



Doctoral Thesis

Maintaining consistency in collaboration over hierarchical documents

Author(s):

Ignat, Claudia-Lavinia

Publication Date:

2006

Permanent Link:

<https://doi.org/10.3929/ethz-a-005361349> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Diss. ETH No. 16766

Maintaining Consistency in Collaboration over Hierarchical Documents

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Sciences

presented by

Claudia-Lavinia Ignat

Diploma Engineer in Computer Science,
Technical University of Cluj-Napoca, Romania
born November 29, 1976
citizen of Romania

accepted on the recommendation of

Prof. Dr. M. C. Norrie, examiner
Prof. Dr. P. Molli, co-examiner

2006

Abstract

Collaboration is a key requirement of teams of individuals working together towards some common goal. Computer-supported collaboration is an increasingly common occurrence, driven by the evolving global nature of business, science and engineering and enabled by improvements in computing and communication technologies. Collaborative editing systems have been developed to support a group of people editing a document collaboratively over a computer network. Since not all user groups have the same conventions and not all tasks have the same requirements, it is important to support collaboration for various types of documents. Moreover, since a task has a set of development stages that require various forms of activity, customisation of the collaborative environment should be offered to support different modes of working between different sub-communities of users at different points in time.

The goal of this thesis was to investigate different settings for collaboration over the most common types of documents, such as textual, graphical and XML, with the aim of building a general theoretical framework to support the development of a range of collaborative editors.

A key issue for a general framework for collaboration is a common model that abstracts a large class of documents. The hierarchical model encompasses a wide range of documents and offers support for semantically structured documents. XML documents conform to a hierarchical structure by definition. We modeled text documents using a tree, where the document consists of a sequence of paragraphs, each paragraph of a sequence of sentences, each sentence as a sequence of words and each word as a sequence of characters. Graphical documents are also modeled by a hierarchical structure: groups are represented as internal nodes, while simple objects are represented as leaves. In order to work in a uniform way with different semantic units of the document, we adopted a multi-level editing approach where we associated with an element of the document the editing operations targeting that element.

Operational transformation is a suitable mechanism for maintaining

consistency over copies of shared objects subject to collaboration. We extended the operational transformation approach to work for hierarchical models of documents that have the operations distributed throughout the tree. Our approach for maintaining consistency allows any existing operational transformation algorithm for linear structures to be applied recursively over the hierarchical structure of the document.

The document model enables flexible granularity for the propagation of changes over the network, for the detection and resolution of conflicts and for details about user activity on different parts of the document. For instance, for text documents, the granularity can be dynamically varied at the level of paragraphs, sentences, words or characters, and, for XML documents, at the level of elements, attributes, word nodes or characters.

We first show how the multi-level editing approach was applied to both XML and textual documents. We then show that different mechanisms for maintaining consistency are required for graphical documents.

A novel operation serialisation mechanism was used for consistency maintenance in the case of graphical editing. Conflict handling can be customised to suit the requirements of specific applications. We classified conflicts into real and resolvable, depending on whether an execution order between pairs of operations can be established or not. We offer users the possibility to define the types of conflicts between the operations and the policy for the resolution of conflicts. Our approach for graphical documents is the first one that deals with complex operations such as grouping and working with layers.

In addition to supporting different types of documents, the aim was to also support different modes of communication, such as real-time and asynchronous. Real-time collaboration implies that changes performed by users are seen immediately by other users and asynchronous collaboration over a repository implies that users work in isolation and synchronise their changes against the repository at a later time. We applied the same operational transformation mechanism for maintaining consistency over text and XML documents, for both real-time and asynchronous modes of collaboration. For maintaining consistency over graphical documents, we applied the same serialisation mechanism for both real-time and asynchronous modes of collaboration.

As a proof of concept of the theoretical framework described in the thesis, we built collaborative editors for text, XML and graphical documents, both for real-time and asynchronous collaboration relying on a shared repository.

Zusammenfassung

Kollaboration ist eine Hauptanforderung, um ein Team von Individuen, welche ein gemeinsames Ziel verfolgen, zu unterstützen. Angetrieben durch die zunehmende Globalisierung von Wirtschaft, Wissenschaft und des Ingenieurwesens, unterstützt durch Verbesserungen im EDV und Kommunikationsbereich, nimmt die Zahl der Applikationen für computerunterstütztes kooperatives Arbeiten stark zu. Kollaborative Editoren wurden entwickelt, um es einer Gruppe von Benutzern zu erlauben, Dokumente gemeinsam über ein Rechnernetz zu editieren. Dabei ist es wichtig, verschiedene Arten von Dokumenten zu unterstützen, da nicht alle Benutzergruppen nach den gleichen Richtlinien arbeiten und unterschiedliche Aufgaben verschiedene Anforderungen an ein kollaboratives System stellen. Weil eine Aufgabe in eine Menge von Unteraufgaben aufgeteilt werden kann, welche verschiedene Aktivitäten voraussetzen, ist es entscheidend, dass die kollaborative Umgebung an die unterschiedlichen Arbeitsweisen bestimmter Gruppen von Benutzern während bestimmter Phasen angepasst werden kann.

