博 士（情報科学） パウエル コートニー リカルド

学 位 論 文 題 名

A Formal Methodology for Concurrent Componentwise Development of Rich Internet Applications

（リッチインターネットアプリケーションの並行部品型開発の形式手法）

学位論文内容の要旨

The World Wide Web has rapidly evolved from a simple, page-driven, document-centric platform, in which each client is underpinned by a simple, synchronous request-response model, to a fully application-centric platform in which clients utilize highly interactive User Interfaces (UIs), complex internal interactions, and communicate with server resources both synchronously and asynchronously. At the forefront of the present stage of web evolution is a class of applications called Rich Internet Applications (RIAs). These applications combine the best features of traditional desktop applications, such as partial UI updates and fast UI response times, with the best features of traditional web applications, such as virtual ubiquity and operating system independence.

Traditionally, web applications could simply be developed as sequential, event-driven applications in which each event on the user interface of the client is immediately processed on the server; and the result of processing returned as a completely new page to the client. RIAs however, have introduced fine-grained concurrency, non-sequential event-driven semantics, and client-side computation to the client; resulting in a number of challenges. The first challenge is how to pinpoint the exact areas where concurrency occurs, and also how to deal with this type of concurrency in order to prevent well-known concurrency errors such as race conditions, response reordering, and deadlocks. The second challenge is how to unravel the non-sequential event-driven semantics and client-side computation logic so that program flow is understandable, and applications are easily maintainable and upgradable. The third challenge is how to ensure that the same logic can be easily implemented on dissimilar RIA platforms, without having to go back to the original idea. Finally, as in all web applications, there is the challenge of how to ensure that the different languages used to create the application, work harmoniously together. Overcoming these challenges is gaining increasing importance in light of the rapid trend towards multiple mashups and web services on a single client (in which each mashup and web service gets its data stream from different sources) which is explosively increasing the complexity of the web client in terms of these features.

The Equivalent Transformation Framework (ETF) is a highly abstract computation framework in which sets of Equivalent Transformation Rules (ETRs) are used to rewrite definite clauses, while preserving their declarative semantics. The foundation of the ETF is inherently nondeterministic -

enabling it to comprehensively model a wide variety of sequential and concurrent systems. In addition, the ETF is highly expressive, comprises a rich variety of highly independent ETRs, and has established theorems for the correctness of both sequential and concurrent programs generated within the framework from definite clauses. In this thesis, we propose a formal, systematic, componentwise, model-driven RIA development approach in which we extend the ETF to overcome the RIA challenges outlined above.

In chapter 1, we lay the backdrop for this thesis by first looking at the origin and evolution of both the Internet and the World Wide Web, and their relationship to each other. We then look at the evolution of web applications, and the features and advantages of RIAs which have made them so very indispensable. Finally, we look at the challenges resulting from the advent of RIAs, along with the associated research questions they have inspired.

In chapter 2, we introduce Formal Methods (FMs) and discuss the advantages of using formality during software development. These include the ability to reveal ambiguity, the ability to develop systems that can be reasoned about, and the ability to expose errors before commitment to code. We then establish the ETF as an FM, and also discuss its advantages as a formal framework.

In chapter 3, we give the general case of how dynamic, event-driven systems, comprising interacting concurrent components can be modelled and synthesized by extending the ETF. In doing this, we propose a correct-by-construction method for Dynamic Interactive Systems (DISs) modelling in which we establish behavioral rules for components, and a methodology for deriving the feasible interactions that may occur among concurrent components, during system construction. In addition to laying the foundation for the rule types used to model components and their interactions throughout this thesis, this chapter also gives an example of how we can systematically model and synthesize systems of randomly interacting concurrent components to achieve modular, understandable, easily evolvable systems.

In chapter 4, we first establish that RIAs are a class of DISs, and then introduce a loosely-coupled, layered concurrency model specific to them. We call this model the RIA Concurrency Model (RCM). This RCM captures the client-side fine-grained concurrency, computation, and non-sequential event-driven semantics native to RIAs. It also enables the identification and analysis of the areas of concurrency, and integrates both UI and behind-the-scenes activities in a loosely coupled manner. In addition, it divides the client-side computation into standard, manageable segments.

In chapter 5, we propose a process in which the RCM is used to incrementally construct prototypes of RIAs in a correct-by-construction fashion. In the proposed process, each change in the composition of the RIA user interface is regarded as a unique instance of the RCM, while a completed prototype is simply the union of all these instances. This prototyping activity facilitates practical analysis of the essential behavior of the RIA without commitment to a particular implementation platform.

In chapter 6, we propose a process in which the RCM is further utilized to systematically implement the prototypes constructed using the process outlined in chapter 5. In the process, we transform the prototype to a message passing metamodel in which each layer is considered a separate concurrent entity. This metamodel is then transformed to RIA implementation code, resulting in a modular

structure comprising independently concurrent objects decoupled from their interactions, and easily comprehensible behind-the-scenes processes.

