Edit Patterns in Collaborative Ontology-Engineering Projects

Dipl.-Ing. Simon Walk, BSc

Dissertation
zur Verleihung des akademischen Grades
Doktor der technischen Wissenschaften

Institut für Informationssysteme und Computer Medien
Technische Universität Graz

Graz University of Technology

Begutachter: Prof. Dipl.-Ing. Dr.techn. Markus Strohmaier

Graz, Oktober 2015

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)

Abstract

In the field of computer science, ontologies represent digital data structures that are used to formally and explicitly model a specific domain. More precisely, Gruber [1993]; Borst [1997]; Studer et al. [1998] defined an ontology as “an explicit specification of a shared conceptualization”. In this context, the explicit specification refers to a machine-readable construct, which represents an abstraction of the real world (the shared conceptualization). In practice, ontologies enable computers to “understand” and reason on the defined relationships between the conceptualized objects of the represented domain. Over the last decade, the importance of ontologies, particularly in the biomedical domain, has progressively increased. New and complementing fields of application have been added to existing structured knowledge bases, triggering substantial increases not only in size but also in the complexity of the corresponding ontologies. For example, the next revision of the International Classification of Diseases (ICD), which is currently developed as an OWL-based ontology by the World Health Organization in cooperation with Stanford University, consists of roughly 50.000 classes—four times the size of the current revision. No single individual has the required knowledge to create such ontologies on their own. Hence, a new trend has emerged, emphasizing and augmenting the cooperative development of (large-scale) ontologies by distributed teams of domain experts. In particular, this next revision of ICD is collaboratively engineered by a core development team of roughly 200 domain experts over the Internet. However, collaborative ontology engineering still represents a fairly new field of research, with many (un)known and yet unexplored and unresolved problems associated with it. For example, the quality of an ontology is typically evaluated only by assessing the resulting ontology itself (e.g., by comparing it against a golden standard or by assessing its performance for a specific predetermined task). However, the intricate and dynamic (social) processes that occur when users collaboratively engineer an ontology provide an additional source of information, which should be included in the quality assessment process. Analogously to modern software development methods, which specifically aim at maximizing the quality of the resulting program by managing the development process, it is necessary to better understand the ongoing social processes that occur within collaborative ontology-engineering projects. These newly obtained findings could potentially help to move from traditional ontology evaluation to an empirically informed evaluation and management of the processes around collaborative ontology-engineering projects. Hence, the main objective of this thesis is to gather new empirical insights into the ongoing processes that occur when users collaboratively engineer an ontology. In particular, this work closely investigates regularities and (sequential) patterns in the logs of changes of multiple collaborative ontology-engineering projects. First, the analyses presented in this thesis study if users exhibit sequential patterns while collaboratively editing an ontology using Markov chains. Further, the fitted Markov chain models are extensively analyzed, interpreted and discussed in the context of each investigated project. Second, this thesis demonstrates how the obtained Markov chain models can be leveraged to predict different aspects about future actions of contributors. Finally, this thesis demonstrates how HypTrails—a framework to compare hypotheses about sequential data—can be applied to study user behavior in ontology-engineering projects. Overall, the results presented in this thesis are of relevance to researchers and practitioners interested in studying and improving collaborative ontology-engineering projects.

Kurzfassung

In der Informatik stellen Ontologien digitale Datenstrukturen zur formalen und expliziten Modellierung einer spezifischen Domäne dar. Genauer gesagt definierten Gruber [1993]; Borst [1997]; Studer et al. [1998] eine Ontologie als eine “explizite Spezifikation einer geteilten Konzeptualisierung”. In diesem Kontext bezieht sich die explizite Spezifikation auf ein, von einem Computer lesbares, Konstrukt welches eine Abstraktion der echten Welt (die geteilte Konzeptualisierung) darstellt. In der Praxis erlauben es Ontologien einem Computer zuvor definierte Beziehungen zwischen konzeptualisierten Objekten der zu repräsentierenden Domäne zu “verstehen” und über diese Schlussfolgerungen anzustellen. Im Verlauf des letzten Jahrzehnts hat die Bedeutung von Ontologien, speziell im biomedizinischen Bereich, fortschreitend zugenommen. Neue Anwendungsgebiete wurden zu existierenden Ontologien hinzugefügt, welche einen Anstieg in Größe und Komplexität für die entsprechenden Ontologien ausgelöst haben. Zum Beispiel, die nächste Revision der International Classification of Diseases (ICD), die derzeit als OWL-basierte Ontologie von der Weltgesundheitsorganisation in Kooperation mit der Stanford Universität entwickelt wird, beinhaltet ungefähr $50,000$ Klassen, vier Mal mehr als die aktuelle Revision. Keine Einzelperson hat das nötige Wissen um so eine Ontologie alleine zu erstellen. Deshalb kam es zu einem neuen Trend bei dem die kollaborative und verteilte Durchführung von Ontologieentwicklungsprojekten in den Vordergrund rückt. So wird die nächste Revision der ICD von einem Kernteam, bestehend aus ungefähr 200 Domänenexperten, kollaborativ über das Internet erstellt. Allerdings stellt die kollaborative Entwicklung von Ontologien noch ein relativ junges Forschungsgebiet dar, mit vielen (un)bekannten und ungelösten Problemen. Zum Beispiel wird die Qualität einer Ontologie typischerweise nur durch die Bewertung der resultierenden Ontologie erfasst (z.B., Vergleiche mit einem “Golden Standard” oder über die Messung der Performance für spezielle Aufgaben). Allerdings stellen die komplizierten und dynamischen Prozesse solcher Projekte eine zusätzliche Informationsquelle dar, welche in den Prozess zur Qualitätsbewertung eingebunden werden sollte. Analog zu modernen Softwareentwicklungsprozessen, die gezielt versuchen die Qualität des resultierenden Programms über die Steuerung des Entwicklungsprozesses zu maximieren, ist es notwendig diese Prozesse in kollaborativen Ontologieentwicklungsprojekten besser zu verstehen. Diese neuen Erkenntnisse könnten helfen den Schritt von traditionellen Evaluierungsmethoden für Ontologien zu einer empirisch informierten Evaluierung, welche alle Prozesse rund um die kollaborative Ontologieentwicklung berücksichtigt, zu machen. Daher ist es das Hauptziel dieser Arbeit neue empirische Erkenntnisse über die komplizierten Dynamiken und Prozesse, welche auftreten wenn mehrere Benutzer kollaborativ eine Ontologie entwickeln, zu erforschen. Diese Doktorarbeit untersucht Regelmäßigkeiten und Muster in den Logs kollaborativer Ontologieentwicklungsprojekte. Zuerst, wird durch die Verwendung von Markov Ketten untersucht ob Benutzer Muster aufweisen wenn sie gemeinsam eine Ontologie erstellen. Zusätzlich werden die verwendeten Modelle analysiert, interpretiert und im Kontext des jeweiligen Projektes diskutiert. Zweitens demonstriert diese Arbeit wie die erhaltenen Modelle verwendet werden können um Aspekte von zukünftigen Aktionen von Benutzern vorherzusagen. Schließlich untersucht diese Arbeit wie HypTrails, ein Framework um Hypothesen über sequentielle Daten zu vergleichen, verwendet werden kann um das Benutzerverhalten in kollaborativen Ontologieentwicklungsprojekten zu studieren. Insgesamt sind die präsentierten Ergebnisse relevant für Forscher und Praktiker, welche an der Studie sowie der Verbesserung von kollaborativen Ontologieentwicklungsprojekten interessiert sind.

Acknowledgements

First of all, I would like to thank my Ph.D. supervisor Markus Strohmaier for his support, excellent feedback and guidance throughout my Ph.D. studies. I am really grateful for all the discussions we had, which were always inspiring and highly motivating. Whenever I had a problem or didn't know how to proceed with my work I knew I could ask and would receive very good and valuable advice.

I would also like to thank Denis Helic for all his input throughout my Ph.D. studies. I am grateful for the different collaborations and I hope that we will continue to do so in the future. Further, I would also like to specifically thank Philipp Singer for all of his efforts and support during our different collaborations.

I would like to thank Mark Musen, Natasha Noy, Tania Tudorache, Csongor Nyulas and everybody else I was fortunate enough to meet and collaborate with at the Stanford Center for Biomedical Informatics Research. Not only did they provide a very welcoming environment at the BMIR but they also really helped me to progress with my studies and were always available if I had questions or needed advice.

Also, all of my colleagues at work and friends deserve my gratitude. In particular, I would like to thank Florian Geigl, Daniel Lamprecht, Patrick Kasper and Lukas Eberhard for their collaboration and thoughts. Without all of you, going to work every day would not have been as fun as it was!

Additionally, I would especially like to thank my family, who made all of this possible and supported me throughout my studies. Whenever I needed a break from work or lacked motivation to get up early to start working on my thesis, I could count on Snickers, Bounty and Knoppers (my cats) to keep me occupied (usually with preparing their food).

Finally, I would also like to thank my wife Silvia, the one person who actually supported me the most during my Ph.D. studies. Some of my nights, weekends and weeks have been very work intensive, but I knew that I could always count on your support!

Contents

1 Introduction

1.1 Motivation

Ontologies represent digital and machine-readable data structures that are used to formally and explicitly model a specific domain as an abstraction of the real world. In practice, ontologies enable computers to “understand” and reason on the defined relationships between the conceptualized objects of the represented domain.

Recently, the usage of structured data, particularly in the field of computer science, has significantly increased due to different ambitious and encompassing efforts, such as the Semantic Web, which tries to bring structured data to the Web, or the Linked Open Data initiative, which connects and maps freely available structured open datasets. Additionally, ontologies see increased usage across many other fields of research. For example, biomedical ontologies play a critical role in acquiring, representing and processing information about human health. One of these structured knowledge bases is the International Classification of Diseases (ICD), a taxonomy that is currently in its 10^{th} revision and is used in more than 100 countries around the world to encode patient history data and diseases, as well as to collect and compile health-related statistics. Similarly, the National Institutes of Health developed an important OWL-based vocabulary—the National Cancer Institutes Thesaurus (NCIt)—to classify cancer and cancer-related terms

With increasing relevance and prospects of practical applications, biomedical taxonomies, thesauri and ontologies were extended to model and cover new findings and to expand and complement their original areas of application. For example, the 11th revision of the International Classification of Diseases (ICD-11)—currently under active development by the World

Health Organization and Stanford University—consists of roughly 50,000 classes, representing a vast variety of different diseases and causes of death. In particular, ICD-11 is roughly four times the size (in terms of classes) of its predecessor. Additionally, and in contrast to previous revisions, the foundation component of ICD-11 is implemented as an OWL-based ontology.

As no small group of domain experts, let alone single individuals, have the required expertise and time to develop such highly specialized and large-scale ontologies, new requirements for the engineering process emerged. To that end, it has become essential for user-interface designers and project administrators not only to find answers but also to identify these new requirements by analyzing and broadening our understanding of the intricate and ongoing processes that occur when users collaboratively engineer ontologies. This is true especially since the development of an ontology already represents a complex task, which even further increases in complexity when adding a layer of social interactions on top of the development process.

Traditional evaluation methods for structured data are mainly focused on assessing the quality of the resulting ontology itself, for example, by comparison to a golden standard or by testing and measuring the performance and accuracy for a predetermined task. Analogously to modern software development methods, which specifically aim at maximizing the quality of the resulting program by managing the development process, it is important to improve our understanding of the dynamic and social processes that occur within collaborative ontology-engineering projects.

Hence, uncovering such empirical insights, which can potentially help to adapt and improve existing ontology-engineering tools or even to devise new and adapt existing development and evaluation strategies, which include the social processes around collaborative ontology-engineering projects and allow for empirically informed decisions to be made, represent very important first steps towards overall improved and more easily maintainable structured knowledge representations.

This thesis primarily deals with analyzing the processes that occur when users collaboratively engineer biomedical ontologies by investigating, modeling and leveraging regularities and edit patterns in the change-logs of such projects. Thus, this thesis is relevant for researchers and practitioners interested in the topic of collaborative ontology engineering.

In Section 1.2 I provide a brief overview of the problems and challenges assigned to collaborative ontology engineering. This is followed by a description of the main problems and approaches this thesis addresses in Section 1.3, the declaration of the main research questions in Section 1.4 as well as the listing of the main publications this thesis builds upon in Section 1.5. Finally, I elaborate on the contributions and implications of this thesis in Section 1.6 and outline the structure of this thesis in Section 1.7.

1.2 Collaborative Ontology Engineering

This thesis analyzes regularities and patterns in the change-logs of different collaborative ontology-engineering projects from the biomedical domain. As such large-scale ontology-development projects emerged only recently, collaborative ontology engineering still represents a relatively young and unexplored field of research with many new problems, risks and challenges associated with it. Hence, these problems still have to be identified, defined and can only then be addressed by researchers and practitioners, for example, to better manage the ongoing and complex social dynamics within such projects or to improve the general tool support. Analogously to traditional and well studied online peer production systems, such as Wikipedia, contributors to collaborative ontology-engineering projects engage remotely (e.g., via the Internet or a client–server architecture) and usually in a distributed manner to develop and maintain ontologies.

One particular problem of ontology-engineering projects, especially when conducted in a distributed manner, is finding consensus between authors on certain subjects. As an ontology represents a formalized and abstract representation of a specific domain, contradicting opinions on specific topics can occur. Hence, similar to face-to-face meetings, such collaborative ontology-engineering projects are in need of adapted and specialized tools

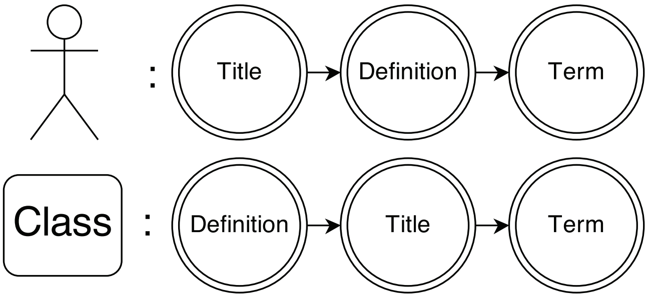


Figure 1.1: **Types of edit sequences.** This plot depicts the different types of edit sequences investigated in this thesis. The top row depicts a sequence based on the properties (from potentially different classes) of an ontology that were changed by one specific user. The bottom row depicts a property-based edit sequence for a specific class of an ontology. The different properties that were changed, either by the user or for the specific class, are chronologically (ascending) sorted, starting with the one that was changed first and ending with the most recently changed one.

that augment collaboration and actively support contributors in reaching consensus when modeling topics and domains of the real world.

The majority of literature about collaborative ontology engineering is focused on finding and defining requirements for the processes and tools used in these projects [Noy and Tudorache, 2008; Groza et al., 2013; Krötzsch et al., 2006; Auer et al., 2006; Tudorache et al., 2013].

In contrast to previous research, this thesis focuses on the identification and interpretation of regularities and sequential patterns by analyzing the change-logs of different collaborative ontology-engineering projects.

For example, whenever users interact with an ontology, they leave digital traces of these interactions. In the context of collaborative ontology-engineering projects, these traces are represented by chronologically ordered sequences of edit actions (see Figure 1.1), conducted on the corresponding ontologies. If a specific user changes multiple properties of a specific class, it is possible to create a sequence of properties that this specific user has changed—referred to as *user-based edit sequences*. Additionally, it is also possible to create a chronologically ordered sequence of changed properties for each class in the ontology—referred to as *class-based edit sequences*.

One specific user-based edit sequence for properties of an ontology is depicted in the top row of Figure 1.1. This particular user first changed the property *Title*, then *Definition* and finally *Term* (of potentially different classes of one ontology). In contrast, the bottom row of Figure 1.1 depicts a class-based edit sequence for properties, where for this specific class, the first property that was changed was *Definition*, then *Title* and finally *Term* (by potentially different users). In these examples, the properties of the ontology represent the different states of the edit sequences. These can be replaced by other states, which allow for different analyses, such as the types of changes that were conducted, the distance between the changed classes or the sequence of users who worked on a class.

1.3 Problem Statement, Objectives and General Approach

Problem Statement. Today, many important structured knowledge bases, particularly from the biomedical domain, are developed in a collaborative manner. However, the inherent and dynamical processes that occur when users collaborate on developing an ontology are mainly uncharted territory and add an additional layer of complexity to the already difficult development task. Analogously to software development methods, which focus on managing the engineering processes to increase the quality of the resulting program, it is important to broaden our understanding of the dynamic and intricate social processes that occur in collaborative ontology-engineering projects to help move from traditional ontology evaluation approaches to quality assessment approaches that include the evaluation of the (social) processes around collaborative ontology-engineering projects.

Additionally, the complexity and size of these collaboratively created ontologies is increasing due to new and extended fields of application and ever growing, highly specialized domain knowledge. As a direct consequence, users have to constantly overcome several obstacles before being able to contribute to such collaborative endeavors. In particular, they have to get acquainted with the specific domain that is to be modeled and the tools that are used to create the structured knowledge representation. Further, users working on such projects have to locate the parts and areas of an ontology that they have sufficient specialized domain knowledge about to be able to add meaningful contributions. Hence, it has become increasingly important not only to aid users in their task of developing ontologies but also to broaden our understanding of the ongoing processes that occur while users collaboratively engineer ontologies.

Objectives. To that end, this thesis aims at providing actionable insights that allow project administrators, ontology tool developers as well as contributors to make (empirically) informed decisions about the engineering process. The results presented in this work represent a first step towards a better understanding of the processes that occur when users collaboratively engineering structured knowledge bases. In that regard, the main focus of this thesis is set on identifying if and to what extent regularities and sequential patterns can be identified in the change-logs of collaborative ontology-engineering projects. Further, this thesis has the objective of discussing and interpreting the structure and usefulness of the identified patterns in general, as well as in the context of each project individually. Additionally, this thesis sets out to investigate and compare different hypotheses—inferred via empirical analyses—about how users collaboratively develop ontologies.

General approach. This thesis makes use of different methods to extract sequential patterns in sequences of categorical data. Markov chains are used to model and predict aspects of future actions conducted by users as well as to investigate and compare different hypotheses about how users edit ontologies in collaborative ontology-engineering endeavors.

1.4 Research Questions

The greater challenge of the analyses and articles presented in this thesis is the study of the ongoing processes of how users collaboratively engineer ontologies. In particular, I am interested in (i) investigating how sequential edit patterns of users who collaboratively create an ontology can be identified and extracted, (ii) modeling and leveraging these patterns to predict different aspects of future actions conducted by the users of such collaborative ontology-engineering projects and (iii) explaining these patterns by defining, formulating, comparing and evaluating hypotheses about how users collaboratively develop ontologies.

See Table 1.1 for an overview of the research questions, the related topics, as well as the corresponding main contributions for each article.

RQ1: How can we extract and identify edit patterns in collaborative ontology-engineering projects?

Problem. Contributors of collaborative ontology-engineering projects are confronted with many different obstacles when editing the content of an ontology. First of all, users should have comprehensive and specialized knowledge of the domain that is to be modeled by the ontology. Due to growing potential fields of application, particularly in the biomedical domain, and thus increases in size and complexity of the resulting ontologies, it has become progressively harder to search and quickly retrieve classes, which users have sufficient knowledge about to add meaningful contributions. As a consequence, it has also become increasingly important to support and augment contributors in their tasks of adding or editing content in an ontology. To be able to provide this kind of support it has become crucial to broaden our understanding of the ongoing social processes that occur when users collaboratively develop an ontology. In particular, when learning more about the habits of these users—in the form of sequential usage patterns and regularities in general—new methods and tools can be developed to aid them in their task of creating, editing and maintaining (especially large-scale) ontologies.

The objective of this first research question is to present a detailed description of all steps required to analyze such habits by identifying and extracting regularities and sequential patterns in the change-logs of different collaborative ontology-engineering projects.

Approach. In a first step it is necessary to conceptualize a consistent process that guides users, who are interested in analyzing sequential usage patterns in collaborative ontology-engineering projects, and also describes in detail how the obtained results can be interpreted. In particular, the analysis presented in Walk et al. [2015b] makes use of Markov chains as well as different information criteria and a cross-fold prediction experiment to identify and extract sequential usage patterns from the change-logs of collaborative ontology-engineering projects.

Findings and contributions. To tackle this research question I present a detailed description of the process on how to use and apply Markov chains of varying orders to detect sequential patterns in the change-logs of collaborative ontology-engineering projects in Walk et al. [2015b]. Further, the presented work comprises a first exemplary interpretation of the findings for a large-scale and real-world collaborative ontology-engineering project and showcases that edit patterns can not only be identified but also used for predicting different aspects of future changes of users.

RQ2: Do edit patterns in collaborative ontology-engineering projects exist and, if so, how do they look like?

Problem. Having described the process and methods required for extracting sequential usage patterns, this research question aims at analyzing if and to what extent such patterns can be detected in different collaborative ontology-engineering projects. Further, this research question aims at investigating the structure of the discovered edit patterns. In general, identifying and extracting such patterns represents an important step towards a better understanding of the complex social processes of users while engaging in the task of collaboratively engineering an ontology. The findings for answering the first research question, described in Walk et al. [2015b], already indicate that sequential patterns in collaborative ontology-engineering projects can be uncovered and contain valuable information for project administrators, ontology tool developers and contributors. The main objective of this research question is to expand on these results by further identifying, analyzing, modeling, leveraging and discussing regularities and sequential patterns in the change-logs of multiple collaborative ontology-engineering projects.

Approach. To that end, I fit first-order Markov chain models on the change-logs of different collaborative ontology-engineering projects. Further, I investigate and discuss the structure of the resulting transition matrices of the fitted models in the context of each project individually. To assess the extent of sequential patterns in the change-logs of collaborative ontology-engineering projects, colleagues and I present a case study to identify sequential action patterns in five collaborative ontology-engineering projects from the biomedical domain. The main focus of this work was set on analyzing, detecting and measuring the extent of regularities and sequential patterns, relying on PrefixSpan, a well-known pattern-mining algorithm, and Markov chain models. Different model selection methods and a cross-fold prediction experiment were used to determine the appropriate order, which represents the best trade-off between model complexity and predictive accuracy, for the fitted Markov chain models.

Findings and contributions. In Walk et al. [2014b] I show that the transition matrices of first-order Markov chain models can be used to identify different user-roles and that edit workflows are influenced by the ontological structure, as users exhibit a preference towards consecutively editing closely related classes. Further, colleagues and I found out that PrefixSpan can be used to mine larger sequential patterns (with sufficient support) from the change-logs of collaborative ontology-engineering projects in Walk et al. [2014a]. Subsequently, after demonstrating the existence of regularities and sequential patterns, the analysis continues by modeling and predicting future actions of contributors using Markov chains of varying orders. The conducted prediction experiments have shown that multiple models of higher orders were better suited for predicting aspects of future actions of contributors than first-order models.

RQ3: How can we explain sequential edit patterns in collaborative ontology-engineering projects?

Problem. This research question aims at finding explanations for the identified sequential edit patterns by further analyzing the social dynamics that occur when users collaboratively engineering an ontology. In particular, the results of previous empirical investigations and analyses [Walk et al., 2014b,a, 2015b] already led to possible explanations for the identified patterns. Hence, this research question aims at determining which of these explanations best describes the analyzed logs of changes and thus, the extracted edit patterns of the corresponding ontology-engineering project. In Singer et al. [2015], the authors present HypTrails, a coherent research framework that allows comparing hypotheses about sequential trails of users. For example, if users edit an ontology, they leave a chronologically ordered sequence of edited classes. In this particular case, hypotheses represent beliefs about how users select which class to edit next. One such hypothesis could be that users always select classes in close proximity to the previously changed class. Using HypTrails it is possible to compare different hypotheses to identify the one that best describes the creation of the change-logs, and thus the extracted patterns and the development of the resulting ontology at hand.

Approach. Previous empirical analyses conducted in Walk et al. [2014b,a, 2015b] already indicated that users exhibit patterns and regularities when collaboratively engineering an ontology. Using these identified patterns and their interpretations in previous works, colleagues and I formulated hypotheses—based on the hierarchical structure (*isA* relationships) and the content of the classes of the ontology—about how users collaboratively edit and engineer ontologies to further evaluate the empirical findings. Finally, colleagues and I used HypTrails to investigate the plausibility of the hypotheses for explaining the change-logs, and thus the extracted patterns and the resulting ontologies of the different projects.

