect to the specification. Moreover, it generates appropriate marshalling/demarshalling code based on the bindings specified by the WSDL. We have developed an Eclipse-based editing environment that offers benefits such as content-assist and syntax highlighting that further simplify development. We will show through this demonstration that the tight integration of XML and WSDL support in Java significantly simplifies Web Service development. Attendees can follow along using an XJ prototype installation downloaded off IBM alphaWorks.

Keywords: XML, Java

Full Paper:
http://www.research.ibm.com/xj

Burning Question: The Web State Problem

Peter Thiemann (Universität Freiburg, D)

Web programming deals with several different notions of state.

Each tier has different requirements on the state and employs different mechanisms for its concurrent manipulation.

Is there a uniform way to handle all kinds of web state?

Joint work of: Neubauer, Matthias; Thiemann, Peter; Wehr, Stefan