We conclude this thesis in chapter 7 by summarizing the contributions made in our work and discussing how they addressed the research questions posed in chapter 1. We also discuss possible avenues for future work in this area.

学位論文審査の要旨

主　査　教　授　赤　間　　清
副　査　教　授　栗　原　正　仁
副　査　教　授　髙　井　昌　彰
副　査　准教授　棟　朝　雅　晴

学位論文題名

A Formal Methodology for Concurrent Componentwise Development of Rich Internet Applications

(リッチインターネットアプリケーションの並行部品型開発の形式手法)

リッチインターネットアプリケーション (RIA) は、ウェブブラウザなどのクライアントの機能を活かした、柔軟なインターフェースをもつ Web アプリケーションである。現在多くの RIA は、ユーザインターフェースに AJAX、Adobe Flex、SilverLight、JavaFx などを利用して作られており、HTML で記述されたページよりも操作性や表現力に優れている。しかし RIA はいくつかの異なる種類の部分システムからなり各部分を別々の言語で書く必要があることや、並行計算の複雑さなどにより、RIA の作成は容易ではなく、新しい明快な構築方法論の開発が望まれている。

本研究では、RIA の構造と RIA の望ましい作成過程への考察に基づいて、RIA の新しい構築方法論を提案している。提案する方法論では、手続き的な部品であるルールの集合によって、RIA の持つ本質的な並行性を細粒度で表現し、作成すべき RIA を単一の言語でモデル化するとともに、それをインプリメンテーションで通常使われている複数の命令型の言語で書かれたプログラムに自動変換する。これによって、本研究における RIA の構築では、並行性を的確に表現する高レベルのモデル化を、ルールを１つ１つ積み上げることによって行うことができ、それらのルールの妥当性を個別に確認していくことにより、正しさを確信しながらモデル化を行うことを意図している。また、モデルからいくつかの言語で書かれたプログラムを自動生成することにより、いろいろな言語で書かれた複雑な構造を持つ最終プログラムを直接書くことにより生じる不整合や誤りのリスクを回避することができる。

本研究は、等価変換ルールを核としたプログラム構築論 (Equivalent Transformation Framework, ETF) の成果を踏まえている。その先行研究では、論理式で書かれた仕様から、正当な等価変換ルール (ET ルール) を多数生成して、それを組み合わせて得られる多様なアルゴリズムの中から効率の高い逐次プログラムを得る。この方法は ET ルールを媒介としてプログラムを構築するが、ET ルールに課される正当性の十分条件が他の方法より緩いことにより、最も広範囲のアルゴリズムを提供でき、効率の良いアルゴリズムの発見に結びつくと考えられる。この理論は近年、逐次だけでなくある範囲の並列プログラムをも生成できる理論に拡張されており、逐次や並列などのいろいろな計算環境下での正当性を厳密に保証したプログラム構築方法論となっている。

本研究の方法は、この理論をさらに手続き的プログラミングの世界に拡張したものと見ることができる。ETFにおいて論理式仕様から得られるルールを部分的に利用しながら、手続き的部品であるルールを積み上げて並行システムを作りあげることが可能であり、先行研究で得られる正当性のメリットをシームレスに取り込むことができる。これは、本提案の枠組みの強力さと適切性を示している。

第1章では、RIAの特徴と長所について述べるとともに、RIAの出現がもたらした研究課題について述べている。第2章では、フォーマルメソッドについて述べるとともに、ETFを説明し、本方法の基礎を与えている。第3章では、Dynamic Interactive System (DIS) のクラスを導入するとともに、部品を積み重ねてDISを作る方法における挑戦すべき課題と、ETFを基礎としてそれを解決するアプローチの妥当性について議論している。第4章では、RIAをDISの1つのクラスとし、RIAに特徴的な緩い結合の階層化された並行性を持つモデルとしてRIA並行モデル (RIA Concurrent Model, RCM) を導入している。第5章では、RCMを用いて、ルールを1つ1つ追加して、それらのルールの集合としてRIAのプロトタイプを作る方法を提案している。第6章では、上記の方法で作られたプロトタイプをRIAのプログラムに変換する方法を与えている。第7章では、本研究の貢献をまとめるとともに、第1章で述べた研究課題に対する解答を与え、この分野での今後の課題について述べている。

本研究のRIA構築方法論は、記述の統一性、並行計算の細粒度記述、部品からの合成、抽象度、表現力、正当性、コードの自動生成などの観点から有望な提案であると考えることができる。

これを要するに、著者は、現代のWebアプリケーションに必須であるRIAの構築に関する問題点を指摘し、それらを解決するために、並行性を捉えたRCMモデルを提案し、それを基礎としてRIA構築の新しい方法論を提案し、システム構築の構造に関する評価やシステムの構築実験に基づいてその有効性を検証したものであり、情報科学とソフトウエア工学に対して貢献するところ大なるものがある。

よって著者は北海道大学博士 (情報科学) の学位を授与される資格あるものと認める。