Findings and contributions. The results presented in Walk et al. [2015a] indicate that the structural representation of an ontology, inferred from *isA* relationships, has a consistently high influence on the editing behavior of contributors across different versions of tools and projects and confirm the results of previous empirical analyses. Additionally, users exhibit a tendency to consecutively edit classes with similar property values and/or in close proximity to each other.

1.5 Main Publications

This cumulative thesis consists of the following publications:

- **Article 1:** [Walk et al., 2015b] Walk, S., Singer, P., Strohmaier, M., Helic, D., Noy, N. F., and Musen, M. A. (2015). How to Apply Markov Chains for Modeling Sequential Edit Patterns in Collaborative Ontology-Engineering Projects. In *International Journal of Human–Computer Studies.*, 84:51–66

- **Article 2:** [Walk et al., 2014b] Walk, S., Singer, P., Strohmaier, M., Tudorache, T., Musen, M. A., and Noy, N. F. (2014). Discovering Beaten Paths in Collaborative Ontology-Engineering Projects using Markov Chains. In *Journal of Biomedical Informatics*, 51:254–271

- **Article 3:** [Walk et al., 2014a] Walk, S., Singer, P., and Strohmaier, M. (2014). Sequential Action Patterns in Collaborative Ontology-Engineering Projects: A Case-Study in the Biomedical Domain. In *International Conference on Information and Knowledge Management*, pp. 1349–1358

- **Article 4:** [Walk et al., 2015a] Walk, S., Singer, P., Noboa, L. E., Tudorache, T., Musen, M. A., and Strohmaier, M. (2015). Understanding How Users Edit Ontologies: Comparing Hypotheses About Four Real-World Projects. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference Proceedings, pages 551—568*

1.6 Contributions and Implications

The majority of contributions of this thesis consist of empirical investigations of sequential patterns, as well as the investigation of different hypotheses about how users collaboratively develop an ontology.

In particular, the main contributions of this thesis are three-fold:

- First, this thesis provides a detailed description of the process on how to apply Markov chains to identify, extract, model, visualize and interpret sequential edit patterns.
- Second, this work presents a detailed empirical investigation of patterns and regularities in different collaborative ontology-engineering projects using autocorrelation, an adapted version of the runs test to work with categorical data and the pattern mining algorithm PrefixSpan. Additionally, this thesis presents analyses that use Markov chain models of varying orders to fit and model sequential patterns in the change-logs of different collaborative ontology-engineering projects from the biomedical domain. A strong focus was set on empirically exploring, discussing and interpreting the transition matrices of first-order Markov chains, as well as showcasing the usefulness of Markov chains for predicting various aspects of actions that contributors are most likely to conduct next.
- Third, this thesis presents an analysis aimed at explaining the observed sequential edit patterns by analyzing and comparing hypotheses about how users collaboratively engineer ontologies. The different hypotheses are defined as beliefs, inferred from the results of previous empirical analyses and represented by weighted first-order Markov chain transition matrices. Using HypTrails, the hypotheses are then ranked according to their relative plausibility and compared against each other.

The results of the empirical analyses presented in this thesis indicate that the investigation of sequential edit patterns in collaborative ontology-engineering projects represents an important stepping stone not only towards a better understanding of how users collaboratively engineer ontologies but also towards more efficient ontology editor designs.

1.7 Structure of this Thesis

In the remainder of this thesis I first discuss related work in Chapter 2. In particular, I first focus on the introduction of (collaborative) ontology-engineering methodologies, ontology development tools as well as approaches for ontology evaluation and collaborative ontology engineering in Section 2.1. In Section 2.2 I elaborate on the methods used in this thesis to study regularities and patterns in collaborative ontology-engineering projects.

Chapter 3 builds the main body of this cumulative thesis and contains the main publications as described in Section 1.5. Further, Section 3.1 outlines my contributions to each article. See Figure 1.2 for a graphical illustration regarding the main publications and which research questions they are tackling.

I conclude this thesis in Chapter 4 by first providing a summary of the obtained results and contributions in Section 4.1, and implications as well as potential applications in Section 4.2. In Section 4.3 I briefly outline the limitations of the analyses presented in this thesis and discuss future work in Section 4.4.

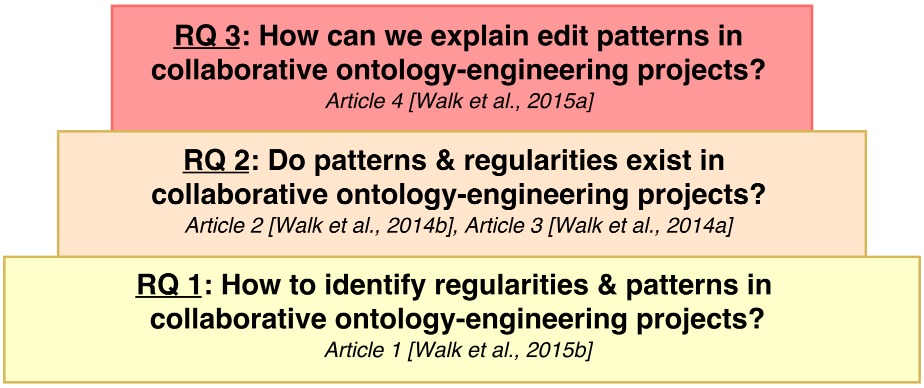


Figure 1.2: **Structure of this thesis.** This figure provides a brief overview of the structure and context of the main articles with the corresponding research questions. The foundation of this thesis, RQ 1, deals with a detailed description and analysis of how to identify usage patterns in collaborative ontology-engineering projects. The second research question (RQ 2) tackles the problem of actually identifying, extracting, modeling and leveraging such usage patterns in different, real-world collaborative ontology-engineering projects. Finally, RQ 3 addresses the problem of explaining the extracted usage patterns by testing hypotheses about how users collaboratively edit ontologies, inferred from the results of the empirical analyses conducted to answer RQ 1 and RQ 2.

Table 1.1: **Context of main articles.** This table lists the main articles of this thesis, as well as their research questions, topics and main contributions.

Article	RQ	Topic	Main Contribution
Article 1 [Walk et al., 2015b]	RQ 1	analyzing and modeling sequential patterns	Description of the process of how to apply Markov chains on collaborative ontology-engineering projects.
Article 2 [Walk et al., 2014b]	RQ 2	analyzing sequential patterns	Analysis of first-order Markov chain models in the context of each investigated project.
Article 3 [Walk et al., 2014a]	RQ 2	extracting and predicting edit actions using sequential patterns	Demonstration of sequential pattern analysis and prediction experiment.
Article 4 [Walk et al., 2015a]	RQ 3	formulating and testing hypotheses	Hypotheses, inferred from empirical analyses, are tested to determine which best describe how users edit ontologies.

2 Related Work

This chapter is intended to provide a high-level overview of all topics related to the analyses presented in this thesis. To that end, Section 2.1 discusses research from the domain of ontology engineering while Section 2.2 covers work related to the identification and analysis of regularities and (sequential) patterns in data. Finally, in Section 2.3, I review literature related to (online) peer production systems.

2.1 Ontology Engineering

There exist many different definitions for what ontologies actually represent. In the context of Computer Science, the most prominent definition for ontologies is based on Gruber [1993], who first defined an ontology as an "explicit specifications of a conceptualization" in 1993. The explicit specification refers to a formal definition and description (i.e., defined and written in a common language) that represents an abstraction of entities and concepts of the real world (the conceptualization). A few years later in 1997, Borst [1997] rephrased this definition to a "formal specification of a shared conceptualization", meaning that the concepts and relationships modeled by the ontology should represent a shared view (i.e., representing consensus) on the modeled facts. Finally, Studer et al. [1998] combined these two definitions, stating that ontologies should represent not only (standardized and) formalized data structures but also explicit and unambiguous specifications of a shared view on a specific domain.

Today, ontologies represent a specific type of data structures, written in dedicated programming and markup languages, such as the Resource Description Framework (RDF) Schema [Brickley and Guha, 2004] or the Web Ontology Language (OWL) [Patel-Schneider et al., 2004], which (among other things) allow computers to "understand" and reason on relationships between as well as properties of the entities modeled in the ontology.

Particularly in the biomedical domain, (large-scale) ontologies see widespread use for knowledge representation. For example, the 11th revision of the International Classification of Diseases, an ontology that models diseases and causes of death, is currently developed by Stanford University and the World Health Organization [Tudorache et al., 2010]. Its predecessor, the International Classification of Diseases in its 10th revision, is used to encode patient history data and diseases as well as to compile health-related statistics in many countries around the world.

2.1.1 Methodologies & Best-Practices

The whole process of creating an ontology already represents a very complex task. First, authors of an ontology have to determine the level of granularity and abstraction they want to pursue and model. Second, the resulting ontology should represent an unambiguous and shared view of the modeled domain. In general, research in the field of ontology engineering comprises all the tasks, actions, tools and processes required for developing ontologies [Gomez et al., 2004]. As a result, many researchers and practitioners were interested in developing guidelines, methodologies and best practices for engineering ontologies. CommonKADS [Schreiber et al., 1994] or TOVE [Grüninger and Fox, 1995] are among the first methodologies that explicitly tackle the problems associated with ontology engineering and provide first guidelines for creating ontologies. In Fernández-López et al. [1997] the authors propose the METHONTOLOGY methodology, which describes a (well-structured) set of activities to develop an ontology from scratch, including descriptions of activities and their context in the ontology life cycle. In 2001, Noy et al. [2001] provide practical guidelines—in the form of an iterative process (enumerated steps)—that outline how an ontology could be created.

Due to the increasing complexity of ontologies and a general shift towards distributed and collaborative ontology-engineering projects, researchers started to investigate and describe best practices for such projects. One of the first methodologies, which specifically describes and tackles the problems of collaborative ontology engineering, was presented by Holsapple and Joshi [2002], which is referred to by the authors as an initial descriptive framework of knowledge manipulation activities. Two years later, in 2004, the DILIGENT (distributed, loosely-controlled and evolving engineering of ontologies) methodology was first presented by Pinto et al. [2004]; Davies et al. [2005] and aims at enhancing the collaborative ontology-engineering process by augmenting and facilitating interactions between ontology and domain experts. To that end, they define five main activities, which are targeted towards local adaptions of the ontology by everyone involved in the ontology life cycle and one central group of users responsible for analyzing and merging the conducted local adaptions. In 2007 Tempich et al. [2007] conducted a detailed case study using the DILIGENT methodology in combination with an adapted JSPWiki to create an ontology and discuss the advantages and pitfalls of the chosen method. In particular, the results of their experiment suggested that the ontology developers were able to identify and reach consensus on conflicting topics very fast and non-experts were able to quickly pick up the editor and start working on the ontology. However, if advanced primitives and constructs are required (e.g., axioms) other tools might provide better support, albeit more expert-knowledge is required to properly use these tools. CICERO [Dellschaft et al., 2008], an extension to the Semantic MediaWiki[1], follows a similar approach and augments user discussions and documentation as well as efficiency by supporting the design rationale of ontology engineers and is also based on main principles of DILIGENT.

A similar approach was taken by Kotis and Vouros [2006], when they presented the Human-Centered Ontology-Engineering Methodology (HCOME). The main goal of HCOME was to empower and actively involve the knowledge workers—domain experts responsible for creating the shared conceptualization—and ontology developers in the ontology lifecycle by establishing and including them in a feedback-loop. In contrast, the NeOn methodology [Suárez-Figueroa et al., 2012; Gómez-Pérez and Suárez-Figueroa, 2009] describes a total of 9 different scenarios (among

---

[1] `http://semantic-mediawiki.org`

other things), reflecting real-world situations in (collaborative) ontology-engineering projects and how to deal with them. In Debruyne et al. [2010], the authors suggest that communities will be more efficient in their endeavor to collaboratively engineer an ontology if they can use natural language, as opposed to descriptive languages such as RDF. To that end, they propose GOSPL, a fact-oriented hybrid ontology-engineering method.

Sharing many commonalities with eXtreme programming [Beck, 2000]—a well-known agile software development methodology—Auer [2006] presented RapidOWL. In contrast to other ontology-engineering methodologies, RapidOWL does not commit to a predefined process model. Instead, the guidelines provided by RapidOWL concentrate on adding the feedback of the domain experts as fast as possible into the development process of the ontology.

However, once an ontology is engineered it still needs to be maintained and updated with new information (if necessary), meaning that already existing concepts and properties have to be edited and might become obselete. A framework for the task of ontology evolution—essentially representing the task of maintaining an ontology and keeping already existing information up to date—was discussed and proposed by Noy et al. [2006] in 2006. Further, Braun et al. [2007] proposed an ontology maturing framework that describes the importance of the consensus finding processes of the users for advancing, evolving and maturing an ontology.

2.1.2 Ontology Development Tools

The Semantic Web community has developed a number of tools aimed at supporting the collaborative development of ontologies. For example, Protégé, and its versions for collaborative ontology development, such as WebProtégé [Tudorache et al., 2013], iCAT [Tudorache et al., 2010] and Collaborative Protégé [Tudorache et al., 2008], are prominent standalone tools that are used by a large community worldwide to develop ontologies for a variety of different projects. WebProtégé and Collaborative Protégé have proven to be robust and scalable environments for the collaborative development of ontologies and are still used in multiple (large-scale) projects today, including the development of ICD-11 [Tudorache et al., 2010].

One of the first ontology tools that emphasized the distributed development of ontologies was WebOnto [Domingue, 1998], a web-based ontology editor that also includes a customized web server and allows for browsing and editing ontologies over the Internet. Another early system that enabled collaborative ontology engineering was the Ontolingua server [Farquhar et al., 1997], which allowed multiple users to work on the same ontology and sent updates to users if the ontology was changed. Further, Semantic MediaWikis [Krötzsch et al., 2006] and some of its derivatives, such as OntoWiki [Auer et al., 2006], IkeWiki [Schaffert, 2006] and MoKi [Ghidini et al., 2009], have added semantic as well as ontology modeling capabilities and collaborative features to traditional MediaWiki systems. In particular, the developers of OntoWiki specifically concentrated on adding semantic capabilities to support the acquisition of instance data rather than concentrating on the development of the ontology or schema itself. On the other hand, MoKi is implemented as an extension to the Semantic MediaWiki and includes multiple features that allow for rich ontology modeling and editing in a collaborative manner and has already been deployed in a number of real world projects. Another commercial editor, built on top of a Wiki, is Knoodl[2]. The main intent of this editor was to combine the structure of an ontology with a free-text Wiki page allowing for increased searching capabilities and potential linking to SPARQL endpoints. Soboleo [Zacharias and Braun, 2007], a tool developed to implement and augment the knowledge maturing methodology, and PoolParty [Schandl and Blumauer, 2010] both support lightweight editing of taxonomies and focus on providing services that take advantage of these vocabularies. For example, annotating or tagging resources, faceted browsing, or semantic search. OntoEdit [Sure et al., 2002] represents another ontology editor that actively augments and supports interactions and consensus finding between users, particularly during early stages of collaborative ontology-engineering projects. To ensure consistency the

---

[2] `http://knoodl.com`

editor blocks the parts of the ontology, which are currently edited by users.

Pöschko et al. [2012], and Walk et al. [2013] have created and extended *PragmatiX*, a tool to browse aspects of the history of collaboratively engineered ontologies. PragmatiX also provides quantitative insights, which allow for an easier monitoring of the progress of collaborative ontology-engineering projects.

2.1.3 Collaborative Ontology Engineering

The majority of literature about collaborative ontology engineering sets its focus on surveying, finding and defining requirements for the engineering process and/or tools used in these projects [Noy and Tudorache, 2008; Groza et al., 2013]. Simperl and Luczak-Rösch [2014] provide an exhaustive overview of different collaborative ontology-engineering methodologies and tools.

Researchers recently started to investigate the history and evolution of collaborative ontology-engineering projects to shed light into the intricate and dynamic processes that occur while distributed users or teams collaboratively develop an ontology and their implications for the resulting ontology. For example, Falconer et al. [2011] investigated if contributors of collaborative ontology-engineering projects exhibit specific roles, and if these roles can be used to group and classify these users when contributing to the ontology. Strohmaier et al. [2013] investigated the hidden social dynamics that take place in collaborative ontology-engineering projects from the biomedical domain and provided new metrics to quantify various aspects of the collaborative engineering processes. Pesquita and Couto [2012] showed that the location and specific structural features can be used to determine if and where (i.e., classes) the next change is going to take place in the Gene Ontology[3]. To analyze user editing patterns, Wang et al. [2013] used association-rule mining on the change-logs of collaborative ontology-engineering projects and conducted a prediction experiment to showcase the utility of the identified editing patterns.

---

[3] `http://www.geneontology.org`

In Debruyne and Nijs [2013], the authors present a generic reputation framework to identify leaders in collaborative ontology-engineering projects. They make use of different characteristics (referred to as reputation sensors) to classify users as leaders, such as activity, engagement quality as well as features of the social interaction graph. To evaluate their suggested framework, the authors compare their automatically calculated results with the results of a user survey and find overlaps. Similarly, yet with a different scope, De Leenheer et al. [2009] use different social performance indicators to monitor and better understand the constantly changing social arrangement and interactions of collaborative ontology-engineering projects. For example, the circumstance that users are heavily/not at all engaged in discussions about classes but very little (formal) content is added or edited, might indicate that the required expertise for adding meaningful contributions is missing. Hence, the authors argue that the discussions represent one important source of information and should be actively considered when monitoring and managing the collaborative ontology-engineering process.

Recently, Van Laere et al. [2014] used k-means and the GOSPL methodology to classify users by analyzing and clustering the different interactions that users engage in while collaboratively working on engineering an ontology. In Di Francescomarino et al. [2014] the authors conducted a theoretical analysis and multiple user studies to investigate if and which collaborative features for ontology authoring of the MoKi have an impact on the ontology life cycle and the entities of the resulting ontology.

Gil and colleagues [Gil et al., 2013; Gil and Ratnakar, 2013] conducted multiple empirical analyses regarding different aspects of a number of publicly available instances of Semantic MediaWikis. They were particularly interested in the study of the evolution of semantic features, such as properties and concepts, in these Wikis. Among other things, they found out that categories are still more widely used than concepts. In Gil et al. [2015], the authors present the Provenance Bee Wiki[4], a Wiki that aggregates, analyses and presents data and statistics about publicly available Semantic MediaWiki communities.

---

[4] `http://skc.isi.edu/provenancebeewiki`

Wikidata [Vrandečić, 2012; Vrandečić and Krötzsch, 2014]—another sophisticated collaborative ontology-engineering endeavor, initiated by the Wikimedia Foundation—is gathering structured data in multiple languages to represent facts in articles, linking to and between Wikipedia and its different language editions.

2.1.4 Ontology Evaluation

Due to the increasing importance of ontologies, researchers have come up with many different approaches for assessing, testing, measuring and evaluating the quality of ontologies. However, as ontologies represent shared conceptualizations, which are very hard to validate in the first place, automatically assessing the quality of ontologies still represents an open problem. In general, Brank et al. [2005] identified a total of four different techniques for ontology evaluation, which were used by other researchers to assess the quality of an ontology:

1. Maedche and Staab [2002] used a **“golden standard”** to compare their ontologies against by calculating different measures of semantic similarity. This approach for ontology evaluation will only work if such a “golden standard” is available.

2. Porzel and Malaka [2004] suggest to evaluate ontologies via an **application based** approach. To that end, the ontology under question is used for a specific task, where the success or accuracy of the results—also referred to as fitness of an ontology—can be measured and compared against other ontologies.

3. Similar to the previous approach Brewster et al. [2004] suggest to evaluate ontologies by assessing the similarity or **ontological “fit”** with a related text corpus.

4. If neither option is possible, **manual evaluation**, as a last-resort, has to be conducted [Mika and Alani, 2005]. This typically involves a user study with human subjects, who compare and measure aspects of an ontology against a predefined set of requirements or metrics.

Researchers also investigated the potential of online crowdsourcing platforms for ontology engineering and evaluation [Sarasua et al., 2012; Noy et al., 2013]. Mortensen et al. [2013] suggested that crowdsourcing can represent a viable tool for ontology developers to quickly audit particularly large-scale ontologies. However, generating and evaluating micro tasks for crowdsourcing workers represents a rather complex and time-consuming task. Further, crowd-based evaluations, due to the nature of micro tasks, are usually conducted for very specific problems, such as hierarchy verification or ontology alignment. Although, Mortensen et al. argue that the benefits of these crowd-based approaches outweigh their limitations and should be considered by ontology developers when presented with the task of evaluating (large-scale) ontologies.

Aside from evaluating the content and the purpose of an ontology, it can also be beneficial to evaluate the consistency of an ontology in terms of the defined axioms used for reasoning tasks [Haase and Qi, 2007; Lam, 2007]. According to Sabou et al. [2007], automatically extracting information and data from the semantic web represents another method for creating automatic task-based evaluations, which can be used to assess and evaluate the quality of ontologies. Obrst et al. [2007] first surveyed state-of-the-art evaluation techniques before concluding that ontology evaluation should be an integral part of all ontology-engineering projects, particularly in the early stages (i.e., during the engineering and development processes). Similarly, Neuhaus et al. [2013] present an adapted version of the ontology life cycle that includes evaluation tasks to assess the quality of an ontology during all stages of the life cycle. The authors of Poveda-Villalón et al. [2014] conducted an empirical analysis of over 693 ontologies and identified and classified the extracted pitfalls within all these projects. Further, they present OOPS!, an abbreviation for Ontology Pitfall Scanner!, which is a tool for (automatically) detecting such pitfalls.

2.2 Regularities & Patterns

To broaden our understanding of the dynamic and ongoing processes in collaborative ontology-engineering projects, I am particularly interested in investigating sequences or patterns of actions that are commonly performed by contributors of such projects. Hence, I will first review literature regarding sequential pattern mining in Section 2.2.1 and elaborate on Markov chain models in Section 2.2.2.

2.2.1 Sequential Pattern Mining

Agrawal and Srikant [1995] first discussed and addressed the problem of mining sequential patterns from text or from large databases. To that end, the authors first defined the task of sequential pattern mining as "discovering all sequential (chronologically ordered) patterns in a collection of chronologically ordered sequences, weighted according to the number of sequences that contain these patterns". The weights that are used to rank the importance of the identified patterns is also referred to as support and represents the fraction of sequences that exhibit the corresponding pattern. For example, a pattern with a support of 25% is present in 25% of all investigated sequences.

Further, Agrawal and Srikant also introduced the first *a priori* sequential pattern mining algorithms AprioriAll and AprioriSome. In general, a priori algorithms first generate and calculate support for all possible sequential patterns. AprioriAll and AprioriSome only differ in the way the patterns are generated. The former incrementally generates and calculates support for all possible patterns. The latter concentrates on patterns of a given length and only proceeds if no pattern with a predefined minimum support is found. Other examples of a priori algorithms are introduced by Ng et al. [1998]; Sarawagi et al. [1998]. One year later, Srikant and Agrawal [1996] introduced the widely used generalized sequential pattern algorithm (GSP), which considers time-constraints and sliding windows for candidate generation. Further, the authors showed that specific patterns are limited in their occurrences to the number of occurrences of any sub-pattern of this pattern. Many additional examples of a priori algorithms have been reviewed and discussed in literature [Mannila et al., 1997; Wang et al., 1994; Bettini et al., 1996; Garofalakis et al., 1999; Masseglia et al., 1998], with SPADE [Zaki, 2001] being one of the most prominently used and referred to algorithms.

As mentioned before, a priori algorithms create a very large (in the worst case exponential) set of candidates, which drastically increases the time required for calculating the different support values. Hence, Han et al. [2000] invented the so-called FP-growth algorithm that is based on the pattern-growth methodology, which naturally limits candidate generation. A practical implementation of this algorithm can be found in Borgelt [2005]. Due to the advantages in processing time and memory of the methodology, many algorithms adopted the pattern-growth approach and refined it, such as PrefixSpan [Pei et al., 2001]. In particular, PrefixSpan circumvents the exponential candidate generation by strategically expanding (spanning) found patterns and systematically rejecting patterns that are not present in the data. Further, PrefixSpan is a very special pattern-mining algorithm, as it does not allow for patterns to exhibit gaps between elements of the patterns.