Das Ziel dieser Dissertation war es, verschiedene Szenarien für die Kollaboration mit verbreiteten Dokumenttypen wie Text-, Graphik- und XML-Dokumenten zu untersuchen und daraus einen allgemeinen theoretischen Rahmen abzuleiten, der die Entwicklung einer Reihe kollaborativer Editoren zu unterstützen vermag.

Eine Voraussetzung für ein allgemeines Kollaborationssystem ist ein allgemeingültiges Modell, welches eine möglichst grosse Klasse von Dokumenttypen abstrahiert. Ein hierarchisches Modell umfasst zahlreiche Dokumenttypen und unterstützt semantisch strukturierte Dokumente. XML Dokumente sind nur ein Beispiel von Dokumenten, welche per Definition hierarchisch strukturiert sind. Wir modellieren Textdokumente als Baumstruktur, wobei jedes Dokument aus einer Reihe von Paragraphen besteht. Ein Paragraph wiederum enthält eine Folge von Sätzen, welche ihrerseits aus einer Sequenz von Wörtern zusammengesetzt sind. Die einzelnen Wörter schliesslich bestehen aus einer Folge von Buchstaben. Graphi-

sche Dokumente werden ebenfalls mit Hilfe einer hierarchischen Baumstruktur modelliert, wobei Gruppen von Objekten als interne Knoten und einzelne Objekte als Blattknoten repräsentiert werden. Um die unterschiedlichen semantischen Einheiten eines Dokumentes einheitlich behandeln zu können, wenden wir einen mehrstufigen Editieransatz an, bei welchem ein Element des Dokumentes mit den dazugehörigen Editieroperationen verknüpft wird.

Das “Operational Transformation” Verfahren eignet sich, um die Konsistenz verschiedener Kopien eines verteilten Objektes zu gewährleisten. Wir haben dieses Verfahren dahingehend erweitert, dass es auch für hierarchische Dokumentenmodelle, bei denen die Operation über den ganzen Baum verteilt sind, eingesetzt werden kann. Unser Ansatz zur Konsistenzhaltung wendet beliebige existierende Operational Transformation Algorithmen für lineare Strukturen rekursiv auf die hierarchische Struktur eines Dokumentes an.

Unser hierarchisches Dokumentenmodell ermöglicht eine flexible Granularität bei der Übertragung von Änderungen über ein Netzwerk. Die gleiche Flexibilität besteht bei der Erkennung und Auflösung von Konflikten, sowie beim Überwachen von Benutzeraktivitäten in unterschiedlichen Teilen eines Dokumentes. Für Textdokumente kann die Granularität auf dem Level von Paragraphen, Sätzen, Wörtern oder Buchstaben dynamisch variiert werden, während dies bei XML-Dokumenten auf dem Niveau von Elementen, Attributen, Wörtern oder Buchstaben möglich ist.

Wir zeigen zuerst, wie der mehrstufige Editieransatz für XML- und Textdokumente verwendet wurde. Anschliessend erörtern wir, wie sich die Mechanismen zur Konsistenzhaltung in graphischen Dokumenten von denjenigen für Text- und XML-Dokumente unterscheiden.

Ein neuartiger Ansatz zur Serialisierung von Operationen wurde für die Konsistenzhaltung im Falle eines graphischen Editors entwickelt. Dabei kann die Konfliktbehandlung entsprechend den Anforderungen einer spezifischen Applikation angepasst werden. Wir unterscheiden zwischen wirklichen und auflösbaren Konflikten, je nachdem ob eine Ausführungsreihenfolge für Paare von Operationen gefunden werden kann oder nicht. Benutzer haben die Möglichkeit, die Konflikttypen zwischen einzelnen Operationen und die entsprechenden Methoden zur Auflösung dieser Konflikte zu definieren. Unsere Lösung zum Editieren von graphischen Dokumenten ist die erste, die sich mit komplexen Operationen wie dem Gruppieren von Objekten und der Verwaltung von Layern beschäftigt.

Zusätzlich zur Unterstützung unterschiedlicher Dokumententypen war es das Ziel dieser Dissertation verschiedene Kommunikationsarten, wie syn-

chrone (Echtzeit) und asynchrone Kommunikation zu unterstützen. Echtzeitkollaboration impliziert, dass die Änderungen eines jeden Benutzers sofort für andere Benutzer sichtbar sind, während die Benutzer bei asynchroner Kollaboration isoliert arbeiten und ihre Änderungen zu einem späteren Zeitpunkt mit den Daten eines Repositories abgleichen. Wir haben den gleichen Operational Transformation Mechanismus zur Konsistenz-erhaltung in Text- und XML-Dokumenten sowohl für Echtzeit- als auch für asynchrone Kollaboration verwendet. Für die Konsistenz-erhaltung bei graphischen Dokumenten kommt bei Echtzeit- und asynchrone Kollaboration ein und der selbe Serialisierungsmechanismus zur Anwendung.

Als Beweis für den theoretischen Rahmen dieser Dissertation haben wir kollaborative Editoren für Text, XML und graphische Dokumente implementiert, die sowohl Echtzeit- als auch asynchrone Kollaboration basierend auf einem gemeinsam benutzen Repository ermöglichen.