Many researchers have adapted these pattern mining algorithms and approaches for different domains. For example, Hsu et al. [2007] used sequential pattern mining to identify hot regions in protein-protein interactions in the biomedical domain. In Perera et al. [2009], the authors extract sequential patterns from a software development project to identify and analyze the differences in terms of patterns—and thus factors of success—of better and weaker groups. Further, researchers have also tried to use pattern mining algorithms for compression purposes, for example, for web graphs [Buehrer and Chellapilla, 2008], or to learn more about how users interact with websites in general by mining and analyzing access patterns from change logs [Pei et al., 2000].

2.2.2 Markov Chain Models

Markov [2006] first introduced the notion of a Markov chain in 1903 and applied them for calculating and modeling the most likely sequences of vowels and consonants in the Russian novel *Eugene Onegin*.

In general, a Markov chain consists of a finite state-space S where each state $s_1, s_2, ..., s_n \in S$ with $n = |S|$ and a transition matrix P, which lists all probabilities p_{ij} to traverse from the state s_i to the state s_j and $\sum_j p_{ij} = 1$.

For example, if the state space of one specific Markov chain consists of all the classes of an ontology, the values listed in P reflect the probabilities of transitioning between the different states (e.g., to browse or edit the content of the corresponding classes).

Whenever the succeeding state of a sequence only depends on the current state, as opposed to multiple previous states, the process (or sequences) modeled by the corresponding Markov chain exhibits the Markovian property. This is also referred to as a first-order Markov chain and can be formally defined as

$$P(X_{t+1} = s_j | \underbrace{X_1 = s_{i_1}, ..., X_{t-1} = s_{i_{t-1}}, X_t = s_{i_t}}_{\text{all previous transitions}}) =$$
$$P(X_{t+1} = s_j | \underbrace{X_t = s_{i_t}}_{\text{current transition}}) = p_{ij}.$$

Analogously, Markov chains can not only be used to model the transition probabilities between a set of states for 1 transition in a row, but also for multiple transitions or higher orders. This means that the next, most likely state to occur, does not only depend on the current state but on a sequence of k previous states as well. Meaning that the state space for a second-order ($k = 2$) Markov chain increases, as all permutations of possible states up to length k have to be included in the state space as well. This leads to the following formal definition:

$$P(X_{t+1} = s_j | \underbrace{X_1 = s_{i_1}, ..., X_{t-1} = s_{i_{t-1}}, X_t = s_{i_t}}_{\text{all previous transitions}}) =$$
$$P(X_{t+1} = s_j | \underbrace{X_{t-k+1} = s_{i_{t-k+1}}, ..., X_t = s_{i_t}}_{\text{k transition}})$$

In this thesis, higher order Markov chains are converted to first-order Markov chains by making use of compounded states [Chierichetti et al., 2012a], meaning that the single compounded state (s_1, s_2) consists of the states s_1 and s_2. Ultimately the state space of a k^{th}-order Markov chain consist of $n^k n$ states.

In the past, Markov chains have seen widespread use for analyzing aspects of the navigational behavior of users on the web. For example, Borges and Levene [2007]; Lempel and Moran [2000]; Pirolli and Pitkow [1999] have analyzed variable order Markov chains to model the browsing behavior of users and to determine if and to what extent visited pages of a website determine which page is most likely visited next by a user. On the other hand, Sen and Hansen [2003]; Zukerman et al. [1999]; Deshpande and Karypis [2004]; Singer et al. [2013]; Lamprechta et al. [2014] used Markov chains to predict pages of a website or actions most likely visited/conducted by users next.

One assumption, which is frequently used in literature and also modeled by the Random Surfer model that builds the foundation of PageRank [Brin and Page, 1998], is that the navigation of users on the Web is Markovian, meaning that only the current state contains predictive information about the next state. Only recently, Chierichetti et al. [2012b] conducted an analysis that questioned if a first-order chain best represents the navigation behavior of humans on the Web and suggested that the Markovian assumption might not hold.

As mentioned before, one peculiar problem assigned to higher order Markov chains is their increased complexity in terms of an increased state space. Borges and Levene [2000]; Pirolli and Pitkow [1999] studied the appropriateness of higher order Markov chains to model navigational behavior on websites but found out that the benefit of such higher order models does not outweigh the complexity. In Singer et al. [2014], the authors presented multiple model selection techniques, which can be used as a solid guideline to detect and select the appropriate Markov chain order for a given task. The analyses presented in this thesis make use of these selection techniques to determine the most appropriate Markov chain models for modeling (different aspects of) edit actions in collaborative ontology-engineering projects.

In Singer et al. [2015] the authors present HypTrails—a generic methodology that allows researchers to compare and rank hypotheses about digital trails on the web. In particular, HypTrails models digital trails as a first-order Markov chain and represents hypotheses as priors. In this thesis I have made use of HypTrails to compare multiple hypotheses about how users collaboratively develop ontologies.

2.3 Peer Production Systems

The Internet, particularly since the Web 2.0 movement, provides users with the means to participate and collaborate on a variety of different online peer production systems. Analogously to collaborative ontology engineering, users engage in various cooperative activities, such as writing and editing articles for online encyclopedias (e.g., Wikipedia), asking and answering questions on question answer portals (e.g., StackOverflow[5]) or collecting and annotating various online resources (e.g., links on del.icio.us[6]). Research on these collaborative peer production systems has in part focused on developing methods and studying factors that (i) improve the quality of the collaboratively developed artifact, (ii) increase user participation or (iii) explain certain social phenomena.

In fact, research focusing on improving software quality by developing process-oriented models and methods has a long tradition in the field of software development. For example, the main idea of the well-known sequential waterfall model [Royce, 1970] was to provide a structured development process in order to increase the quality of the final product (the software) itself. In 1968, Zurcher and Randell [1968] first discussed agile software development methods—a very prominent software development paradigm that specifically provides mechanisms and guidelines, which are intended to shift attention from the final product to the management of the development process. The early 1990s saw a proliferation of suggestions to

---

[5] `http://stackoverflow.com`
[6] `https://delicious.com`

elevate the production process in software development projects using agile programming methods, such as Scrum [Schwaber, 1995], Crystal Clear [Cockburn, 2004], Extreme Programming [Beck, 1999; Beck et al., 2001] or Feature Driven Development [Coad et al., 1999; Palmer and Felsing, 2002].

To increase user participation, Cabrera and Cabrera [2002] introduced the notion of pay-off functions. In particular, administrators of online peer production systems should restructure these pay-off functions by either reducing the (perceived) costs assigned with contributing to a system or by increasing the benefits for each contribution. The latter can involve rewards for either single contributions, which are tied to the users that performed them, or for the quality of the collaboratively engineered product itself, equally distributed across all participating users. Further, if the perceived usefulness of a contribution to a system is increased, for example by implementing mechanisms that provide feedback if a resource was helpful, the perceived efficacy increases. Additionally, when providing an environment that encourages and promotes frequent interactions between its users over a long period of time, some form of group identity can be established and even promote personal responsibility. Tackling these problems can help to not only improve the quality and frequency of contributions, but potentially increases the overall quality of the collaboratively developed artifact in general. Cabrera and Cabrera [2002] further suggested, that it is possible to resolve problems common to peer production environments, such as the *free-riding* and *ramp-up* problems, by restructuring the pay-off functions. The free-riding problem characterizes the fact that users would rather enjoy a resource than contribute to it, which could be alleviated by strengthening the group identity or increasing the perceived usefulness of contributions. On the other hand, the *ramp-up* problem describes the issue of motivating users to contribute to a system when only limited amounts of either content or activity (or both) are present in the system, which could be tackled by reducing the efforts required to add content.

Kittur et al. [2007] analyzed how activity is distributed across users in Wikipedia and del.icio.us, two well-known peer production systems. The authors found out that participation across users during the initial starting phase is unevenly distributed, resulting in few users (administrators) with very high participation and contribution rates, while the rest of the users (common users) conduct only a very small number of contributions. However, due to steadily increasing numbers of common users in these peer production systems over time, the total number of changes of this group of users increases as well. Finally, the overall number of contributions by common users will surpass contributions from administrators, albeit each common user only conducts a very small number of contributions individually. This shift in participation indicates that analyzing patterns in activity [Kittur and Kraut, 2008] can provide meaningful insights into different aspects of the engineering process. In Wilkinson and Huberman [2007], the authors conducted an analysis that showed that the number of distinct users who work on an article in Wikipedia correlates with the quality of the corresponding article. Further, Keegan et al. [2011] analyzed patterns of activity and coverage on Wikipedia following the Tõhoku earthquake, while Shachaf [2010] investigated instances and occurrences of vandalism and sabotage in Wikipedia articles.

Suh et al. [2009] have discovered that the number of active editors and newly created articles in Wikipedia is saturating and slowly declining, while Halfaker et al. [2011] analyzed if reverts—actions that reverse and delete one or more of the preceding contributions to Wikipedia articles—have an impact on the contribution rates of new Wikipedia editors.

Solomon and Wash [2014] recently argued that it is still not clear which features of an online community characterize critical mass—representing a state where an online peer production system has become self-sustaining—for the whole system. Two very intuitive approximations for critical mass, which were used by Solomon and Wash, are activity and community growth in WikiProjects. Activity for WikiProjects is represented by the aggregated number of revisions over all articles, while community growth is represented as the total number of registered and unique users that have contributed at least one change to a WikiProject. In particular, the authors argue that activity and community growth are very good representatives for critical mass, as systems that lack either one are more likely to become inactive than to become self-sustaining.

In Walk and Strohmaier [2014], the authors have adopted the same approach for characterizing critical mass to shed light into the complex dynamics that drive activity and community growth in Semantic MediaWiki communities. In general, once activity and user diversity are increasing, critical mass is likelier to follow, as interest (and pay-off) increases, reinforcing increases in user contributions until the maximum potential of a system is reached. However, it is also possible that this self-sustaining state will only last for a very short amount of time, depending on a variety of different factors, making it particularly hard for certain Semantic MediaWikis to keep such a self-sustaining state.

Recently, Ribeiro [2014] analyzed the daily number of active users, who visit specific websites. To that end, the author fitted a model that allows to determine if a website reached self-sustainability. This self-sustainability is defined by the shape of the curve of the daily number of active users over time. To that end, two constants α and β are used. α represents the rate of active members influencing inactive members to become active again. Analogously, β describes the rate of an active member to spontaneously switch state and become inactive. In the presented model, whenever $\frac{\beta}{\alpha} \geq 1$, a website was classified as unsustainable and without (external) intervention the daily number of active users will likely converge towards zero. On the other hand, if $\frac{\beta}{\alpha} < 1$ and the number of daily active users is initially higher than the asymptotic one, a website was categorized as a self-sustaining website.

3 Papers

3.1 Contributions to Main Articles

The following section lists all of *my* contributions to the main publications of this cumulative thesis.

- [Walk et al., 2015b] Walk, S., Singer, P., Strohmaier, M., Helic, D., Noy, N. F., and Musen, M. A. (2015). How to Apply Markov Chains for Modeling Sequential Edit Patterns in Collaborative Ontology-Engineering Projects. In *International Journal of Human–Computer Studies.*, 84:51–66

I was the main author of this article and responsible for the conceptualization and description of the whole process of how to apply the Markov chain framework on the change-logs of collaborative ontology-engineering projects, the experimental design as well as the interpretation of the results. In particular, this involved the application of the Markov chain framework on the change-log of one large-scale collaborative ontology-engineering project, including all steps associated with the preprocessing and cleaning of the investigated dataset, the practical execution of the experiments, the development of the research questions and the discussion of the results.

The Markov chain framework that I have used in this publication was mainly developed by Philipp Singer in cooperation with Denis Helic and Markus Strohmaier. The idea for this article originated from discussions between Markus Strohmaier, Philipp Singer and me. All authors of this article were involved in writing the paper, discussing and interpreting the (intermediate) results and provided feedback for the experimental design.

- [Walk et al., 2014b] Walk, S., Singer, P., Strohmaier, M., Tudorache, T., Musen, M. A., and Noy, N. F. (2014). Discovering Beaten Paths in Collaborative Ontology-Engineering Projects using Markov Chains. In *Journal of Biomedical Informatics*, 51:254–271

I was the primary author of this article, responsible for the experimental design as well as the execution and the description of the conducted analyses. The interpretation of the results was mainly done by me in close coordination with Natalya Noy, Tania Tudorache and all authors of this article. Further, I collected, preprocessed, cleaned, and prepared the investigated datasets.

The Markov chain framework, used throughout all analyses of this article, was mainly developed by Philipp Singer. All authors of this article were involved in writing the paper.

- [Walk et al., 2014a] Walk, S., Singer, P., and Strohmaier, M. (2014). Sequential Action Patterns in Collaborative Ontology- Engineering Projects: A Case-Study in the Biomedical Domain. In *International Conference on Information and Knowledge Management*, pp. 1349–1358

For this article I designed and executed the experiments to detect sequential patterns using a pattern-mining algorithm and to apply Markov chains on the logs of changes to model sequential patterns. Further, I conducted the prediction experiments presented in this article and devised the research questions. The results of this article were mainly interpreted by Philipp Singer and me. Additionally, I was responsible for collecting, preprocessing, cleaning, and preparing the investigated datasets.

The design of the Markov chain framework, which was used in this article to model sequential patterns and to determine the appropriate order of the Markov chain models, was developed by Philipp Singer. The tests conducted to study randomness and regularities were also conducted by Philipp Singer. All authors contributed to the writing of the paper and the interpretation of the results.

- [Walk et al., 2015a] Walk, S., Singer, P., Noboa, L. E., Tudorache, T., Musen, M. A., and Strohmaier, M. (2015). Understanding How Users

> Edit Ontologies: Comparing Hypotheses About Four Real-World Projects. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference Proceedings, pages 551—568*

I was the main author for this article and responsible for devising and formulating the different hypotheses, developing the experimental setup and conducting all experiments. Further, I collected, preprocessed, cleaned and prepared the investigated datasets. Philipp Singer, Denis Helic, Andreas Hotho and Markus Strohmaier developed the framework that was used to systematically compare and rank the different hypotheses. The graphical representations of the sample-ontologies were created by Lisette Espín Noboa.

The ideas for this paper stem from multiple discussions between all authors of the paper. The description and interpretation of the results was mainly done by me in close coordination with Tania Tudorache and Philipp Singer. All authors of this article were involved in writing the paper and in the discussion of different ideas for the experimental design.

3.2 How to Apply Markov Chains for Modeling Sequential Edit Patterns in Collaborative Ontology-Engineering Projects

First of all, this article addresses the first research question by presenting a novel application of Markov chains to model sequential usage patterns in the change-logs of different collaborative ontology-engineering projects. Additionally, it provides a detailed description of the analysis process, highlighting all steps required to determine the most appropriate order of a Markov chain model for a given set of edit sequences. The process starts with the preprocessing of the data, continues with the fitting and determination of the appropriate Markov chain model and finishes with the interpretation of the obtained results. For determining the appropriate model order colleagues and I make use of likelihoods, information criteria (i.e., Akaike and Bayesian information criteria) as well as a cross-fold prediction validation.

The fitted models and results not only allow to identify patterns and regularities in the logs of changes but can also be used to predict future actions based on the modeled edit sequences. For this article, colleagues and I were specifically interested in determining the appropriate Markov chain orders—postulating on how many previous actions future ones depend on.

In particular, this article presents several sequential pattern analyses on a large-scale collaborative ontology-engineering dataset, which are intended to demonstrate the practical usefulness of the fitted Markov chains. For most of the presented analyses, models of order > 1 were determined to represent the best trade-off between model complexity and predictive accuracy.

Hence, this indicates that sequential edit patterns in collaborative ontology-engineering projects can be modeled using Markov chains. However, it also strengthens the warrant for further analyses to confirm and generalize the results presented in this article.

How to Apply Markov Chains for Modeling Sequential Edit Patterns in Collaborative Ontology-Engineering Projects

Simon Walk[a], Philipp Singer[c], Markus Strohmaier[c,d], Denis Helic[b], Natalya F. Noy[e], Mark A. Musen[e]

[a]*Institute for Information Systems and Computer Media, Graz University of Technology, Austria*
[b]*Knowledge Technologies Institute, Graz University of Technology, Austria*
[c]*GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany*
[d]*Dept. of Computer Science, University of Koblenz-Landau, Germany*
[e]*Stanford Center for Biomedical Informatics Research, Stanford University, USA*

Abstract

With the growing popularity of large-scale collaborative ontology-engineering projects, such as the creation of the 11^{th} revision of the International Classification of Diseases, we need new methods and insights to help project- and community-managers to cope with the constantly growing complexity of such projects. In this paper, we present a novel application of Markov chains to model sequential usage patterns that can be found in the change-logs of collaborative ontology-engineering projects. We provide a detailed presentation of the analysis process, describing all the required steps that are necessary to apply and determine the best fitting Markov chain model. Amongst others, the model and results allow us to identify structural properties and regularities as well as predict future actions based on usage sequences. We are specifically interested in determining the appropriate Markov chain orders which postulate on how many previous actions future ones depend on. To demonstrate the practical usefulness of the extracted Markov chains we conduct sequential pattern analyses on a large-scale collaborative ontology-engineering dataset, the International Classification of Diseases in its 11^{th} revision. To further expand on the usefulness of the presented analysis, we show that the collected sequential patterns provide potentially actionable information for user-interface designers, ontology-engineering tool developers and project-managers to monitor, coordinate and dynamically adapt to the natural development processes that occur when collaboratively engineering an ontology. We hope that presented work will spur a new line of ontology-development tools, evaluation-techniques and new insights, further taking the interactive nature of the collaborative ontology-engineering process into consideration.

Keywords: Markov chains, sequential patterns, usage patterns, collaborative ontology engineering,

1. Introduction

In recent years, we have seen significant increase in the use of structured data. In many cases, workers have used ontologies to integrate and interpret this data. As a result, we have seen an increase in the number of large-scale projects, focusing on collaboratively engineering ontologies. For example, the World Health Organization (WHO) is leading the collaborative online

Preprint submitted to Elsevier *December 14, 2015*

development of the new revision of the International Classification of Diseases (ICD), which represents an important classification scheme that is used in many countries around the world for health statistics, insurance billing, epidemiology, and so on. Wikidata[1], another collaborative ontology-engineering project initiated by the Wikimedia Foundation,[2] is gathering structured data in multiple languages to link to and between Wikipedia and its different language editions. To understand and support the new requirements that this collaborative approach introduces, researchers have analyzed and developed new ontology-engineering tools, such as Collaborative Protégé and WebProtégé [1, 2]. These tools not only provide a collaborative environment to engineer ontologies, but also include mechanisms that are targeted towards augmenting collaboration and increasing the overall quality of the resulting ontologies by supporting contributors in reaching consensus. For user-interface designers, community managers as well as project administrators, analyzing and understanding the ongoing processes of how ontologies are engineered collaboratively is crucial. When provided with detailed and quantifiable insights, the used ontology-engineering tools or even the development strategy can be automatically revised and adjusted accordingly. Engineering an ontology by itself already represents a complex task; this task becomes even more complex when adding a layer of social interactions on top of the development process. In the light of these challenges, we need new methods and techniques to better understand and measure the social dynamics and processes of collaborative ontology-engineering efforts.

In this work, we want to focus on sequences of actions that users perform when collaboratively engineering ontologies. For example, when the change of a property by a user is succeeded by another change of a property by that user, the two changes can be used to represent the sequence of properties that this specific user has been working on. Better understanding such sequential processes can help system designers to increase the quality of an ontology or contributor satisfaction, among other things. To come back to our previous example, if we better understand the process of how users sequentially edit properties of concepts, we can recommend to users the property that they potentially might want to edit next. Alternatively, we can steer users away from their typical behavior in order to cover niche parts of the ontology. We know from previous studies, that sequential patterns of human actions can usually be predicted quite well. For example, Song et al. [3] showed that human mobility patterns are predictable; they also hypothesize that all human activities contain certain regularities that can be detected. We explore whether these regularities might also apply to our ontology-editing sequences.

Consequently, our main goal in this paper is the presentation of methods and techniques for acquiring detailed insights into these ongoing (sequential) processes when users collaboratively engineer an ontology. Hence, we introduce a novel application of a methodology based on Markov chains. We base our elaboration of this method on previous work that has focused on studying human navigational paths through websites [4]. We focus not only on the structure of given paths (e.g., the identification of common sequences), but also on the detection of memory (e.g., on how many previous changed properties does the next property a user changes depend on). We lay our focus on determining the appropriate Markov chain orders which allows us to get insights into on how many previous actions users reason their future actions.

The **main objectives** of this paper are:

- The presentation of a novel application of Markov chains on the change logs of collaborative ontology-engineering projects to gather new insights into the processes that occur

---

[1] `http://www.wikidata.org`

[2] `http://wikimediafoundation.org`

when users collaboratively create an ontology.

- The demonstration of the utility of the presented and adapted Markov chain framework by applying it on a large scale collaborative ontology-engineering project.

Tackling these two objectives enables us to answer questions that are of practical relevance for the development of collaborative ontology-engineering tools, such as: Do users have to switch frequently between the user-interface sections when working on the ontology? Which concept is a user likely to change next, the one closer to or further away from the root concept of the ontology? Which change type is a user most likely to perform next? Do users move along the ontological hierarchy when changing content? Can we identify edit behaviors, such as *top-down* or *bottom-up* editing? Do users only reason their future actions on the current ones or do they depend on a series of preceding ones? However, other kinds of questions are conceivable and can be studied in straight-forward manner by researchers by focusing on the methodological aspects presented in this work.

Results: Our results indicate that the application of Markov chains on the change-logs of collaborative ontology-engineering projects provides new and potentially actionable insights into the processes that occur when users collaboratively create an ontology for project administrators and ontology-engineering tool developers.

Contributions: We provide (i) a detailed description of the process for applying Markov chains on the change-logs of collaborative ontology-engineering projects and (ii) an evaluation of the extracted Markov chain models by applying the methodology on the change-logs of ICD-11, representing a large-scale collaborative ontology-engineering project that exhibits Markov chains of varying orders. Our **high-level contribution** is the presentation of a novel approach that can be used to gather new insights into ongoing processes when collaboratively engineering an ontology by making use of Markov chains to model sequential usage sequences. Amongst others, this allows practitioners to identify structural properties and regularities as well as predict future actions based on usage sequences.

The remainder of the paper is structured as follows: In section 2 we provide a brief introduction into collaborative ontology-engineering. We then continue to review related work in section 3. In section 4, we briefly describe and characterize the history of ICD-11 as well as the dataset and the underlying change-log. We continue with the description of the process in section 5, describing all necessary steps to extract and interpret Markov chains for a given dataset. In section 6, we apply the previously described process to ICD-11, extracting Markov chains of different orders for two different types of analyses. In section 7, we discuss potential implications and conclude our work in section 8.

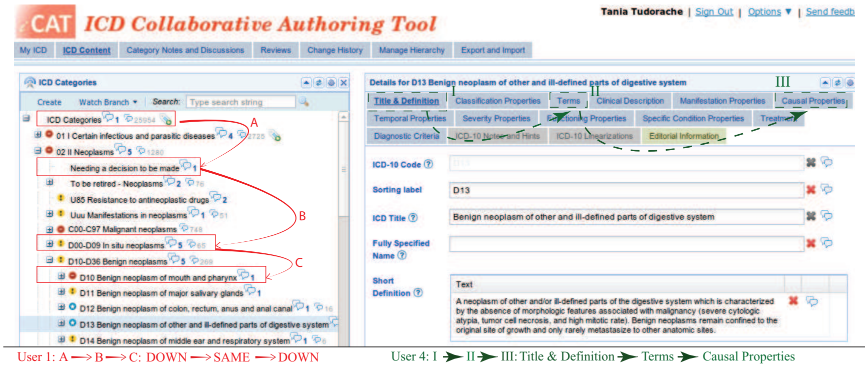
2. Collaborative Ontology Engineering

According to Gruber [5], Borst [6] and Studer et al. [7], an ontology is an explicit specification of a shared conceptualization. In particular, this definition refers to a machine-readable construct (the formalization) that represents an abstraction of the real world (the shared conceptualization), which is especially important in the field of computer science as it allows a computer (among other things) to “understand” relationships between entities and objects that are modeled in an ontology.

The field of collaborative ontology engineering and its environment pose a new field of research with many new problems, risks and challenges. In general, contributors of collaborative ontology-engineering projects, similar to other collaborative online production systems (e.g., Wikipedia), engage remotely (e.g., via the internet or a client–server architecture) in the development process to create and maintain an ontology. Given the complexity assigned to engineering an ontology, researchers and practitioners have already discussed and proposed different development methodologies. Analogously to the plethora of different software development processes and methodologies (i.e., the Waterfall-Model, agile development or SCRUM), methodologies and guidelines exist for (collaboratively) creating an ontology which define multiple different aspects of the engineering process. For example, the *Human-centered ontology engineering methodology* (HCOME) [8, 9, 10] represents such an approach that sets its focus on (continuously and) actively integrating the knowledge worker—the users who will rely on and use the created ontology—in the ontology life-cycle (i.e., by including the users in all planning stages, discussions, requirements analyses, etc.). Similarly, the *DILIGENT* process [11, 12, 13] defines principles for the distributed development of an ontology, including different stakeholders (e.g., developers or users of the ontology, who both have different purposes and needs for the resulting ontology). Debruyne et al. [14], Debruyne and Meersman [15], proposed the *Grounding Ontologies with Social Processes and Natural Language* (GOSPL) approach and tool in 2010. Again, a strong focus was put on the formalization of social processes, which directly result in and impact the evolution of the collaboratively engineered ontology.

3. Related Work

For the analysis and evaluation conducted in this paper, we identified relevant information and publications in the domains of (i) sequential pattern mining, (ii) Markov chain models and (iii) collaborative authoring systems. We discuss each domain next.



ICD-10 Code

Sorting label: D13

ICD Title: Benign neoplasm of other and ill-defined parts of digestive system

Fully Specified Name:

Short Definition: Text
A neoplasm of other and/or ill-defined parts of the digestive system which is characterized by the absence of morphologic features associated with malignancy (severe cytologic atypia, tumor cell necrosis, and high mitotic rate). Benign neoplasms remain confined to the original site of growth and only rarely metastasize to other anatomic sites.

Figure 1: **The iCAT User-Interface.** A screenshot of the iCAT interface, a custom tailored version of WebProtégé, developed for the collaborative engineering of ICD-11. The inline annotations represent exemplary transitions between states for two of our three analyses. The letters $A - C$ represent the sequential *Edit-Strategy Path* (see section 6.2) for **one** user, while the roman numbers $I - III$ constitute a representative sequential path for the *User-Interface Sections Path* analyses (see section 6.3) for another users. Note that for the *Edit-Strategy Paths*, every letter represents the transition between two consecutively changed concepts by the corresponding user. Analogously, for the *User-Interface Sections Paths* each number represents one section of the user-interface that was used by the corresponding users to contribute to the ontology.

3.1. Collaborative Authoring Systems

Research on collaborative authoring systems such as Wikipedia has in part focused on developing methods and studying factors that improve article quality or increase user participation. For example, Kittur et al. [16] have shown that for Wikipedia and del.ico.us, two collaborative online authoring systems, participation across users during the initial starting phase is unevenly distributed, resulting in few users (administrators) with a very high participation and contribution rate while the rest of the users (common users) exhibit little if any participation and contributions. However, over time, contributions shift from administrators towards an increasing number of common users, which at the same time still make little contributions individually. Thus, an analysis of the distribution of work across users and articles (as mentioned in Kittur and Kraut [17]) can provide meaningful insights into the dynamic aspects of the engineering process. This line of work is also related to research on problems that are common in these types of environments, such as the *free-riding* and *ramp-up* problems [18]. The free-riding problem characterizes the fact that users would rather tend to enjoy a resource than contribute to it. The *ramp-up* problem describes the issue of motivating users to contribute to a system when either content or activity (or both) in the overall system is very low. Researchers have proposed different types of solutions to these—sometimes called—knowledge-sharing dilemmas [18]. Wilkinson and Huberman [19] have shown that the quality of Wikipedia articles correlates with the number of changes performed on these articles by distinct users. More recent research which uses collaborative authoring systems, such as Wikipedia as a data source, focuses not only on describing and defining the act of collaboration amongst strangers and uncertain situations that contribute to a digital good [20] but also on antagonism and sabotage of said systems [21]. It has also been discovered that Wikipedia editors are slowly but steadily declining [22]. Therefore Halfaker et al. [23] have analyzed what impact reverts have on new editors of Wikipedia, showing that users have a much higher tendency to either stop working on Wikipedia articles after their contributions have been reverted or drastically decrease the amount of contributions.

Further, Viegas et al. [24] have shown that the history of an article and discussion pages in Wikipedia contain valuable information for administrators and moderators. In [25] the authors conclude that collectives in Wikipedia follow their self-imposed rules regarding well defined and formalized processes, such as featured articles. Schneider et al. [26, 27, 28, 29] discussed multiple different aspects and the importance of consensus finding on Wikipedia and the Social Semantic Web, by analyzing the history of articles in said systems, further strengthening the need for tools and analyses to be able to better understand and support digital collaborative endeavors.

3.2. Collaborative Ontology-Engineering Tools

A number of tools, such as the OntoWiki [30], the MoKi[31], Soboleo [32] or PoolParty [33] support collaborative ontology engineering, focusing on supporting and augmenting different aspects of collaborative development processes of ontologies. For example, Semantic MediaWikis [34] add semantic capabilities to traditional Wiki systems. They are intended to help users navigating the Wikis by introducing more meaningful semantic links and support of richer queries. Some of the Semantic Wikis available today focus on *enhancing content* with semantic links in order to allow more meaningful navigation and to support richer queries. Semantic Wikis usually associate a page to a particular instance in the ontology, and the semantic annotations are converted into properties of that instance. As an ontology represents a formalized and abstract version of a specific domain, disagreements between authors on certain subjects can occur. Similar to face-to-face meetings, these collaborative ontology-engineering projects need tools that augment collaboration and help contributors in reaching consensus especially when modeling (controversial) topics of the real world.

In fact, the majority of the literature about collaborative ontology engineering sets its focus on surveying, finding and defining requirements for the tools used in these projects [35, 36].

Protégé, and its extensions for collaborative development, such as WebProtégé and iCAT [2] (see Figure 1 for a screenshot of the iCAT ontology-editor interface) are prominent tools that are used by a large community worldwide to develop ontologies in a variety of different projects. Both WebProtégé and Collaborative Protégé provide a robust and scalable environment for collaboration and are used in several large-scale projects, including the development of ICD-11 [37].

Pöschko et al. [38], and Walk et al. [39] have created and further developed *PragmatiX*, a tool to browse an ontology and visualize aspects of its history. PragmatiX also provides quantitative insights into the creation process. The authors applied it to the analysis of the ICD-11 project.

3.3. Collaborative Ontology-Engineering Analyses

Strohmaier et al. [40] investigated the hidden social dynamics that take place in collaborative ontology-engineering projects from the biomedical domain and provided new metrics to quantify various aspects of the collaborative engineering processes. Falconer et al. [41] investigated the change-logs of collaborative ontology-engineering projects, showing that contributors exhibit specific roles, which can be used to group and classify these users, when contributing to the ontology. Pesquita and Couto [42] investigated if the location and specific structural features can be used to determine if and where the next change is going to occur in the Gene Ontology[3]. Wang et al. [43] have used association-rule mining to analyze user editing patterns in collaborative ontology-engineering projects. The approach presented in this paper uses Markov chains to extract much higher detailed user-interaction patterns incorporating a variable number of historic editing information.

Walk et al. [44] provided a detailed analysis of the commonalities and differences between five different collaborative ontology-engineering projects. Contrary to the presentation of the Markov chain framework in this paper, Walk et al. [44] concentrated their efforts on the interpretation of the differences and commonalities in first-order sequential patterns between five different collaborative ontology-engineering projects using aspects of the Markov chain framework presented in detail in this paper.

Debruyne and Nijs [45] presented a generic reputation framework to identify leaders in collaborative ontology-engineering projects. In their framework, they classified users as leaders according to a set of different characteristics (or reputation sensors), such as activity, engagement quality as well as features of the social interaction graph. In De Leenheer et al. [46], the authors suggested the use of social performance indicators to gather insights and broaden our understanding of the (ever changing) social arrangement collaboratively evolving an ontology.

Recently, Van Laere et al. [47] analyzed behavior-based user profiles in collaborative ontology-engineering projects, relying on GOSPL (Grounding Ontologies with Social Processes and Natural Language) and K-means clustering to group similar users. Di Francescomarino et al. [48] investigated multiple different features of wiki collaborative features for ontology authoring and showed their impact on the ontology lifecycle and the engineered ontology entities.

---

[3] `http://www.geneontology.org`

3.4. Sequential Pattern Mining

Agrawal and Srikant [49] first addressed the problem of sequential pattern mining in 1995. In their work the authors defined sequential pattern mining as: given a collection of chronologically ordered sequences, sequential pattern mining is about discovering all sequential (chronologically ordered) patterns, weighted according to the number of sequences that contain these patterns. They also introduced AprioriAll and AprioriScale, which also represent the first *a priori* sequential pattern mining algorithm. One year later, in 1996, Srikant and Agrawal [50] further included time-constraints and sliding windows to the definition of sequential patterns and introduced the widely popular and used generalized sequential pattern algorithm (GSP). With this work the authors showed that specific patterns cannot occur more frequently (above a threshold) if a subpattern of this pattern occurs less often (below that threshold). Many additional examples of a priori algorithms have been reviewed and discussed in literature [51, 52, 53], with SPADE [54] being one of the most prominently used and referred to algorithms. One major problem assigned to the a priori based sequential pattern mining algorithms was (in the worst case) the exponential number of candidate generation. As a priori based sequential pattern mining algorithms create (in the worst case) an exponential number of candidates Han et al. [55], Pei et al. [56] invented so called pattern-growth approaches. They circumvent the exponential candidate generation by strategically expanding found patterns and ignoring patterns that are not present in the data.

Today, many researchers have adapted different sequential pattern mining algorithms and approaches for different domains and use-cases. For example, Hsu et al. [57] analyzed algorithms for sequential pattern mining in the biomedical domain.

In this work we use Markov chain models (see next section) as opposed to sequential pattern mining techniques for our experiments as they also allow us to directly gain insights into memory effects in our sequential data at interest. Furthermore, we can simply vary the length of patterns that we want to detect by changing the order of the Markov chain model.

3.5. Markov chain models

Previously, Markov chain models have been heavily applied for modeling Web navigation—some sample applications of Markov chains can be found in [58, 59, 60, 61, 62, 63]. Detailed specifications of the parameters used in a Markov chain—e.g., transition probabilities or also the specification of model orders—have previously been used to capture specific assumptions about the real human navigational behavior. One frequently used assumption is that human navigation on the Web is memoryless. This is further postulated in the *Markovian assumption* which states that the next state in a system only depends on the current one and not on a sequence of preceding ones. This is, for example, also modeled in the Random Surfer model in Google's PageRank [64].

Previously, researchers have investigated whether human navigation really is memoryless in a series of studies (e.g., [65, 61]). However, they mostly have shown that the benefit of higher orders is not enough to compensate the extreme high number of parameters needed. Hence, the memoryless model seems to be a plausible abstraction (see e.g., [66, 67, 62, 63]). Recently, a study picked up on these investigations and again suggested that the Markovian assumption might be wrong for Web navigation patterns [68]. Based on these controversies regarding memory effects in human navigation, Singer et al. [4] presented a framework for determining the appropriate Markov chain order. Their studies on several navigational datasets revealed that the memoryless model indeed seems to be a plausible abstraction. However, their work also highlighted that on a topical level (by looking at paths over topics instead of pages) clear memory effects can be observed. In this work, we adapt the corresponding framework in order to apply it to the process of collaborative ontology engineering.

Using Markov chains we want to learn more about the ongoing processes when collaboratively engineering an ontology, thus the work presented in this paper partly builds upon this and related lines of research and tries to expand them towards collaborative ontology authoring systems.

4. Datasets

In this section, we present the main data studied in this paper. Mainly, we focus on the International Classification of Diseases (ICD-11) (Section 4.1). For deriving the change-logs, we utilize the Change and Annotation Ontology (ChAO) (Section 4.2).

4.1. International Classification of Diseases, 11th Revision

ICD-11[4], developed and maintained by the World Health Organization, is the international standard for diagnostic classification that is used to encode information relevant to epidemiology, health management, and clinical use. Health officials use ICD in all United Nations member countries to compile basic health statistics, to monitor health-related spending, and to inform policy makers. As a result, ICD is an essential resource for health care all over the world.

The development of ICD-11 represents a major change in the revision process. Previous versions were developed by relatively small groups of experts in face-to-face meetings. ICD-11 is being developed via a web-based process with many experts contributing to, improving, and reviewing the content online. It is also the first version to use OWL as its representation format.

We choose ICD-11 as an example ontology to demonstrate the effectiveness of the Markov chain methodology as the ontology satisfies several critical requirements for the applicability of our method: (i) at least two users have contributed to the project, and (ii) a structured log of changes (see section 4.2) without ambiguous references to the elements in the ontology is available. These characteristics can be seen as the minimum requirements to allow for an application of Markov chains onto collaborative ontology-engineering projects. For a list of characteristics for ICD-11 see Table 1.

4.2. The Change and Annotation Ontology (ChAO)

The ontology that we use for the demonstration of the Markov chain-based sequential usage pattern analysis, the International Classification of Diseases in its 11th revision, is created using

---

[4] `http://www.who.int/classifications/icd/ICDRevision/`

Table 1: **Characteristics of the International Classification of Diseases 11th revision (ICD-11)** that we used for the demonstration to extract sequential patterns in collaborative ontology-engineering projects. The number of users corresponds to the number of users that have contributed at least 1 change to ICD-11.

	ICD-11
concepts	48,771
changes	439,229
users	108
development tools	iCAT
first change	18.11.2009
last change	29.08.2013
log duration (ca.)	4 years

Figure 2: **The Analysis Process.** This figure depicts the different steps of the process that have to be performed to determine and evaluate the best fitting order of a Markov chain for a given dataset. The first two steps of the process involve a *Mapping* (section 5.1) of the change-log data onto the underlying ontology and *Session Separation* (section 5.2) tasks. The *State Selection* step (section 5.3) is split into two separate tasks. First, questions have to be formulated that are to be investigated relying on the presented Markov chain analysis. Second, features of changes, which correspond to the previously formulated questions, have then to be identified and selected. In the *Path Extraction* (section 5.4) step, all of the previously identified features of changes have to be extracted and chronologically sorted. Once the paths are extracted, they can be used as input for the *Model Fitting* (section 5.5), where the transition probabilities for the Markov chains are calculated. In the *Model Selection* step (section 5.6), we determine the best fitting order of a Markov chain according to over- and under-fitting of the underlying data. The last step of the process, *Interpretation* (section 5.7), is used to combine the results of the different approaches of the *Model Selection* to determine the best-fitting Markov chain order for the underlying data.

a custom tailored versions of WebProtégé called *iCAT*. The tool provides a web-based interface as well as change-logs, which can be directly mapped onto the ontology that is to be created. The mapping of the change-log entries and the ontology depends on the availability of unique IDs for entities, such as users and concepts. These unique IDs are internally (unambiguously) mapped to the IDs (or URIs) of the corresponding elements of the ontology, allowing us to track, extract and analyze changes of concepts even if, for example, their title and all of their attributes are changed or their values are ambiguous. This means that for every entry in the change log we have unique IDs that can be used to retrieve all involved entities. In traditional change-logs, which are usually separated from the productive environment, one minimalistic change could, for example, solely consist of one string, such as "*The title of concept 02 II Neoplasms was changed from Neoplasm to Neoplasms*". The change logs provide a direct mapping to the concept and user (among others) affected by the changes, avoiding ambiguity, even if multiple concepts exhibit the same property values (i.e., have the same title "Neoplasms"). Note that whenever we refer to the underlying ontology, we refer to ICD-11 and not ChAO or the change-logs.

Protégé and all of its derivatives use the **Change and Annotation Ontology (ChAO)** [69] to represent these changes. In contrast to traditional change-logs, ChAO itself represents a *structured* log of changes that allows for explicitly (semantically rich) defined classes, properties and relationships. This means that change types are represented as ontology classes in ChAO and changes in the domain ontology (e.g., ICD-11) are instances of these classes (Figure 3). Similarly, notes that users attach to concepts or threaded user discussions (represented as *Annotations* in Figure 3) are also stored in ChAO. Further, ChAO contains unique and unambiguous references to all entities in the ontology, for which ChAO is storing the changes and annotations.

ChAO records two types of changes, so-called "Atomic" and "Composite" changes. "Atomic" changes represent one single action within the ontology and they consist of several different types of changes such as *Superclass Added*, *Subclass Added* or *Property Value Changed*. "Composite" changes combine several atomic changes into one change action that usually corresponds to a single action by a user. For example, moving a concept inside the ontology is represented by one composite change that consists of—at least—four "atomic" changes for removing and adding parent and child relations for all involved concepts. Every change and annotation provides information about the user who performed it, the involved concept or concepts, a time stamp and a short description of the changed or annotated concepts/properties. Whenever we talk about changes we refer to the $439,229$ changes stored in the ChAO (see Table 1), which are

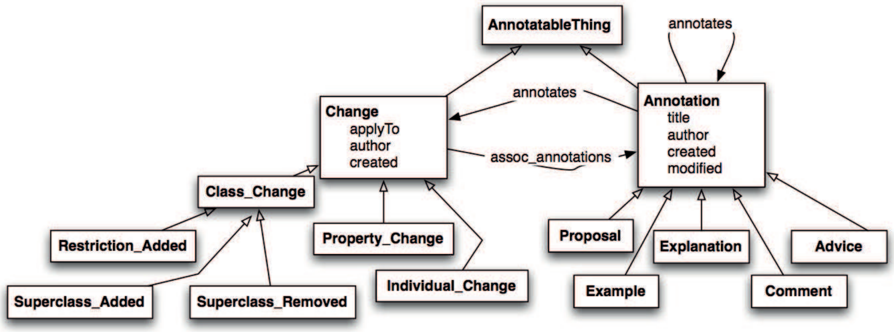


Figure 3: **The Change and Annotation Ontology.** The figure depicts a visual excerpt of the structure of the Change and Annotation Ontology (ChAO) used by Protégé [69]. Boxes represent classes and lines with arrows represent relationships (labeled) and subclasses.

always actual changes to the ontology (e.g., changes performed on ICD-11; opposed to proposed changes).

5. The Analysis Process

Figure 2 depicts an abstraction of all the steps necessary to better understand the process of how users sequentially edit properties of concepts in collaborative ontology-engineering projects. The first two steps of the analysis process, *Mapping* (section 5.1) and *Session Separation* (section 5.2), involve a mapping of the structured logs of changes onto the ontology as well as session separation tasks to prepare the data. In the *State Selection* step (section 5.3) research questions are formulated allowing for the corresponding features of changes to be identified and selected. In the *Path Extraction* (section 5.4) step, all of the previously identified features have to be extracted and chronologically sorted as they are needed as input for the Markov chain analysis.

For the *Path Extraction* step, we already have to know which questions we want to have answers for, as this determines the features of the changes that we are going to extract. Once the change data is mapped, extracted and converted into the required format, we can start the *Model Fitting* (section 5.5). In this step, we use the extracted and preprocessed data to calculate the transition probabilities for the different orders of the Markov chain models. To determine which Markov chain order provides the best trade-off between model complexity and predictive performance we conduct several *Model Selection* tasks (section 5.6). In the last step of the process, *Interpretation* (section 5.7), we combine the gathered information of the model selection tasks and provide insights on choosing the Markov chain order that statistically significantly best models the sequential data.

5.1. Step 1: Mapping

Given the structured nature of ChAO, it already provides the necessary internal IDs to map the referenced entities, which are involved in the corresponding stored change-actions, to the corresponding concepts, properties and users of the actual ontology (for more details see section 4.2). For example, if a specific property of a specific concept was changed, ChAO would provide us with the necessary IDs to unambiguously identify the changed concept and property. Hence, the mapping process for ICD-11 consists of simple id look-ups and joins between entries of ChAO and the actual ontology. For other datasets, individual mapping strategies have to be developed or derived, which allow for an unambiguous identification of all involved entities, such as users, concepts or properties.

5.2. Step 2: Session Separation

Ontologies of the size of ICD-11 cannot be developed in one single day, hence we decided to introduce what we call *artificial session breaks* to be able to gather more detailed information of the ongoing processes. As neither iCAT nor ChAO provide information about user sessions, we manually added these artificial session breaks, which allow us to identify (or at least approximate) concepts and properties that users will work on, after or shortly before they take a break from editing the ontology. These session break states are named *BREAK* throughout all of our analyses and are specifically used to uncover the states before and after a break occurs in the change-logs for all analyses that investigate user-based activities (opposed to concept-based activities, which are only analyzed in section 6.3).

Figure 4 depicts the total amount of timespans between the changes of each user for ICD-11. The y-axis depicts the percentage of all changes performed within the corresponding timespan on the x-axis. The x-axis depicts the different timespan intervals in minutes. The majority ($> 95\%$) of all changes in ICD-11 are performed within 5 minutes. Thus, if two changes of the same user are apart longer than 5 minutes, we have introduced an *artificial session break* represented as a *BREAK* state in all the conducted user-based analyses.

5.3. Step 3: State Selection

To be able to select the states for the Markov chain analysis we have to first define what kind of questions we seek answers for and then identify and extract the corresponding states. For example, if we are interested to know what kind of change a user is most likely to conduct next, the set of states to be extracted are all the different types of changes in the system. If we are interested in the relative movement of users, allowing us to predict if a user will move closer, further away or stay at the same distance to the root node, we have to extract the depth-levels of the changed concepts and compare the previous level with the current level to extract relative movement states (i.e., *UP*, *DOWN* and *SAME*; for more info see section 6.2).

It is important to understand that, using Markov chains, we are mainly interested in predicting which state to occur next for a given user or a given concept. Note that if we do not have enough information to extract a chronologically ordered sequence of states, Markov chains cannot be used.

5.4. Step 4: Path Extraction

To be able to analyze sequential usage patterns, we first have to extract sequential paths from the preprocessed structured logs of changes, which we can then use as input data for the Markov chains.

A path represents a chronologically ordered list of changes or features that can be associated with that change, which are performed either by a user or are performed on a concept (Figure 5). For example, when predicting the property that a user is most likely to work on next, we extract a chronologically ordered list of all changed properties for all users. We then store these lists in a

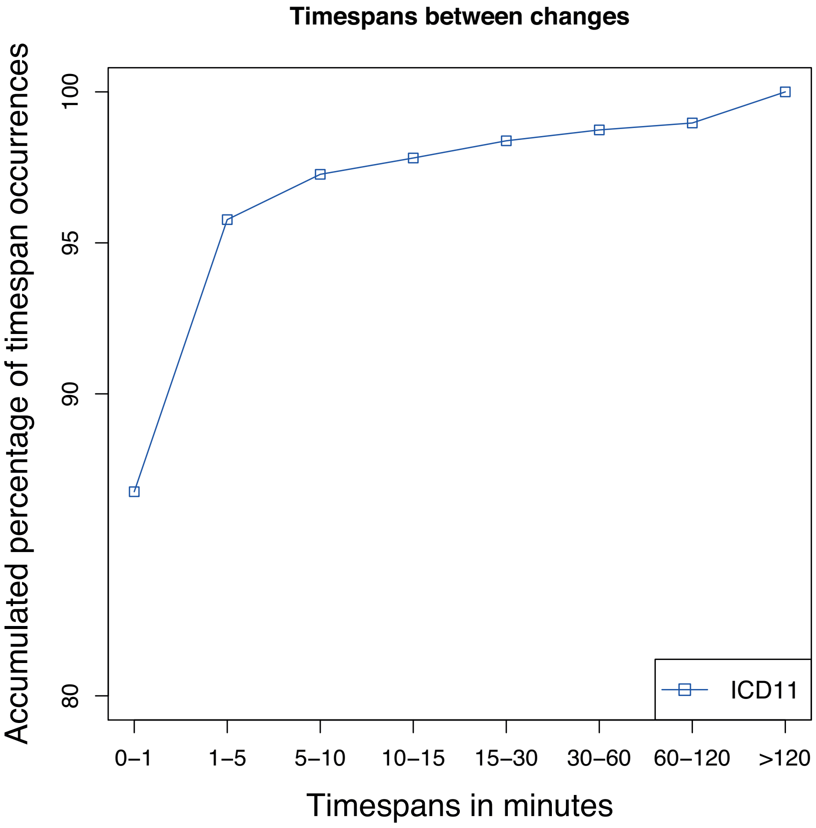


Figure 4: **Occurrence of different timespans.** This plot depicts the percentage of all changes that have been performed within a specific timespan for ICD-11. The x-axis lists the timespans in minutes and the y-axis lists the accumulated percentage of all timespans between two consecutively conducted changes for every user. To avoid the introduction of too many *artificial session breaks*, we decided to insert breaks for timespans between changes that are greater to the timespan so that $> 95\%$ of all changes do not introduce new sessions. In the case of ICD-11, this timespan is the $1 - 5$ minutes one, meaning that *BREAK*s have been introduced if the two changes in question are apart longer than 5 minutes.

file, where each user is represented by one line and the content of each line is the chronologically ordered list of changed properties of that user.

If we want to predict which property is most likely to be changed next for a given concept, we have to collect a chronologically ordered list of changed properties for each concept. Again, each line of the resulting file represents a concept while the content of each line is the chronologically ordered list of changed properties for that concept, *not* including *artificial session breaks* as this analysis is now concept-based.

For some of our analyses, we merged multiple consecutive changes of the same user on the same concept into two consecutive changes, resulting in *one self-loop*. For example, if one user would change the same property (e.g., title) on the same concept 5 times, we would merge these

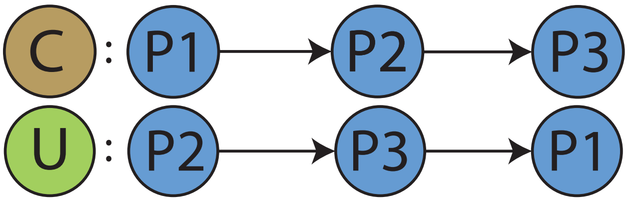


Figure 5: **Sequential Paths Sample.** The top row of the figure depicts an exemplary **concept-based** sequential property path ($P1$ to $P3$) for concept C. This means that for concept C the property $P1$ was changed first, then property $P2$ and most recently changed was property $P3$. The bottom row of the figure depicts the sequential property path ($P1$ to $P3$) for a user U (**user-based**). Analogously, user U has first changed $P2$, continued to change property $P3$ and most recently changed $P1$.

5 changes of the same property into two changes, resulting in one *self-loop* in the extracted path from title to title, opposed to four transitions from title to title. We performed this process of merging multiple consecutive changes into one single *self-loop* to minimize the detection of higher order Markov chains that are biased towards transitions between the same states from the same concepts. This is particularly useful as there is no, or only minimal, actionable information when predicting that a user is going to perform the same change on the same concept again. If an ontology would provide multilingual properties and we are specifically interested in potential change-sequence patterns between these multilingual property values, we would have to create additional states accordingly (e.g., *property_eng*, *property_ger*, etc.)

5.5. *Step 5: Model Fitting*

Markov chain models are well-known tools, among others, for modeling navigation on the web. We resort to and recapitulate the established methods first described by Singer et al. [4].

In general, a Markov chain consists of a finite *state-space* and the corresponding *transition probabilities* between these states. For our analysis, we will make use of the transition probabilities to identify likely transitions for a variety of different states. To be able to do so, it is important to understand the nature of Markov chains. Formally, a finite and discrete (in time and space) Markov chain can be seen as a stochastic process that contains a sequence of random variables–$X_1, X_2, \ldots, X_n$. One of the most well-known assumptions about Markov chains is the so-called *Markovian property* that postulates that the next state of a sequence depends only on the current state and not on a sequence of preceding ones. Such a first-order (also called memoryless) Markov chain holds if:

$$\begin{aligned} P(X_{n+1} = x_{n+1} | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) &= \\ P(X_{n+1} = x_{n+1} | X_n = x_n) & \end{aligned} \tag{1}$$

We assume *time-homogeneity* which means that the probability of a transition is independent of n. For all our Markov chains and for simplification we will refer to data (i.e., sequential paths) on which we fit a Markov chain model as a sequence $D = (x_1, x_2, ..., x_n)$ with states from a finite set S. Hence, we can rewrite the Markovian property as:

$$p(x_{n+1} | x_1, x_2, ..., x_n) = p(x_{n+1} | x_n) \tag{2}$$

Furthermore, as we are also interested in higher order Markov chains (i.e., the next state not only depends on the current one but on a series of preceding ones), we can state that in a k-th order Markov chain the next state depends on k previous ones. This leads to the following, more general equation:

$$p(x_{n+1}|x_1, x_2, ..., x_n) = p(x_{n+1}|x_n, x_{n-1}, ..., x_{n-k+1}) \tag{3}$$

Note that we can easily convert higher order Markov chains to first-order Markov chains by modeling all possible sequences of length k as states and adjusting the probabilities accordingly. Hence, we can focus on defining the methods for first-order chains solely, as this applies for higher ones as well.

A Markov chain model is usually represented via a stochastic transition matrix P with elements $p_{ij} = p(x_j|x_i)$ where it holds that for all i:

$$\sum_j p_{ij} = 1 \tag{4}$$

For easier understanding, one could think of a first-order Markov chain model as a matrix, where each column and row correspond to a state of the *state-space* and the elements within the matrix represent the transition probabilities to and from each state towards the corresponding other states. For higher order Markov chain models, the states would include the combinations of all states, which is drastically increasing the state-space and thus, the complexity of the Markov chain.

Furthermore, we also allow k to be zero, resulting in a so-called *zero-order* Markov chain model. This can be seen as a lower baseline and corresponds to a *weighted random selection* [4] – i.e., the probabilities are defined by the number of occurrences of states.

Maximum Likelihood Estimation (MLE): To be able to determine the transition probabilities p_{ij} between the states x_i and x_j, we apply Equation 5, where n_{ij} corresponds to the total number of transitions between states x_i and x_j:

$$p_{ij} = \frac{n_{ij}}{\sum_j n_{ij}} \tag{5}$$

Hence, the maximum likelihood estimate (MLE) for the transition probability p_{ij} simply is the number of times we observe a transition between state x_i to state x_j in our data D divided by the total number of outgoing transitions from state x_i to any other state.

5.6. Step 6: Model Selection

As our goal is to determine the most appropriate Markov chain order, we need to establish some methods for choosing the right one. Basically, we always want to compare a null model with an alternative model. To give an example, in our case the null-model could refer to a first-order Markov chain model while the alternative-model could refer to a second-order Markov chain model. Simply comparing likelihoods of two alternative models with each other is not enough though. Higher-order Markov chain models are always better fits to the data compared to lower-order ones by definition. This is reasoned by the higher complexity (higher number of parameters) of such higher-order Markov chain models. Thus, we need to balance the goodness of fit with the corresponding complexity when we want to compare models with each other.

To do so, we first focus on the *Akaike information criterion (AIC)* and *Bayesian information criterion (BIC)* to compare varying order Markov chain models with each other. In the following, we describe both methods, but we want to guide the reader to the work by Singer et al. [4] for a more thorough description.

Likelihood Ratio: To be able to calculate AIC and BIC, we have to calculate the likelihood ratio, which simply is the ratio of the maximum likelihoods of the alternative and the null model. The ratio gives us an indicator quantifying how much more likely the observed data is with the alternative model compared to the null model. As a result, we always compare lower order models with higher order models. In order to avoid underflow, we calculate the log likelihood ratio. We follow the notation by Tong [70] who defines the log likelihood ratio as ${}_k\eta_m$:

$$
{}_k\eta_m = -2(\mathcal{L}(\mathcal{P}(\mathcal{D}|\theta_k)) - \mathcal{L}(\mathcal{P}(\mathcal{D}|\theta_m))) \tag{6}
$$

$\mathcal{L}(\mathcal{P}(\mathcal{D}|\theta_k)$ represents the MLE for the null-model, while $\mathcal{L}(\mathcal{P}(\mathcal{D}|\theta_m)$ represents the MLE for the alternative model. Note that simply using this likelihood ratio as a proper indicator for choosing between two models is not enough due to the reasons outlined above. Hence, we resort to the AIC and BIC methods which we outline next.

Akaike information criterion (AIC): This information criterion can help us to determine the optimal model from a class of competing models – i.e., the appropriate Markov chain order. The final method is based on the minimization of the AIC – minimum AIC estimate also called MAICE – [71] and has been first used for Markov chains by Tong [70]. We define the AIC based on the work by Tong [70]:

$$
AIC(k) = {}_k\eta_m - 2(|S|^m - |S|^k)(|S| - 1) \tag{7}
$$

Basically, AIC subtracts the degrees of freedom from the likelihood ratio—thus, it penalizes models by their complexity. In our analysis, the degrees of freedom $(2(|S|^m - |S|^k)(|S| - 1))$ represent two times the difference between the number of parameters for the null-model (order k) and the alternative model (order m). The basic idea is to choose m as the maximum order we want to study and compare it with lower order models until the optimal Markov chain order is found. The most appropriate one is the one that exhibits the lowest AIC score.

Bayesian Information Criterion (BIC): This information criterion is very similar to the AIC except for the difference in penalization, as it increases negative weight on higher order models even more [72]:

$$
BIC(k) = {}_k\eta_m - (|S|^m - |S|^k)(|S| - 1)ln(n) \tag{8}
$$

We proceed similar as for AIC and choose m reasonably high. The specific penalty function is the degree of freedoms multiplied with the natural logarithm of the number of observations n [72], where an observation is always represented as a state in the change-logs.

Prediction Task: In addition to our information-theoretic methods for determining the appropriate Markov chain order, we use a cross validation prediction for this task. This prediction task is conducted to actually measure which model order is best suited for predicting the next state, with the available change-logs as input. The main idea behind this approach is to calculate the parameters on a training set and to validate the model on an independent test set. We apply Laplace smoothing in order to be able to predict states that are present only in the test set and not in the training set. To reduce variance, we perform a stratified 7-fold cross validation. In this case, we stratify the folds in order to keep the number of visited states in each fold equal.

The validation is based on the task of predicting the next step in a path of the test set. This validation also enables us to get detailed insights into the prediction possibilities of distinct Markov chain order models. Simply, one could predict the next state by taking the state with the highest probability in the transition matrix P. In the following, we describe the process of calculating the prediction accuracy.

First, we start by calculating the prediction accuracy for each fold separately. For doing so, we determine the average rank of each observation in a test set given the transition matrix as learned from the training data. In detail, given a current state x_n (or series of preceding states for higher order models), we look up the rank of the next state x_{n+1} in the sorted list of transition probabilities. Next, we average over the rank of all observations in the test set. We follow the notation of Singer et al. [4] and define the average rank $\overline{r(D_f)}$ of a fold D_f for some model M_k the following way[5]:

$$\overline{r(D_f)} = \frac{\sum_i \sum_j n_{ij} r_{ij}}{\sum_i \sum_j n_{ij}}, \tag{9}$$

where n_{ij} is the number of transition from state x_i to state x_j in D_f and r_{ij} denotes the rank of x_j in the i-th row of P. As frequently ties occur in these rankings, we assign the maximum rank to such ties (i.e., modified competition ranking). This method also includes a natural Occam's razor (penalty) for higher order models. After we have calculated the prediction accuracy of all folds, we average them and suggest the model with the lowest average rank.

In the last part of the *Model Selection*, we have to manually assess and combine the different results from the information criteria, the significance tests and the prediction task (see section 5.6), to determine the Markov chain order, which provides the best trade-off between model complexity (and thus, also computation time) and predictive power. Depending on the size of the change-log and the number of states that we want to investigate and predict, the different information criteria yield different suggestions for the best fitting Markov chain order, avoiding over- and under-fitting. The significance tests provide information about the highest Markov chain order, that is still significantly different to the remaining Markov chain orders.

In general, BIC exhibits a tendency to suggest lower Markov chain orders than AIC, due to the heavier weighted bias on model complexity. In contrast, the prediction task usually suggests the usage of higher order Markov chains. However, on closer investigation, the absolute differences between the suggested orders of AIC and BIC versus the suggested order of the prediction task, most of the time, do not justify the drastically increased model complexity (and thus computation time) of higher order Markov chains.

Overall, all presented methods try to achieve the same goal, i.e., balancing the goodness of fit with the number of parameters of varying Markov chain orders. Higher order Markov chain models have much higher complexity and thus, are potentially prone to overfitting. AIC and BIC achieve this in a natural manner by having direct complexity balance terms in their equations. For cross-validation, we try to include a natural Occams razor by our corresponding choice of how to rank ties. Thus, we believe that contrasting all presented methods in this article provides really thorough insights into the appropriate Markov chain order given the data.

However, as mentioned, the results of these methods (which frequently match anyhow) might be weighted differently according to the goal of the study. If the main goal is to study predictability, one might want to focus on cross-validation as it also directly provides a measure of how well we can predict with varying order models. However, the calculation of the cross-validation is quite expensive, which is why one want to focus on AIC and BIC. The focus of these two methods is to provide an answer to how well varying order models fit the data in relation to each other.

---

[5] alternatively, one could also use measures like perplexity

As mentioned, complexity is incorporated; BIC has a higher penalty for complexity compared to AIC. According to Singer et al. [4] , AIC might be best suited for prediction, while BIC might be better for explanation. This is also reasoned by the observation that AIC is asymptotically equivalent to cross validation if both use MLE. As a final note, we want to mention that BIC is asymptotically consistent. For further information of the advantages and disadvantages—as well as further methods for order estimation—please refer to the work by Vrieze [73] and Singer et al. [4].

Limitations: Note that the model-estimation methods described in this work balance the goodness of fit with the number of parameters needed for each Markov chain order model. This trade-off is necessary, as specifically higher order models need an exponentially growing number of parameters which might not be offset by the statistically significant benefit against lower order models and is also reflected by the initial choice about the set of states to consider. Thus, the results are also influenced by the amount of finite data available which is a common problem of statistical methods that mostly rely on asymptotic approximations. Basically, the more data we observe, the more amenable we are towards more complex models—i.e., higher order Markov chain models. Hence, if the underlying process actually accords to a higher order Markov chain process, we need a certain amount of data for a given complexity, to be able to properly detect this order. With insufficient data, lower orders might be identified as being appropriate as the goodness of fit cannot compensate the complexity. Hence, it is also necessary to have large change-logs available in order to have the opportunity to detect higher order Markov chain models.

The required total number of available observations, that is the number of performed changes, for detecting potential higher orders is directly related to the number of unique states that are extracted. For example, if all changes are mapped on two unique states (e.g., *structural changes* and *property changes*), smaller change-logs might already yield satisfying results, whereas higher numbers of unique states might require exponentially larger change-logs for the detection of higher orders.

In this work (see Table 1), we study a dataset with around $440,000$ changes and with a limited number of distinct states. Also, our results highlight several higher order models as being more plausible compared to lower order models. Thus, we can be confident that we have sufficient data to detect higher order Markov chain models as being appropriate, if they actually are. If a zero order Markov chain model would be suggested each time, we would need to rethink our data base.

5.7. Step 7: Interpretation

After determining the best fitting Markov chain order we can start interpreting the results. For example, when investigating the next, most likely change type to be performed by a user, we can look at the transition probabilities and given n previous changes, where n equals the order of the best fitting Markov chain model, infer a ranked list of most probable transitions.

6. Demonstration & Evaluation

In this section, we investigate the qualitative analysis that we can do using sequential pattern analyses. We present the types of questions that we can ask and provide the example analysis based on the change logs for the editing process of ICD-11 (Table 1).

In section 6.1, we investigate if and to what extent sequential patterns of performed change types can be detected.

Table 2: **Change Types.** The table depicts all types of changes that are used in the *Change-Type Analysis* in Section 6.1 The change types MOVE and CREATE represent the corresponding changes performed on the classes. Note that classes in ICD-11 are not deleted, but are moved to specific areas in the ontology, hence the omission of the DELETE type.. As the different properties in ICD-11 have been determined very early on in the development process and additional properties are very rarely introduced—which can only be done by administrators—we have neglected these types of changes and concentrated our analysis on the different edit actions that can be performed on properties.

Change Type	**Description**
MOVE	Changes that move a class.
CREATE	Changes that create a new class.
BOT	Changes that were automatically performed by bots.
OTHER	Any change that does not fit any other change type.
EDIT_REPLACE	Changes that replace the value of a property.
EDIT_REMOVE	Changes that remove the value of a property.
EDIT_IMPORT	Changes that import the value of a property.
EDIT_ADD	Changes that add a value to a property.

To see where and how users contribute to the ontology and if they exhibit sequential patterns when doing so, we analyze edit strategy patterns, such as *bottom-up* or *top-down* editing behavior (section 6.2).

In section 6.3, we report on our investigation on whether users have to switch frequently between different sections of the user-interface while contributing to ICD-11 and how often (and in which order) do they use the different sections of the user-interface in order to add information for a concept.

Step 1, Mapping and **Step 2, Model Separation** are the same for all types of analyses that we present in this section. We describe these steps in sections 5.1 and 5.2. In the remainder of this section we focus on the remaining steps, which differ from one type of paths to the other. **Step 7, Interpretation** is mainly focusing on the implications of the best fitting order of Markov chains, rather than an in-depth investigation of the transition probabilities. A detailed interpretation of the transition probabilities for the *Change-Type Paths* analysis, can be found in section 7.

6.1. Change-Type Paths

Step 3: State Selection. The analysis of change types provides information about the type of a change that a user will most likely conduct next. The information of what kind of change a user is most likely to perform next could be used by, for example, user-interface designers and ontology-engineering tool developers to automatically adapt the interface. Additionally, knowing if users primarily concentrate their efforts on the same change types or engage in multiple diverse actions while editing the content of the ontology can also be used by project administrators for curation purposes. Furthermore, when investigating the transition probabilities between the different change types, it is possible to identify certain pairs of changes that "usually" occur together, providing again information for automatic user-interface adaptions.

Step 4: Path Extraction. We aggregated the change types into more abstract change-classes to minimize the necessary state space for detecting Markov chains, which still provide useful information for curation and work-delegation purposes. Note that these change types only represent an abstracted fraction of all available change types in ChAO. In general, these change-type classes are *CREATE* and *MOVE*, which include all changes that have a corresponding effect on classes in the ontology. Note that classes in ICD-11 are not deleted, but are moved to specific areas in the ontology, hence the omission of the DELETE type. Furthermore, we have created the classes *EDIT_ADD*, *EDIT_IMPORT*, *EDIT_REMOVE* and *EDIT_REPLACE*, which are used when values of properties are either added, imported, removed or replaced. There are two special cases for ICD-11, namely *BOT* and *OTHER*. The first change-type is used for automatically performed changes while the latter is used to mark changes that are not included in the other listed change-type classes, such as addition of direct types or adding and removing sub- and superclasses (see Table 2 for a short description of all change types).

In general, all types of changes with the "*EDIT_*" prefix are changes performed on the properties of a class. As the different properties in ICD-11 have been determined very early on in the development process and new properties are rarely introduced—which can only be done by administrators—we have neglected these types of changes (i.e., are aggregated as part of *OTHER*) and concentrated our analysis on the different edit actions that can be performed on existing properties.

For creating the sequential paths, we first mapped all the changes of each user in our datasets to the different aggregated change-classes. In a second step we stored them as chronologically ordered lists for each user and each dataset individually. Multiple consecutive identical change types of the same user on the same concept were merged into one *self-loop*.

Step 5: Model Fitting & Step 6: Model Selection. We used the extracted paths to calculate the transition probabilities between the different change-type classes in the *Model Fitting* step. We then calculated AIC and BIC for the extracted Markov chain models of varying order (Figure 6) to identify the appropriate order that reflects to what extent contributors exhibit memory patterns when changing concepts.

AIC and BIC suggest the usage of a third- and second-order Markov chain respectively. The likelihood ratio tests strengthen this observation as a second-order Markov chain for ICD-11 is significantly different from a first-order Markov chain, thus suggesting the selection of a second-order Markov chain model for predicting the next change type.

To determine which order of a Markov chain contains the highest predictive power, we conducted a stratified cross-fold validation prediction task (see section 5.6 for a detailed explanation). As depicted in Figure 6, the stratified cross-fold validation encourages the usage of a third-order Markov chain for ICD-11.

The combined results of the model selection tasks indicate the best performance with the usage of a third-order Markov chain for ICD-11 for the task of predicting the change type a user is most likely to conduct next.

Step 7: Interpretation. A Markov chain of third order indicates that the last three change types a user has performed provide the best amount of information on the change type that is most likely to be performed next by that user. This information can (potentially) be used by programmers and designers of ontology development tools to automatically adjust parts of the interface according to the change-action a user is most likely to perform next. For example, if the next change will most likely involve deleting a concept the user-interface could already present and/or highlight specific parts that correspond to the anticipated action or display additional information, such as recently deleted concepts by the corresponding user. Note that these results are specific for ICD-11 and iCAT and might differ for other collaborative ontology-engineering projects.

6.2. Edit-Strategy Paths

Step 3: State Selection. The analysis of *Edit Strategy Paths* focuses on the investigation of relative movement along the ontological structure. Using the gathered data we can identify if

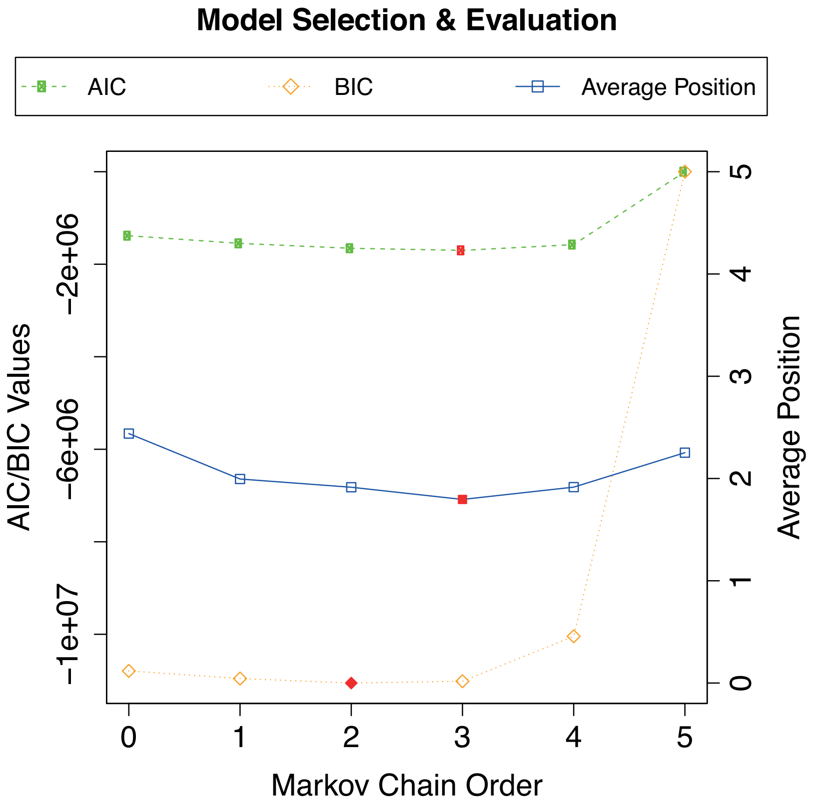


Figure 6: **Change Type Paths Model Selection and Evaluation.** This plot depicts the results of the AIC and BIC model selection criteria as well as the stratified cross-fold evaluation for the *Change Type Paths* analysis. The *x*-axis represents the different Markov chain orders. The left *y*-axis lists the AIC and BIC values of our model selection, while the right *y*-axis shows the average position values for the prediction task. The filled elements represent the corresponding Markov chain models, which achieved the best (lowest) average position score in the prediction task or lowest AIC and BIC values for the model selection. The information criteria, AIC and BIC, suggest the usage of a third- and second-order Markov chain respectively. The prediction task performed best relying on the predictive information extracted from a third-order Markov chain.

users who are contributing to the ontology are more likely to follow a *bottom-up* or *top-down* editing strategy. For example, if users would create or edit an ontology in a *bottom-up* manner, they would first model very specific concepts and continue to devote their work on more abstract concepts, while a *top-down* approach would work the opposite way. Note that this analysis can identify edit-strategy tendencies, however it could lead to wrong conclusions without manual verification of the change-logs. For example, if users generally tend to work on concepts in an alphabetical order, it is possible that this analysis could yield either, a *bottom-up*, a *top-down* or a non-apparent or random edit strategy, even though users do not purposely move along the semantic structure of the underlying ontology when contributing to the system. To make sure that our dataset does not exhibit such a behavior we have manually investigated the structured log of changes of ICD-11 to verify that the mentioned kind of contribution behavior is not present.

Furthermore, we were not able to recreate the exact class hierarchy of ICD-11 for every single change. This limitation is partly due to a lack of detail in the change-logs (e.g., some changes were conducted by the administrators of iCAT in the database, circumventing iCAT and ChAO. Hence, no change-logs are available of these actions, preventing a complete reconstruction of the ontology at every point in time). Thus, we decided to use the ontology *as is* at the latest point in time for our analysis. This basically means that if a class was changed by a user and afterwards moved, we would assume that this class has always been at the new location. To measure the extent of the potential bias, we counted all changes that were performed on a class before it was moved within in the ontology resulting in a total of 116,204 of 439,229 changes representing about 1/4 of all changes for ICD-11.

In particular, this analysis allows us to predict if the concept a user is going to contribute to next is on the same, a lower (more abstract) or a higher (more specialized) hierarchy level of the ontology. Using the gathered information we can infer if users follow a *top-down* or *bottom-up* edit strategy while contributing to ICD-11.

Step 4: Path Extraction. The states in this analysis indicate if a user, when contributing to the ontology, moved either closer to (state *UP*), further away (state *DOWN*) or kept the same distance (state *SAME*) from the root concept of the ontology.

We gathered the sequences for this analysis by calculating the shortest paths between all the concepts in the ontology and the root node, following *isA*[6] relationships. For ICD-11 the root category is *ICDCategory*, which is an equivalent of *owl:Thing*. Again, we merged multiple *self-loops*, represented by consecutive changes performed by the same user on the same concept, into one single transition. We have removed the data on users who contributed fewer than two changes from the analysis, as we require at least two changes to infer transitions between concepts.

A sample path is depicted in Figure 1. When following the annotations $A-C$, which represent the changes performed by one user, we can extract the following path: *DOWN, SAME, DOWN.* Note that for the creation of the first state we have to look at the first two classes that were changed by the corresponding user.

Step 5: Model Fitting & Step 6: Model Selection. We used the extracted paths to calculate the transition probabilities between the different change-type classes in the *Model Fitting* step. We then calculated AIC and BIC for the extracted Markov chain models (Figure 7) to identify the appropriate Markov chain order when modeling edit-strategy patterns of contributors changing concepts. For ICD-11 both AIC and BIC suggest a fourth- and third-order Markov chain respectively. Our likelihood ratio tests show that a third-order Markov chain for ICD-11 is still significantly different from a fifth-order Markov chain, indicating that either a third, fourth- or fifth-order Markov chain provides the best balance between model complexity and predictive power.

To determine the best-fitting Markov chain model orders to predict the next relative depth-level we conducted a stratified cross-fold validation prediction task (see Figure 7). The results of our prediction experiment suggest the usage of a fourth-order Markov chain for ICD-11.

As the differences between the higher-order Markov chains and the third-order Markov chain are very small, yet different, we agree with BIC and the significance test on the usage of a

---

[6] For our analysis we only consider *isA* relationships with regards to the *rdfs:subClassOf* property. In particular, classes connected via (directed) *isA* relationships specify that all the instances of one class (source) are also instances of the other class (target).

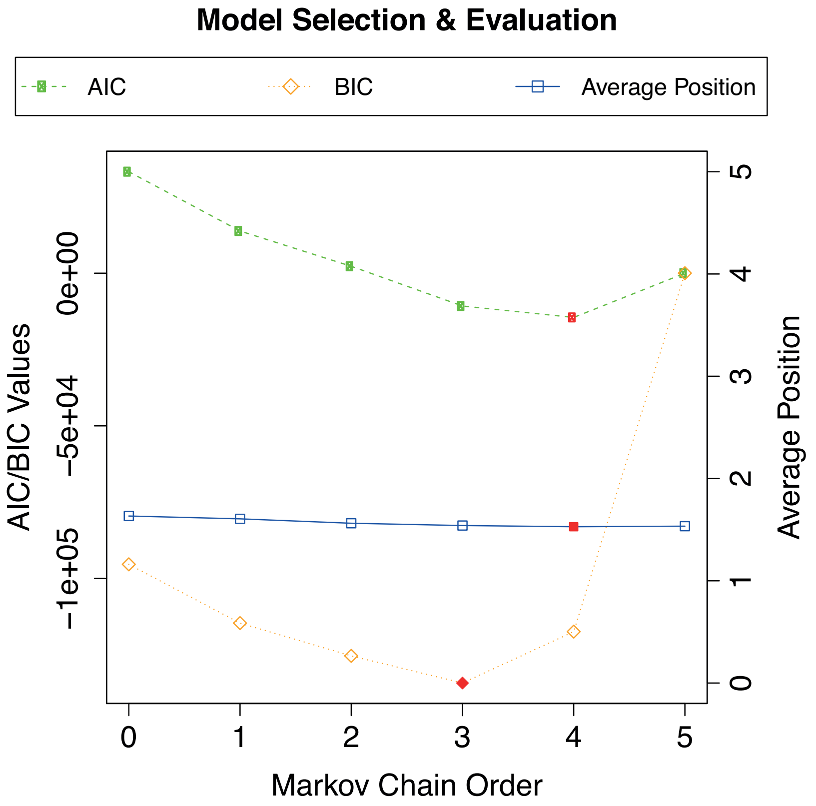


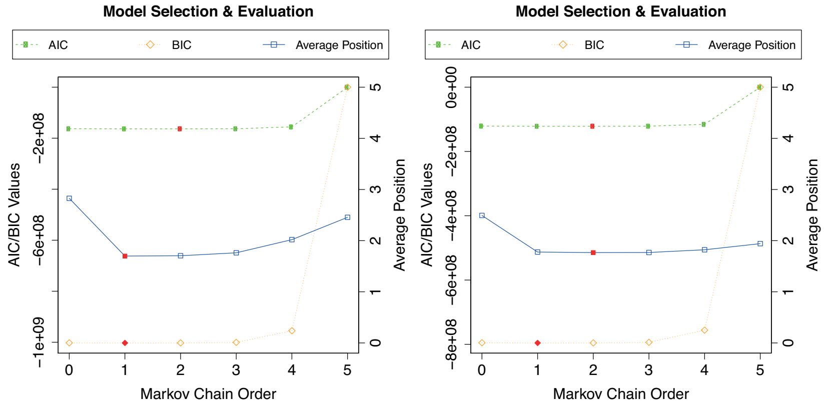
Figure 7: **Edit Strategy Paths Model Selection and Evaluation.** This plot depicts the results of the AIC and BIC model selection criteria as well as the stratified cross-fold evaluation for the *Edit-Strategy Paths* analysis. The x-axis represents the different Markov chain orders. The left y-axis lists the AIC and BIC values of our model selection, while the right y-axis shows the average position values for the prediction task. The filled elements represent the corresponding Markov chain models, which achieved the best (lowest) average position score in the prediction task or lowest AIC and BIC values for the model selection. The information criteria, AIC and BIC, were able to detect a fourth- and third-order Markov chain respectively. The prediction task yielded the best results with a fourth-order Markov chain model.

third-order Markov model for predictive tasks, due to the high increase in complexity of the higher-order models.

Step 7: Interpretation. A Markov chain of first order indicates that the last relative depth-level of a change performed by a user provides better information on where the user is going to change a concept next (as relative depth-level) than randomly selecting either *UP*, *DOWN* or *SAME*. After inspecting the resulting transition probabilities between the different states, we can conclude that users in ICD-11 exhibit a top-down edit strategy. Particularly, they are likelier to stay on the same or switch to a lower level of the ontology than they are, changing a class on a higher level of the ontology. In particular, this information could be exploited by project administrators to adjust milestones (i.e., first completing branches of the ontology, rather than adding properties to all concepts of the ontology). Note that these results are specific for ICD-11 and iCAT and might differ for other collaborative ontology-engineering projects.

6.3. User-Interface Sections Paths

Step 3: State Selection. The goal of this analysis is to investigate if we can map changes that occur in the ontology to actual areas and sections of the user-interface of iCAT, the collaborative ontology-engineering tool used to develop ICD-11. The user-interface of iCAT is divided into several sections, thematically grouping properties of concepts. For example, as depicted in Figure 1, the user-interface section *Title & Definition* groups the properties *ICD-10 Code*, *Sorting label*, *ICD Title*, *Fully Specified Name* and *Short Definition*. Other user-interface sections, grouping different properties, are for example, *Classification Properties*, *Terms* or *Clinical Description*. We investigate two different approaches: First, the user-based approach, where we analyze the sections of the user-interface used by contributors when editing the ontology. Second, the concept-based approach, where we investigate which sections of the user-interface are used when concepts are populated with data. If patterns can be detected, ontology-engineering tool developers can use this information to minimize the necessary effort for users to be able to contribute. It is important to note that not all properties and sections of iCAT are already actively used as ICD-11 is still under active development. Hence, the results of the presented analysis are limited by the properties and sections that are already available and actively used in iCAT. Rather than focusing on the results, this specific analysis was selected to demonstrate the feasibility and potential of the Markov chain analysis.



(a) User-based approach (b) Concept-based approach

Figure 8: **User-Interface Sections Path Model Selection and Evaluation.** This plot depicts the results of the AIC and BIC model selection criteria as well as the stratified cross-fold evaluation prediction task for the user- and concept-based approaches of the *User-Interface Sections Paths* analyses. The x-axes represent the different Markov chain orders. The left y-axes list the AIC and BIC values of our model selection, while the right y-axes show the average position values for the prediction task. The filled elements represent the corresponding Markov chain models, which achieved the best (lowest) average position score in the prediction task or best (lowest) AIC and BIC values for the model selection. For both approaches, AIC and BIC suggest a second- and first-order Markov chain respectively, while the prediction task produced the best average position with a Markov chain of first-order for the user-based and second-order for the concept-based approach.

Step 4: Path Extraction. The states for this analysis are represented by the different user-interface sections of iCAT. An excerpt of all different user-interface sections of iCAT can be seen in Figure 1.

To be able to analyze sequential patterns of different user-interface sections we extracted the chronologically ordered list of changed properties for (i) each user and (ii) each concept. We then continued by mapping the extracted properties to sections in the user-interface of iCAT. Whenever a change did not affect a property (e.g., because the change-action dealt with moving or creating a concept) and thus did not affect a user-interface section, the *no property* state was used. Analogously to the previous analyses, we merged consecutive changes of the same user on the same concept on the same property into one *self-loop* for the user-based analysis. For the concept-based analysis consecutive changes on the same concept and property have been merged into one *self-loop*.

A sample path is depicted in Figure 1. When following the annotations $I - III$, which represent consecutive changes performed by one user, using the highlighted sections of the user-interface, the following path can be extracted: *Title & Definition, Terms, Causal Properties*.

Step 5: Model Fitting & Step 6: Model Selection. We calculated AIC and BIC for the extracted Markov chain models (see Figures 8(a) and 8(b)) to determine the appropriate Markov chain order when modeling how users switch between sections of the interface when contributing to the ontology. For both approaches AIC and BIC suggest a second- and first-order Markov chain respectively. The conducted significant tests show that a second-order Markov chain for both approaches is significantly different from a first-order Markov chain, indicating that either a second-order or first-order Markov chain provide the best balance between model complexity and predictive power.

To determine the predictive power of the investigated Markov chain models of varying orders for predicting the section most probably used to edit a property next, a stratified cross-fold validation prediction task (see Figure 8) was conducted. For the user-based approach a first-order and the concept-based approach a second-order Markov chain yielded the best predictions.

Due to the fact that the determined second-order Markov chain performed nearly as well as a first-order Markov chain, it is best to use a first-order Markov chain to predict the next user-interface section, that a user is going to use, as it provides the best balance between model complexity (and thus computation time) and predictive power.

Step 7: Interpretation. A first-order Markov chain indicates that the last user-interface section, used to conduct a change by a specific user, contains information about the user-interface section that this specific user is most likely to use for the next change. If we would observe high transition probabilities between a fraction and frequently used sections of the user-interface, this could indicate that users have to visit many different sections while following their normal work-flow. If our inherent goal was to increase activity and contributions, a first potential approach could involve the restructuring of the user-interface to better accommodate this inherent edit-process by reducing or even minimizing the required clicks (and hence time) to contribute. Note that the proposed applications and implications of our analyses are of theoretic nature, to highlight the potential of the Markov chain analysis process. For future work we plan on further analyzing, validating and evaluating the recommendations and predictions generated via our Markov chain analysis in live-lab studies for multiple different ontology-development tools.

7. Discussion

In section 6 we have shown that the presented and adapted Markov chain model selection framework can be used to extract sequential patterns in the form of first and higher order Markov chains.

As shown in Table 3, Markov chains of third or higher order yield the best results in our prediction tasks. The information criteria AIC and BIC, putting a negative bias on model complexity, tend to suggest minimally lower Markov chain orders. After manually inspecting and comparing the performance of the different Markov chain models, the conducted significance tests and the model complexity, we identified that a third-order Markov chain provided the best balance between said attributes for the *Change-Type Paths* analysis and the *Edit-Strategy Paths* analysis. For both approaches of the *User-Interface Sections Paths* analyses a first-order Markov chain constitutes the best tradeoff between model complexity and performance. The identification of at least one higher-order Markov chain in our *Model Selection* tasks indicates that the Markovian assumption is *not* universally true for all features of the collaborative ontology-engineering change-logs. However, even if models of higher order are identified and, theoretically, provide better results than models of lower order, for the majority of the investigated change-log features a first-order Markov chain still represents the best tradeoff between model complexity and predictive power.

This result means that the previous three changes of a user contain predictive information about the change action that is most likely conducted next by that user in ICD-11. Analogously, the last change conducted by a user contains predictive information about the section of the user-interface that this user is most likely to use for the next change and if the user will stay on the same depth-level or moves up or down.

To expand further on the usefulness of Markov chains for analyzing change-logs of collaborative ontology-engineering projects we will provide an exemplary investigation of the structure of the extracted Markov chain model for the *User-Interface Paths (user-based)* analysis, including information about potential use-cases in productive environments.

Markov chain structure of the User-Interface Paths (user-based) Analysis:

Figure 9 depicts the transition probabilities of a first-order Markov chain for the user-interface section sequences for properties changed by users in ICD-11. The figure clearly shows that the sections of the user-interface frequently receive consecutive changes with minimal transition probabilities to different sections of the user-interface. Note that we removed all rarely used sections from Figure 9 as they do not contain valuable information, however, their removal drastically increases the readability and ease of interpretability of Figure 9.

iCAT provides a special export functionality, which allows users to export parts of the ontol-

Table 3: **This Table depicts a summary of all gathered results for ICD-11 and the performed analyses** of section 5. The numbers in this table represent the calculated and suggested Markov chain orders from our model selection (AIC and BIC), significance tests (Significant Diff.) and evaluation tasks (Prediction Task). *Best Balance* indicates the manually selected best-fitting order of a Markov chain, which represents the best trade-off between complexity of the Markov chain (and thus calculations) and the average position in our evaluation task.

	Markov chain orders for				
	AIC	BIC	Significant Diff.	Prediction Task	Best Balance
Change-Type Paths (cf. section 6.1)	3	2	${}_1\eta_3$	3	3
Edit-Strategy Paths (cf. section 6.2)	4	3	${}_3\eta_5$	4	3
User-Interface Sections Paths (User) (cf. section 6.3)	2	1	${}_1\eta_2$	1	1
User-Interface Sections Paths (Concept) (cf. section 6.3)	2	1	${}_1\eta_2$	2	1

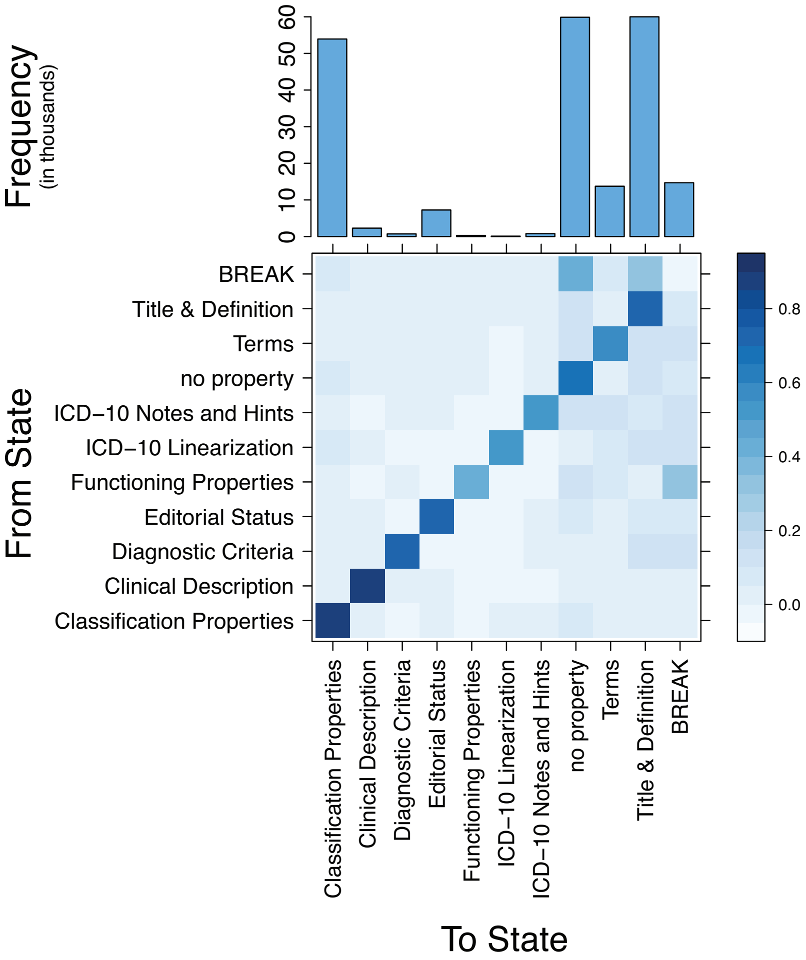


Figure 9: **Results for the *User-Interface Sections Paths* (user-based) analysis.** The states for these analyses are represented by the different sections of the user-interfaces of the ontology-engineering tool iCAT (see Figure 1). The transition probabilities for the first-order Markov chains are depicted in the transition map. Columns and rows represent the states, where rows are source states and columns are target states, indicating that a sequence always is read *from row to column*. Darker colors represent higher transition probabilities while lighter colors indicate lesser transition probabilities. A clear trend towards *self-loops* can be observed, meaning that changes are performed consecutively within the same sections of the user interface. The histogram depicts the absolute number of occurrences for each section for ICD-11 in alphabetical order. Sections with very low numbers of observations have been removed from the plots for reasons of readability.

ogy into a spreadsheet for quick local editing. However, no such automatic import functionality is present. To insert the edited values into the ontology, contributors have to manually add the changed properties in iCAT. This is usually done by selecting one property, changing its value and then cycling through all changed concepts where that property stays selected in the interface, allowing for easy and fast editing sessions.

The majority of changes were concentrated on a few selected sections—*Title & Definition*, *Classification Properties* and *Terms*—as depicted in the histogram of Figure 9.

Contributors to ICD-11 also exhibit a very high tendency either to change *no property* or a property of the *Title & Definition* section when resuming work after a *BREAK*. The state *no property* refers to all changes that do not affect the value of a property (e.g., moving a concept). Hence, these changes cannot be directly mapped to properties and sections of the user-interface. Further, the high number of *no property* changes warrants further inspection in future work. In this paper, we have concentrated our analysis on properties, which can be mapped to specific parts of the user interface and provide potential actionable information for ontology-tool developers.

Interpretation & Practical Implications: When looking at the results of this analysis, we can see that the functionality of the ontology-development tool might be a deciding factor on how users interact with the ontology when contributing. This is especially evident when considering the very high *self-loop* count for ICD-11, which is most likely supported and emphasized by the export functionality present in iCAT, which allows users to export parts of the ontology into a spreadsheet, which later-on has to be manually re-inserted. Conveniently, when switching concepts, the previously selected/edited property remains selected/active in iCAT, allowing for quick edit-workflows when inserting data for the same property (and thus same section) from external resources for multiple concepts.

Furthermore, it is of no surprise that users exhibit a very high probability to consecutively change properties in the *Title & Definition* section, given that it (i) contains the most basic properties with the highest priority to be added/completed and (ii) is the default section that is displayed once a user logs into the system.

The information collected with this analysis is of potential interest for project administrators, as they can adapt the engineering process to the needs of either the community or the project itself. For example, if active collaboration for different parts of the ontology is of utmost importance, the export functionality could be restricted, only allowing an export for certain parts of the ontology. Ontology-editor developers can use the transition probabilities between different sections of the user-interface to adapt, maybe even dynamically adapt the interface towards the inherent contribution processes of the community creating the ontology in question. In particular, by further expanding the User-Interface analysis we could potentially use the results to create adaptive user interfaces that reflect and augment the personal edit-styles of contributors. For example, parts of the interface could automatically adapt towards the processes of the users, relying on the transition probabilities of the extracted Markov chains, to allow for an easy transition between the current and the next, most probable, user-interface section used by a contributor. Different types of sequential paths can be used for a variety of applications. For example, we could use the chronologically ordered list of users conducting changes per class to predict which user is most likely going to change a specific class next.

8. Summary & Conclusions

The detailed description of the process for applying Markov chains on the change-logs of collaborative ontology-engineering projects represents a first step towards a broader methodology to gather new insights into the ongoing processes when collaboratively engineering an ontology.

The main contributions of this paper are as follows: (i) We provide the description of the process for applying Markov chains of varying order on collaborative ontology-engineering projects to extract and analyze sequential patterns. (ii) We categorize the types of qualitative analyses of collaborative ontology-engineering processes that Markov chains enable us to perform. (iii) Finally, we demonstrate the usefulness of such analyses on collaborative ontology-engineering change-logs using ICD-11.

We have made the Markov chain framework publicly available[7], hence the only requirement for replicating the analysis for other datasets is a structured change-log of the required granularity of detail (depending on the desired analyses). Results of the same analyses may differ for different datasets, depending on a multitude of factors. For example, the used ontology editor potentially influences the way users edit the ontology (i.e., changes the edit strategy).

In the conducted prediction experiment, several Markov chains of orders ≥ 1 have been retrieved, indicating that the Markovian assumption does not hold for all aspects of the development processes in collaborative ontology-engineering projects. To further expand on the usefulness of Markov chains, we have provided an example investigation of the structure of a first-order Markov chain and its implications and use-cases for productive environments. Note that for some of our analyses we assume the administrators and contributors to have full control over the used tools (e.g., can freely adapt, change and extend parts of the User-Interface). We are aware that this might not be the case for all collaborative ontology-engineering projects. However, we argue that the presented analyses can still provide valuable and actionable information, without having to directly edit the used tools. For example, by closely inspecting change-types and changed properties. Further, it is possible that due to restrictions in the ontology-engineering tool, users might not be able to deviate from certain paths. Hence, it is important to manually investigate and interpret the obtained patterns and avoid imposing “one specific way“of how to use the ontology-editing tool on users.

For future work we plan on using the presented Markov chain analysis process to study sequential action patterns in collaborative ontology-engineering projects. As a first step, we plan on acquiring the complete change-logs for multiple (> 100) projects created with WebProtégé and MoKi[8], to analyze commonalities and differences over different collaborative ontology-engineering editors.

Further, we plan on applying the presented Markov chain analysis on these datasets to identify and investigate known and established ontology-engineering methods (e.g., HCOME, GOSPL or NeOn) and best practices “in the wild”.

As change-tracking and even click-tracking data will become available more broadly, we believe that the mapped analysis process, presented in this paper, and the potential benefits of applying Markov chains on change-logs of collaborative ontology-engineering projects, represent an important step towards even better (and simpler) ontology editors. Using sequential edit information it is possible to dynamically anticipate the editing-style of the community. Even project administrators can augment the results of the analysis, for example by allowing for easier delegation of work to the most qualified users.

We hope that the presented approach will help project administrators, ontology-engineering tool developers and, most important, the community which is developing an ontology collaboratively, to devise new approaches, tools, mechanisms or even full methodologies to increase the quality of the resulting ontology and make contributing to the projects as easy as possible.

---

[7] `https://github.com/psinger/PathTools`

[8] `https://moki.fbk.eu/website/index.php`

Acknowledgements

This work was generously funded by a Marshall Plan Scholarship with support from Graz University of Technology. Additionally, this work was partially funded by the DFG in the research project "PoSTs II".

References

[1] T. Tudorache, N. F. Noy, S. Tu, M. A. Musen, Supporting Collaborative Ontology Development in Protégé, in: Proceedings of the 7th International Semantic Web Conference 2008 (ISWC 2008), volume 5318, Springer, Karlsruhe, Germany, 2008, pp. 17–32.

[2] T. Tudorache, C. Nyulas, N. F. Noy, M. A. Musen, WebProtégé: A Distributed Ontology Editor and Knowledge Acquisition Tool for the Web, Semantic Web Journal 11-165 (2011).

[3] C. Song, Z. Qu, N. Blumm, A.-L. Barabsi, Limits of Predictability in Human Mobility, Science 327 (2010) 1018–1021. doi:`10.1126/science.1177170`.

[4] P. Singer, D. Helic, B. Taraghi, M. Strohmaier, Detecting Memory and Structure in Human Navigation Patterns Using Markov Chain Models of Varying Order, PloS one 9 (2014) e102070.

[5] T. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220.

[6] W. Borst, Construction of engineering ontologies for knowledge sharing and reuse (1997).

[7] R. Studer, V. R. Benjamins, D. Fensel, Knowledge engineering: Principles and methods, volume 25, 1998, pp. 161–197.

[8] K. Kotis, G. A. Vouros, J. P. Alonso, HCOME: A tool-supported methodology for engineering living ontologies, in: Semantic Web and Databases, Springer, 2005, pp. 155–166.

[9] K. Kotis, G. A. Vouros, Human-centered ontology engineering: The HCOME methodology, Knowledge and Information Systems 10 (2006) 109–131.

[10] K. Kotis, On supporting HCOME-3O ontology argumentation using semantic wiki technology, in: On the move to meaningful internet systems: OTM 2008 Workshops, Springer, 2008, pp. 193–199.

[11] S. Pinto, S. Staab, Y. Sure, C. Tempich, OntoEdit empowering SWAP: a case study in supporting DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies (DILIGENT), in: The semantic web: research and applications, Springer, 2004, pp. 16–30.

[12] C. Tempich, H. S. Pinto, Y. Sure, S. Staab, An argumentation ontology for distributed, loosely-controlled and evolving engineering processes of ontologies (DILIGENT), in: The Semantic Web: Research and Applications, Springer, 2005, pp. 241–256.

[13] H. S. Pinto, C. Tempich, S. Staab, Ontology engineering and evolution in a distributed world using DILIGENT, in: Handbook on ontologies, Springer, 2009, pp. 153–176.

[14] C. Debruyne, Q. Reul, R. Meersman, GOSPL: Grounding Ontologies with Social Processes and Natural Language., in: ITNG, 2010, pp. 1255–1256.

[15] C. Debruyne, R. Meersman, GOSPL: A method and tool for fact-oriented hybrid ontology engineering, in: Advances in Databases and Information Systems, Springer, 2012, pp. 153–166.

[16] A. Kittur, E. Chi, B. Pendleton, B. Suh, T. Mytkowicz, Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie, World Wide Web 1 (2007) 1–9.

[17] A. Kittur, R. E. Kraut, Harnessing the wisdom of crowds in wikipedia: quality through coordination, in: Proceedings of the 2008 ACM conference on Computer supported cooperative work, CSCW '08, ACM, New York, NY, USA, 2008, pp. 37–46.

[18] A. Cabrera, E. F. Cabrera, Knowledge-Sharing Dilemmas, Organization Studies 23 (2002) 687–710.

[19] D. M. Wilkinson, B. A. Huberman, Cooperation and quality in wikipedia, in: Proceedings of the 2007 international symposium on Wikis, WikiSym '07, ACM, New York, NY, USA, 2007, pp. 157–164.

[20] B. Keegan, D. Gergle, N. S. Contractor, Hot off the wiki: dynamics, practices, and structures in Wikipedia's coverage of the Tohoku catastrophes., in: F. Ortega, A. Forte (Eds.), Int. Sym. Wikis, ACM, 2011, pp. 105–113.

[21] N. Shachaf, Beyond vandalism: Wikipedia trolls., Journal of Information Science; Jun2010, Vol. 36 Issue 3, p357-370, 14p, 2 Charts (2010).

[22] B. Suh, G. Convertino, E. H. Chi, P. Pirolli, The singularity is not near: slowing growth of Wikipedia, in: WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration, ACM, New York, NY, USA, 2009, pp. 1–10.

[23] A. Halfaker, A. Kittur, J. Riedl, Don't bite the newbies: how reverts affect the quantity and quality of Wikipedia work., in: F. Ortega, A. Forte (Eds.), Int. Sym. Wikis, ACM, 2011, pp. 163–172.

[24] F. B. Viegas, M. Wattenberg, J. Kriss, F. Van Ham, Talk before you type: Coordination in Wikipedia, in: System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, IEEE, 2007, pp. 78–78.

[25] F. B. Viégas, M. Wattenberg, M. M. McKeon, The hidden order of Wikipedia, in: Online communities and social computing, Springer, 2007, pp. 445–454.

[26] J. Schneider, T. Groza, A. Passant, A review of argumentation for the Social Semantic Web, Semantic Web 4 (2013) 159–218. URL: `http://dx.doi.org/10.3233/SW-2012-0073`. doi:`10.3233/SW-2012-0073`.

[27] J. Schneider, K. Samp, A. Passant, S. Decker, Arguments about deletion: how experience improves the acceptability of arguments in ad-hoc online task groups, in: Computer Supported Cooperative Work, CSCW 2013, San Antonio, TX, USA, February 23-27, 2013, 2013, pp. 1069–1080. URL: `http://doi.acm.org/10.1145/2441776.2441897`. doi:`10.1145/2441776.2441897`.

[28] J. Schneider, A. Passant, T. Groza, J. G. Breslin, Argumentation 3.0: how Semantic Web technologies can improve argumentation modeling in Web 2.0 environments, in: Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010., 2010, pp. 439–446. URL: `http://dx.doi.org/10.3233/978-1-60750-619-5-439`. doi:`10.3233/978-1-60750-619-5-439`.

[29] J. Schneider, A. Passant, S. Decker, Deletion discussions in Wikipedia: decision factors and outcomes, in: Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration, WikiSym 2012, Austria, August 27 - 29, 2012, 2012, p. 17. URL: `http://doi.acm.org/10.1145/2462932.2462955`. doi:`10.1145/2462932.2462955`.

[30] S. Auer, S. Dietzold, T. Riechert, OntoWiki–A Tool for Social, Semantic Collaboration, in: Proceedings of the 5th International Semantic Web Conference (ISWC 2006), volume LNCS 4273, Springer, Athens, GA, 2006.

[31] C. Ghidini, B. Kump, S. Lindstaedt, N. Mahbub, V. Pammer, M. Rospocher, L. Serafini, MoKi: The Enterprise Modelling Wiki, in: L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou, E. P. B. Simperl (Eds.), Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications 2009, Springer, Berlin, Heidelberg, 2009, pp. 831–835.

[32] V. Zacharias, S. Braun, SOBOLEO - Social Bookmarking and Lightweight Ontology Engineering, in: Workshop on Social and Collaborative Construction of Structured Knowledge (CKC), 16th International World Wide Web Conference 2007 (WWW 2007), 2007.

[33] T. Schandl, A. Blumauer, Poolparty: SKOS thesaurus management utilizing linked data, The Semantic Web: Research and Applications 6089 (2010) 421–425.

[34] M. Krötzsch, D. Vrandecic, M. Völkel, Semantic MediaWiki, in: Proceedings of the 5th International Semantic Web Conference 2006 (ISWC 2006), Springer, 2006, pp. 935–942.

[35] N. F. Noy, T. Tudorache, Collaborative ontology development on the (semantic) web., in: AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering, AAAI, 2008, pp. 63–68.

[36] T. Groza, T. Tudorache, M. Dumontier, Commentary: State of the art and open challenges in community-driven knowledge curation, Journal of Biomedical Informatics 46 (2013) 1–4. URL: `http://dx.doi.org/10.1016/j.jbi.2012.11.007`. doi:`10.1016/j.jbi.2012.11.007`.

[37] T. Tudorache, S. M. Falconer, C. I. Nyulas, N. F. Noy, M. A. Musen, Will Semantic Web technologies work for the development of ICD-11?, in: Proceedings of the 9th International Semantic Web Conference (ISWC 2010), ISWC (In-Use), Springer, Shanghai, China, 2010.

[38] J. Pöschko, M. Strohmaier, T. Tudorache, M. A. Musen, Pragmatic analysis of crowd-based knowledge production systems with iCAT Analytics: Visualizing changes to the ICD-11 ontology, in: Proceedings of the AAAI Spring Symposium 2012: Wisdom of the Crowd, 2012. Accepted for publication.

[39] S. Walk, J. Pöschko, M. Strohmaier, K. Andrews, T. Tudorache, C. Nyulas, M. A. Musen, N. F. Noy, PragmatiX: An Interactive Tool for Visualizing the Creation Process Behind Collaboratively Engineered Ontologies, International Journal on Semantic Web and Information Systems (2013).

[40] M. Strohmaier, S. Walk, J. Pöschko, D. Lamprecht, T. Tudorache, C. Nyulas, M. A. Musen, N. F. Noy, How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects, Web Semantics: Science, Services and Agents on the World Wide Web 20 (2013).

[41] S. M. Falconer, T. Tudorache, N. F. Noy, An analysis of collaborative patterns in large-scale ontology development projects., in: M. A. Musen, . Corcho (Eds.), K-CAP, ACM, 2011, pp. 25–32.

[42] C. Pesquita, F. M. Couto, Predicting the extension of biomedical ontologies, PLoS Comput Biol 8 (2012) e1002630. URL: `http://dx.doi.org/10.1371%2Fjournal.pcbi.1002630`. doi:`10.1371/journal.pcbi.1002630`.

[43] H. Wang, T. Tudorache, D. Dou, N. F. Noy, M. A. Musen, Analysis of user editing patterns in ontology development projects, in: On the Move to Meaningful Internet Systems: OTM 2013 Conferences, Springer, 2013, pp. 470–487.

[44] S. Walk, P. Singer, M. Strohmaier, T. Tudorache, M. A. Musen, N. F. Noy, Discovering Beaten Paths in Collaborative Ontology-Engineering Projects using Markov Chains, Journal of Biomedical Informatics (2014).

[45] C. Debruyne, N. Nijs, Using a reputation framework to identify community leaders in ontology engineering, in: On the Move to Meaningful Internet Systems: OTM 2013 Conferences, Springer, 2013, pp. 677–684.

[46] P. De Leenheer, C. Debruyne, J. Peeters, Towards social performance indicators for community-based ontology evolution, in: Proceedings of Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK2008), collocated with the 8th International Semantic Web Conference (ISWC 2009), Washington

DC, CEUR-WS, Citeseer, 2009, pp. 38–48.

[47] S. Van Laere, R. Buyl, M. Nyssen, A Method for Detecting Behavior-Based User Profiles in Collaborative Ontology Engineering, in: On the Move to Meaningful Internet Systems: OTM 2014 Conferences, Springer, 2014, pp. 657–673.

[48] C. Di Francescomarino, C. Ghidini, M. Rospocher, Evaluating Wiki Collaborative Features in Ontology Authoring, IEEE Transactions on Knowledge & Data Engineering (2014) 2997–3011.

[49] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proceedings of the Eleventh International Conference on Data Engineering, ICDE '95, IEEE Computer Society, Washington, DC, USA, 1995, pp. 3–14. URL: `http://dl.acm.org/citation.cfm?id=645480.655281`.

[50] R. Srikant, R. Agrawal, Mining sequential patterns: Generalizations and performance improvements, in: Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '96, Springer-Verlag, London, UK, UK, 1996, pp. 3–17. URL: `http://dl.acm.org/citation.cfm?id=645337.650382`.

[51] H. Mannila, H. Toivonen, A. I. Verkamo, Discovery of frequent episodes in event sequences, Data Mining and Knowledge Discovery 1 (1997) 259–289.

[52] J. T.-L. Wang, G.-W. Chirn, T. G. Marr, B. Shapiro, D. Shasha, K. Zhang, Combinatorial pattern discovery for scientific data: Some preliminary results, in: Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, SIGMOD '94, ACM, New York, NY, USA, 1994, pp. 115–125. URL: `http://doi.acm.org/10.1145/191839.191863`. doi:`10.1145/191839.191863`.

[53] C. Bettini, X. S. Wang, S. Jajodia, Testing complex temporal relationships involving multiple granularities and its application to data mining (extended abstract), in: Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS '96, ACM, New York, NY, USA, 1996, pp. 68–78. URL: `http://doi.acm.org/10.1145/237661.237680`. doi:`10.1145/237661.237680`.

[54] M. J. Zaki, Spade: An efficient algorithm for mining frequent sequences, Machine learning 42 (2001) 31–60.

[55] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, ACM, New York, NY, USA, 2000, pp. 1–12. URL: `http://doi.acm.org/10.1145/342009.335372`. doi:`10.1145/342009.335372`.

[56] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu, PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, in: Proceedings of the 17th International Conference on Data Engineering, ICDE '01, IEEE Computer Society, Washington, DC, USA, 2001, pp. 215–224. URL: `http://dl.acm.org/citation.cfm?id=876881.879716`.

[57] C.-M. Hsu, C.-Y. Chen, B.-J. Liu, C.-C. Huang, M.-H. Laio, C.-C. Lin, T.-L. Wu, Identification of hot regions in protein-protein interactions by sequential pattern mining, BMC bioinformatics 8 (2007) S8.

[58] J. Borges, M. Levene, Evaluating variable-length markov chain models for analysis of user web navigation sessions, IEEE Trans. on Knowl. and Data Eng. 19 (2007) 441–452. URL: `http://dx.doi.org/10.1109/TKDE.2007.1012`. doi:`10.1109/TKDE.2007.1012`.

[59] M. Deshpande, G. Karypis, Selective markov models for predicting web page accesses, ACM Trans. Internet Technol. 4 (2004) 163–184. URL: `http://doi.acm.org/10.1145/990301.990304`. doi:`10.1145/990301.990304`.

[60] R. Lempel, S. Moran, The stochastic approach for link-structure analysis (salsa) and the tkc effect, Comput. Netw. 33 (2000) 387–401. URL: `http://dx.doi.org/10.1016/S1389-1286(00)00034-7`. doi:`10.1016/S1389-1286(00)00034-7`.

[61] P. L. T. Pirolli, J. E. Pitkow, Distributions of surfers' paths through the world wide web: Empirical characterizations, World Wide Web 2 (1999) 29–45. URL: `http://dx.doi.org/10.1023/A:1019288403823`. doi:`10.1023/A:1019288403823`.

[62] R. Sen, M. Hansen, Predicting a web user's next access based on log data, Journal of Computational Graphics and Statistics 12 (2003) 143–155. URL: `http://citeseer.ist.psu.edu/sen03predicting.html`.

[63] I. Zukerman, D. W. Albrecht, A. E. Nicholson, Predicting users' requests on the www, Proceedings of the Seventh International Conference on User Modeling, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999, pp. 275–284. URL: `http://dl.acm.org/citation.cfm?id=317328.317370`.

[64] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in: Proceedings of the seventh international conference on World Wide Web 7, WWW7, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1998, pp. 107–117.

[65] J. Borges, M. Levene, Data mining of user navigation patterns, in: Revised Papers from the International Workshop on Web Usage Analysis and User Profiling, WEBKDD '99, Springer-Verlag, London, UK, UK, 2000, pp. 92–111. URL: `http://dl.acm.org/citation.cfm?id=648036.744399`.

[66] I. Cadez, D. Heckerman, C. Meek, P. Smyth, S. White, Model-based clustering and visualization of navigation patterns on a web site, Data Min. Knowl. Discov. 7 (2003) 399–424. URL: `http://dx.doi.org/10.1023/A:1024992613384`. doi:`10.1023/A:1024992613384`.

[67] R. R. Sarukkai, Link prediction and path analysis using markov chains, Proceedings of the 9th international World Wide Web conference on Computer networks: the international journal of computer and telecommunications netowrking, North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2000, pp. 377–386. URL: `http://dl.acm.org/citation.cfm?id=347319.346322`.

[68] F. Chierichetti, R. Kumar, P. Raghavan, T. Sarlos, Are web users really markovian?, in: Proceedings of the 21st international conference on World Wide Web, WWW '12, ACM, New York, NY, USA, 2012, pp. 609–618. URL: `http://doi.acm.org/10.1145/2187836.2187919`. doi:`10.1145/2187836.2187919`.

[69] N. F. Noy, A. Chugh, W. Liu, M. A. Musen, A Framework for Ontology Evolution in Collaborative Environments, in: Proceedings of the 5th International Semantic Web Conference (ISWC 2006), volume LNCS 4273, Springer, Athens, GA, 2006, pp. 544–558.

[70] H. Tong, Determination of the order of a markov chain by akaike's information criterion, J. Appl. Prob. 12 (1975) 488–497. URL: `http://www.citeulike.org/user/bgamari/article/10385495`.

[71] P. Gates, H. Tong, On markov chain modeling to some weather data, Journal of Applied Meteorology and Climatology 15 (1976) 1145–1151.

[72] R. W. Katz, On some criteria for estimating the order of a markov chain, Technometrics 23 (1981) 243–249.

[73] S. I. Vrieze, Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC)., Psychological methods 17 (2012) 228.

3.3 Discovering Beaten Paths in Collaborative Ontology-Engineering Projects using Markov Chains

In this article, colleagues and I uncover how large ontology-engineering projects from the biomedical domain unfold. To that end, the article presents an analysis of the change-logs of five different collaborative ontology-engineering projects of varying sizes and scopes from the biomedical domain using the Markov chain framework introduced in Section 3.2 and Walk et al. [2015b]. From the results of this analysis, the article further lists commonalities and differences between different projects that have implications for project managers, editors, developers and contributors working on or with collaborative ontology-engineering projects and tools.

In particular, colleagues and I investigate patterns in edit sequences of contributors from collaborative ontology-engineering projects by inspecting the transition matrices of fitted first-order Markov chain models. The presented analysis is aimed at tackling and addressing the second research question.

The results of the analyses indicate that it is not only possible to identify, extract and interpret various edit patterns when only fitting and investigating first-order Markov chains, but also that these models are likely to provide potential actionable information for project managers and ontology-engineering tool developers.

Discovering Beaten Paths in Collaborative Ontology-Engineering Projects using Markov Chains

Simon Walk[a,*], Philipp Singer[b], Markus Strohmaier[b,c], Tania Tudorache[d], Mark A. Musen[d],
Natalya F. Noy[d]

[a]*Institute for Information Systems and Computer Media, Graz University of Technology, Austria*
[b]*GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany*
[c]*Dept. of Computer Science, University of Koblenz-Landau, Germany*
[d]*Stanford Center for Biomedical Informatics Research, Stanford University, USA*

Abstract

Biomedical taxonomies, thesauri and ontologies in the form of the International Classification of Diseases as a taxonomy or the National Cancer Institute Thesaurus as an OWL-based ontology, play a critical role in acquiring, representing and processing information about human health. With increasing adoption and relevance, biomedical ontologies have also significantly increased in size. For example, the 11^{th} revision of the International Classification of Diseases, which is currently under active development by the World Health Organization contains nearly $50,000$ classes representing a vast variety of different diseases and causes of death. This evolution in terms of size was accompanied by an evolution in the way ontologies are engineered. Because no single individual has the expertise to develop such large-scale ontologies, ontology-engineering projects have evolved from small-scale efforts involving just a few domain experts to large-scale projects that require effective collaboration between dozens or even hundreds of experts, practitioners and other stakeholders. Understanding the way these different stakeholders collaborate will enable us to improve editing environments that support such collaborations. In this paper, we uncover how large ontology-engineering projects, such as the International Classification of Diseases in its 11^{th} revision, unfold by analyzing usage logs of five different biomedical ontology-engineering projects of varying sizes and scopes using Markov chains. We discover intriguing interaction patterns (e.g., which properties users frequently change after specific given ones) that suggest that large collaborative ontology-engineering projects are governed by a few general principles that determine and drive development. From our analysis, we identify commonalities and differences between different projects that have implications for project managers, ontology editors, developers and contributors working on collaborative ontology-engineering projects and tools in the biomedical domain.

Keywords: Collaborative ontology engineering; Markov chains; sequential patterns; collaboration; ontology-engineering tool; user interface

---

[*] Corresponding author (`s.walk@student.tugraz.at`)

1. Introduction

Today, biomedical ontologies play a critical role in acquiring, representing and processing information about human health. For example, the International Classification of Diseases (ICD) is a taxonomy that is used in more than 100 countries to encode patient diseases, to compile health-related statistics and to collect health-related spending statistics. Similarly, the National Cancer Institute's Thesaurus (NCIt) represents an important OWL-based vocabulary for classifying cancer and cancer-related terms.

With their increase in relevance, biomedical taxonomies, thesauri and ontologies have also significantly increased in size to cover new findings and to extend and complement their original areas of application. For example, the 11th revision of the International Classification of Diseases (ICD-11), currently under active development by the World Health Organization (WHO), consists of nearly $50,000$ classes representing a vast variety of different diseases and causes of death. In contrast to previous revisions, the foundation component of ICD-11 is implemented as an OWL ontology with a broader scope than previous ICD revisions.

This growth was accompanied by a need to adapt the way these ontologies are engineered as no single individual or small group of domain experts have the expertise to develop such large-scale ontologies. New tools and processes have to be developed in order to coordinate, augment and manage collaboration between the dozens or hundreds of experts, practitioners and stakeholders when engineering an ontology.

Understanding the ways in which such a large number of participants – e.g., more than 100 experts contribute to ICD-11 – collaborate with one another when creating a structured knowledge representation is a prerequisite for quality control and effective tool support.

Objectives: Consequently, we aim at understanding how large collaborative ontology-engineering projects such as ICD-11 unfold. In particular, we want to investigate if we can identify usage patterns in the change-logs of collaborative ontology-engineering projects? We approach this problem by analyzing patterns in usage logs of five biomedical ontology-engineering projects of varying sizes and scopes. For this analysis we employ Markov chain models for investigating and modeling sequential interaction paths (c.f. Section 3.2). Such paths are represented by chronologically ordered lists of interactions within the underlying ontology for (a) a single user or (b) a single class (see Figure 2). For example, we study sequences of properties that were either changed by (a) *a single user* on any class or (b) *a single class* by any user in an ontology over time. For example, as depicted in Figure 2, a sequential property path for a single user (user-based) consists of a chronologically ordered list of all properties (e.g., *title*, *definition* etc.), which have been changed by that user on any class, while a sequential property path for a single class (class-based) consists of a chronologically ordered list of properties that were changed on that class by any user. Instead of only modeling sequences for single users or classes, our data contains a set of paths; e.g., each path in the dataset consists of sequences of properties whose value has been changed by a single user over time. This allows us to tap into accumulated patterns. Concretely, we are interested in studying emerging patterns of subsequent steps in such sequential paths – e.g., which properties do users frequently change after a specific given property.

The analyzed datasets range from large-scale datasets such as ICD-11 to smaller ones such as the Ontology for Parasite Lifecycle (OPL). Given the differences of our datasets in a number of salient characteristics, we investigate if specific patterns can be found across all or only in certain biomedical ontology-engineering projects. Furthermore, we investigate and discuss features of these projects that potentially affect observed patterns, which can only be found in specific datasets. This analysis can be seen as a stepping stone for collaborative ontology-engineering project managers to devise infrastructures and tool support to augment collaborative ontology engineering.

ICD-10 Code: D13

Sorting label: D13

ICD Title: Benign neoplasm of other and ill-defined parts of digestive system

Fully Specified Name:

Short Definition: A neoplasm of other and/or ill-defined parts of the digestive system which is characterized by the absence of morphologic features associated with malignancy (severe cytologic atypia, tumor cell necrosis, and high mitotic rate). Benign neoplasms remain confined to the original site of growth and only rarely metastasize to other anatomic sites.

Figure 1: A screenshot of iCAT, a custom tailored, web-based version of WebProtégé, developed for the collaborative engineering of ICD-11. The left part of the interface visualizes the ICD-11 class hierarchy, the class titles, the number of annotations each class has received (speech bubbles) and its overall progress (color and symbol before the class title). The right part of the interface shows the different user-interface sections (e.g, *Title & Definition* or *Classification Properties*), listing all properties and property values for each class.

Contributions: We present new insights on social interactions and editing patterns that suggest that large collaborative ontology-engineering projects are governed by a few general principles that determine and drive development. Specifically, our results indicate that general edit patterns can be found in all investigated datasets, even though they (i) represent different projects with different goals, (ii) use variations of the same ontology-editors and tools for the engineering process and (iii) differ in the way the projects are coordinated.

To the best of our knowledge, the work presented in this paper represents the most fine-grained and comprehensive study of patterns in large-scale collaborative ontology-engineering projects in the domain of biomedicine. In addition, our analysis is conducted across five datasets of different sizes, which have been developed using different versions of Collaborative Protégé (Table 1).

2. Collaborative Ontology Engineering

According to Gruber [1], Borst [2], Studer et al. [3] an ontology is an explicit specification of a shared conceptualization. In particular, this definition refers to a machine-readable construct (the formalization) that represents an abstraction of the real world (the shared conceptualization), which is especially important in the field of computer science as it allows a computer (among other things) to "understand" relationships between entities and objects that are modeled in an ontology.

Collaborative ontology engineering is a new field of research with many new problems, risks and challenges that we must first identify and then address. In general, contributors of collaborative ontology-engineering projects, similar to traditional collaborative online production systems[1] (e.g., Wikipedia), engage remotely (e.g., via the internet or a client–server architecture) in the development process to create and maintain an ontology. As an ontology represents a formalized and abstract representation of a specific domain, disagreements between authors on certain subjects can occur. Similar to face-to-face meetings, these collaborative ontology-engineering projects need tools that augment collaboration and help contributors in reaching consensus when modeling topics of the real world.

Indeed, the majority of the literature about collaborative ontology engineering sets its focus on surveying, finding and defining requirements for the tools used in these projects [4, 5].

The Semantic Web community has developed a number of tools aimed at supporting the collaborative development of ontologies. For example, Semantic MediaWikis [6] and its derivatives [7, 8, 9] add semantic, ontology modeling and collaborative features to traditional MediaWiki systems.

Protégé, and its extensions for collaborative development, such as WebProtégé and iCAT [10] (see Figure 1 for a screenshot of the iCAT ontology-editor interface) are prominent stand-alone tools that are used by a large community worldwide to develop ontologies in a variety of different projects. Both WebProtégé and Collaborative Protégé provide a robust and scalable environment for collaboration and are used in several large-scale projects, including the development of ICD-11 [11].

Pöschko et al. [12] and Walk et al. [13] have created *PragmatiX*, a tool to visualize and analyze a collaboratively engineered ontology and aspects of its history and the engineering process, providing quantitative insights into the ongoing collaborative development processes.

Falconer et al. [14] investigated the change-logs of collaborative ontology-engineering projects, showing that users exhibit specific roles, which can be used to group and classify users, when contributing to the ontology. Pesquita and Couto [15] investigated whether the location and specific structural features can be used to determine if and where the next change is going to occur in the Gene Ontology[2].

Goncalves et. al ([16, 17, 18]) performed an analysis of different versions of ontologies by applying and categorizing *Diff* algorithms, with the goal of categorizing the differences between consecutive and chronologically ordered versions of the ontologies. Furthermore, they conducted reasoner performance tests and identified factors that potentially increase reasoner performance. For the analysis presented in this paper we were able to rely on ChAO ([19]), which is a change-log provided by Protégé and its derivatives that already provides us with detailed and unambiguous logs of changes for the investigated ontologies.

In a similar context Grau et al. [20, 21] proposed a logical framework for modularity of ontologies and a definition of what is to be considered as an ontology module. In general, an ontology module can be used to extract the meaning of a specified set of terms from an ontology. Extracting the right amount of information is especially important for the topic of ontology reuse. According to Grau et al. modularity also represents a crucial factor in collaborative ontology-engineering environments as modular representations of ontologies are easier to understand, to extend and to reuse, similar to modularity in software engineering projects.

Mikroyannidi et al. [22] investigated the detection and use of (design) patterns in the content of an ontology, using a clustering approach. In contrast to Mikroyannidi et al., our analysis focuses on the detection of sequential patterns in interaction data rather than content.

---

[1] Note that the term traditional online production systems refers to online platforms that have users collaborate in engineering digital goods, opposed to a structured knowledge base that is the result of collaborative ontology-engineering.

[2] `http://www.geneontology.org`

Table 1: Detailed information of the datasets used for the sequential pattern analysis to extract beaten paths in collaborative ontology-engineering projects.

		ICD-11	ICTM	NCIt	BRO	OPL
Ontology	classes	48,771	1,506	102,865	528	393
	changes	439,229	67,522	294,471	2,507	1,993
	DL expressivity	$\mathcal{SHOIN}(\mathbf{D})$	$\mathcal{SHOIN}(\mathbf{D})$	$\mathcal{SH}$	$\mathcal{SHIF}(\mathbf{D})$	$\mathcal{SHOIF}$
Editor	tool	iCAT	iCAT-TM	Collaborative Protégé	WebProtégé	Collaborative Protégé
Users	users	109	27	17	5	3
	bots (changes)	1 (935)	1 (1)	0 (0)	0 (0)	0 (0)
Duration	first change	18.11.2009	02.02.2011	01.06.2010	12.02.2010	09.06.2011
	last change	29.08.2013	17.7.2013	19.08.2013	06.03.2010	23.09.2011
	observation period (ca.)	4 years	2.5 years	3 years	1 month	3 months

Strohmaier et al. [23] investigated the hidden social dynamics that take place in collaborative ontology-engineering projects from the biomedical domain and provides new metrics to quantify various aspects of the collaborative engineering processes. Wang et al. [24] have used association-rule mining to analyze user editing patterns in collaborative ontology-engineering projects. The approach presented in this paper uses Markov chains to extract much more fine grained user-interaction patterns incorporating a variable number of historic editing information.

The only requirement to perform the pattern analysis that we present in this paper is the availability of a structured log of changes that can be mapped to the underlying ontology. The majority of the discussed collaborative ontology-engineering environments provide such a log, allowing for a similar analysis. For example, the Semantic MediaWikis store all the changes to the articles, and thus the ontology, allowing to expand the application of Markov chains to analyze sequential patterns as shown in this paper.

3. Materials & Methods

For the analysis conducted in this paper we concentrated our efforts on five ontology-engineering projects in the biomedical domain. Each of the projects (i) has at least two users who contributed to the project, (ii) provides a structured log of changes and (iii) represents knowledge from the biomedical domain. In Section 3.1 we provide a brief history for each dataset and in Section 3.2 we describe the sequential path analysis. To aid readers in understanding the analyses conducted in this paper and its implications we provide a very brief overview of Markov chains and the involved model selection methodology in Section 3.3.

3.1. Datasets

Table 1 lists the detailed features and observation periods for the following five datasets that we used in our analysis. All datasets have been created either with WebProtégé or special versions of WebProtégé. To be able to conduct the pattern detection analysis for a different dataset, there is only one requirement that needs to be satisfied: The availability of a change-log that can be mapped onto the ontology so that changes can be associated with users and classes without ambiguity.

The DL expressivity [25, 26] of the five datasets is added to Table 1 to highlight that the investigated ontologies exhibit different strategies regarding their OWL-DL expressivity. As all levels of expressivity shown in Table 1 allow for the definition and assignment of properties and classes, they do not influence the conducted pattern detection analyses. Also, in the case of WebProtégé and its derivatives, the data used for the pattern detection analysis can be extracted from the change-logs, allowing us to prevent parsing and extracting values from OWL directly.

The International Classification of Diseases (ICD)[3] is the international standard for diagnostic classification used to encode information relevant to epidemiology, health management, and clinical use in over 100 United Nations countries. The World Health Organization (WHO) develops ICD, and publishes new revisions of the classification every decade or more. The current revision in use is ICD-10, a taxonomy that contains over $15,000$ classes. The 11th revision of ICD,[4] **ICD-11**, is currently taking place and brings two major changes with respect to previous revisions. First, ICD-11's foundation component is developed as an OWL ontology using a much richer representation formalism than previous revisions. ICD-11 contains very detailed descriptions of several aspects of diseases, mostly represented as properties in the ontology. Second, the development of ICD-11 takes place in a Web-based collaborative environment, called iCAT (see Figure 1), which allows domain experts around the world to contribute and review the ontology online. ICD-11 is planned to be finalized in May 2017.

The International Classification of Traditional Medicine (ICTM) is a WHO led project that aimed to produce an international standard terminology and classification for diagnoses and interventions in Traditional Medicine.[5] ICTM, similarly to ICD-11, is implements an OWL based ontology as foundation component, which tries to unify the knowledge from the traditional medicine practices from China, Japan and Korea. Its content is authored in 4 languages: English, Chinese, Japanese and Korean. More than 20 domain experts from the three countries developed ICTM using a customized version of the iCAT system, called iCAT-TM. The development of ICTM was stopped in 2012, and a subset of ICTM is also included as a branch in the ICD-11 ontology.[6]

The National Cancer Institute's Thesaurus (NCIt) [27] has over $100,000$ classes and has been in development for more than a decade. It is a reference vocabulary covering areas for clinical care, translational, basic research, and cancer biology. A multidisciplinary team of editors works to edit and update the terminology based on their respective areas of expertise, following a well-defined workflow. A lead editor reviews all changes made by the editors. The lead editor accepts or rejects the changes and publishes a new version of the NCI Thesaurus. The NCI Thesaurus is , at its core, an OWL ontology, which uses many OWL primitives such as defined classes and restrictions. It was named thesaurus due to historical reasons, however fully conforms to OWL semantics, thus represents an actual ontology.

The Biomedical Resource Ontology (BRO) originated in the Biositemaps project,[7] an initiative of the Biositemaps Working Group of the NIH National Centers for Biomedical Computing [28]. Biositemaps is a mechanism for researchers working in biomedicine to publish metadata about biomedical data, tools, and services. Applications can then aggregate this information for tasks such as semantic search. BRO is the enabling technology used in Biositemaps; a controlled terminology for describing the resource types, areas of research, and activity of a biomedical related resource. BRO was developed by a small group of editors, who use a Web-based interface (WebProtégé) to modify the ontology and to carry out discussions to reach consensus on their modeling choices.

The Ontology for Parasite Lifecycle (OPL) models the life cycle of the *T.cruzi*, a protozoan parasite, which is responsible for a number of human diseases. OPL is an OWL ontology that

---

[3] `http://www.who.int/classifications/icd/en/`

[4] `http://www.who.int/classifications/icd/ICDRevision/`

[5] `http://tinyurl.com/ictmbulletin`

[6] The ICD-11 dataset used in our analysis did not include the ICTM branch.

[7] `http://biositemaps.ncbcs.org`

extends several other OWL ontologies. It uses many OWL constructs such as restrictions and defined classes. Several users from different institutions collaborate on OPL development. This ontology is much smaller and has far fewer users than NCIt, ICD-11, or ICTM.

3.2. Sequential Interaction Paths

For our sequential pattern analysis we analyze three different kinds of paths, which all represent interactions with the underlying ontology. A sequential path is represented by the chronologically ordered list of extracted interactions for either a single user or a single class (see Figure 2). For example, a sequential property path for a single user (user-based) consists of a chronologically ordered list of all properties (e.g., *title*, *definition* etc.), which have been changed by that user on any class, while a sequential property path for a single class (class-based) consists of a chronologically ordered list of properties that were changed on that class by any user.

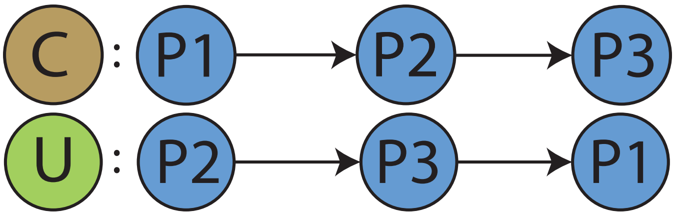


Figure 2: The top row of the figure depicts an exemplary **class-based** sequential property path ($P1$ to $P3$) for class C. This means that for class C the property $P1$ was changed first, then property $P2$ and most recently changed was property $P3$. The bottom row of the figure depicts the sequential property path ($P1$ to $P3$), however this time for a user U (**user-based**). Analogously, user U has first changed $P2$, continued to change property $P3$ and most recently changed $P1$.

User-Sequence Paths: First, we analyze activity patterns within the collaborative ontology-engineering project. This means that we analyze sequences of users who change a class. We want to detect and describe the different sequential patterns (the structure) that can be extracted from the change-logs of the investigated collaborative ontology-engineering projects

Structural Paths: Analogously to the User-Sequence Paths, we investigate edit-strategies, such as *bottom-up* or *top-down* development, that users follow. Is it possible to detect common patterns of which depth level a user frequently contributes to after a given current depth level? In addition to development-strategies, we look at the relationships (e.g., parent, child, sibling etc.) between the current and the next class a user is going to contribute to.

Property Paths: On a content-based level, we investigate the series of property-changes users perform on. In particular, we want to identify common successive property-changes – i.e., which properties *users* (user-based) regularly change consecutively and which properties are changed back-to-back for *classes* (class-based).

3.3. Markov Chain Models

For the analysis conducted in this paper we are adopting the methodology presented by Singer et al. [29] and mapped to collaborative ontology-engineering change logs by Walk et al. [30] to detect sequential patterns identified in and extracted from change-logs of collaborative ontology-engineering projects.

For a better understanding of the collected results, we will provide a short description of Markov chains. For an in-depth description of our methodology we point to Singer et al. [29], Walk et al. [30].

In general, Markov chain models are used for stochastically modeling transitions between states on a given state space. In our case, a Markov chain consists of a finite *state-space* (e.g., properties that a user edits over time; see Section 3.2) and the corresponding *transition probabilities* (e.g., the probability of changing property j after property i) between these states. Markov chain models are usually described as memoryless which means that the next state in a sequences only depends on the current one and not on a sequence of preceding ones (also known as Markovian property). Hence, this property defines serial dependence between adjacent nodes in trajectories – this is where the term "chain" comes from. Such a model is usually called a *first-order* or *memoryless* model.

As we are interested in modeling sequential interaction paths of collaborative ontology-engineering projects (see Section 3.2), we fit a Markov chain model on such sequences $D = (x_1, x_2, ..., x_n)$ with states from a finite set S. Then, we can write the Markovian property as:

$$P(x_{n+1}|x_1, x_2, ..., x_n) = P(x_{n+1}|x_n) \tag{1}$$

After the model fitting on the data, a Markov chain model is usually represented via a stochastic transition matrix P with elements $p_{ij} = P(x_j|x_i)$ where it holds that for all i:

$$\sum_j p_{ij} = 1 \tag{2}$$

For our analysis, we will make use of these transition probabilities to identify likely transitions for a variety of different states[8]. For example, if we fit the Markov chain model on sequential property paths for users (see Section 3.2), element p_{ij} of the transition matrix would tell us the probability that users change property j right after i (e.g., in 60% of all cases). By now, e.g., looking for the highest transition probabilities from state i to all other states of S, we can identify potential high-frequent patterns in our data.

4. Results

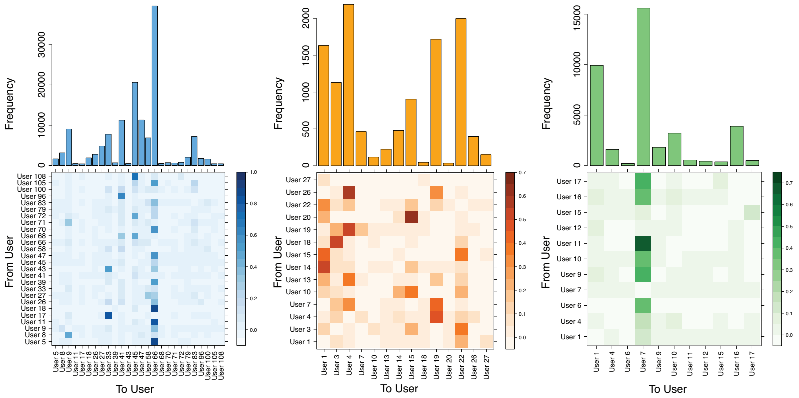
4.1. User-Sequence Paths

In the *User-Sequence Paths* analysis we investigate patterns emerging when looking at sequences of users who contribute to a class of an ontology. Hence, given a sequence of n contributors for a class over time, we identify consecutive users who edit the class (e.g., user Y frequently contribute to a class after user X).

Analyzing the chronologically ordered list of contributors for each class of the five investigated datasets provides the necessary information to identify users who perform changes on classes after (or before) other users. Note that this analysis on its own, without regarding additional factors, such as the changed property or the performed change-action, does not provide information about actual collaboration. The results of this analysis could be used to potentially

---

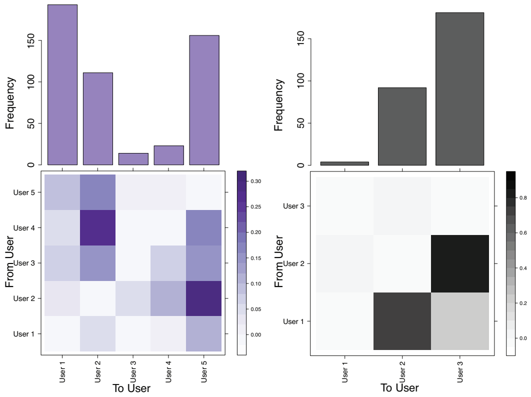
[8] Note that throughout this article we usually refer to the entities modeled (i.e., interactions) instead of states. However, we speak about transition probabilities between these entities as we derive them directly from the resulting model transition matrix.



(a) International Classification of Diseases (ICD-11)

(b) International Classification of Traditional Medicine (ICTM)

(c) National Cancer Institute Thesaurus (NCIt)



(d) Biomedical Resource Ontology (BRO)

(e) Ontology for Parasite Lifecycle (OPL)

Figure 3: **Results for the *User-Sequence Paths* analysis:** The columns and rows of the transition maps (**bottom area** of Figures 3(a) to 3(e)) represent the transition-probabilities between the users of each dataset for a first-order Markov chain, where rows are *source users* and columns are *target users*. A sequence (or transition-probability) is always read *from row to column*. Darker colors represent higher transition-probabilities while lighter colors indicate lesser transition-probabilities. Absolute probability values are dependent on the number of investigated rows and columns, hence relative differences are of greater importance. Darker colored columns identify gardeners, a contributor focused on pruning ontology classes and fixing syntactical errors. The histograms (**top area** of Figures 3(a) to 3(e)) show the number of changes performed by each user (again for a first-order Markov chain) within the five ontologies in alphabetical order. Note, that the y-axes for all histograms are scaled differently for each dataset. All datasets have a few users who contributed the majority of changes, while the rest of the users (the long-tail) only contributed a very small number of changes. Note that the transition-probabilities depicted in the transition maps are relative numbers for each column and row individually. The sum of all transition probabilities for one row in the transition maps is 1. For example, if *User 1* exhibits a transition probability of 0.30 to another *User 2* it means that *User 2* has a 30% probability of changing a class after *User 1*. Thus, an inspection of the transition maps **and** histograms is necessary for proper interpretation. To increase readability we have removed users from the plots who have contributed only a very limited number of changes for ICD-11, ICTM and NCIt.

identify users who work on the same classes, however, we do not know if they actually collaborate with or just clean up (i.e., a *gardener*, a contributor focused on pruning ontology classes and fixing syntactical errors) after other users.

Path & Model Description: To analyze user sequences, we iterated over each class of our datasets and extracted a chronologically ordered list of contributors. For example, a given path for a given class can look like the following: *User A, User B, User B, User C*. As we are interested in uncovering patterns of distinct users, we merged multiple consecutive changes by the same user into a single change – our previous example would then unfold into: *User A, User B, User C*. By doing so we remove biases emerging when one single user consecutively changes the same class over and over as this may result in unreasonable high transition probabilities between equal users.

We fit a first-order Markov chain model on this set of paths, where each path represents a single class of the ontology and each element of a path constitutes a change by a single user on the class. The resulting transition probabilities between users then e.g., tell us the probability that *User B* changed a class after *User A*. Hence, they give us thorough insights into frequent consecutive user patterns that emerge when looking at which users contribute to classes in an ontology. Due to reasons of privacy we obfuscated the usernames and replaced them with generic names.

Results: When investigating the transition probabilities (representing a Markov chain of first order) between contributors (see bottom area of Figures 3(a) to 3(e)) we can identify very active users by looking at darker colored columns of the transition maps. Note that these darker colored columns can also be used to identify gardeners, a contributor focused on pruning ontology classes and fixing syntactical errors. As we have merged all consecutive changes of the same user into one single change, the diagonal, representing the transition probabilities between the same users, is 0. The absolute transition probabilities, depicted next to each transition map, are dependent on the absolute amount of observations and users, thus are to be interpreted relatively to each other for each row individually. When looking at the probabilities between the three most active users (being users 66, 45 and 47), and all corresponding target users in ICD-11 we can see that the probabilities are very evenly distributed among them. Meaning that, when investigating the rows (*From User*) that correspond to the top three most active users, probabilities to all target users (*To User*) are very evenly distributed, with very minor exceptions. This indicates that users who contribute many changes to ICD-11 are not followed by specific other contributors, but exhibit an even distribution of users that edited a class after them. Nonetheless, we can clearly identify *User 66* to be the most likely user that edits a class after nearly all other users. This suggests, that *User 66* may represent a gardener, a contributor focused on pruning ontology classes and fixing syntactical errors, in ICD-11.

For NCIt we can clearly observe that *User 7* appears to be a *gardener*, who is checking all the changes contributed by all other users. For BRO *Users 2* and *5* are prominent target users, evident in the high transition probabilities as *To User* (dark columns) – i.e., they frequently edit a class after other users do. Interestingly, the user with the highest number of changes (*User 1*) exhibits very low and evenly distributed transition probabilities (row) and is not necessarily the user that most likely changes a class after another users. This shows us that there does not need to be a necessary connection between the overall activity of users and their activity as a gardener. This could also mean that *User 1* is possibly working independently from the other users in BRO, or that *User 1* is a domain specialist and all other users only change concepts that have not been worked on by that specialist. However, further investigations in future work are required to confirm this observation as our Markov chain analysis is not able to determine this kind of distinction. For OPL we can observe that *User 3* frequently changes the same classes after *User 2*. A similar observation can be made for *Users 1* and *2*. However, one has to keep in mind that *User 1* has contributed a limited number of changes, rendering the observed transition probabilities less useful as they rarely occur.

The histograms (see top area of Figures 3(a) to 3(e)) indicate that a small number of users contribute the majority of changes (similar to a long-tail distribution). However, this appears to be more dominant for specific ontologies compared to others. In order to measure the inequality among contributions of changes to a specific ontology by users, we analyzed the *Normalized Entropy*[9], which is determined by calculating the *Shannon Entropy* and normalizing the entropy by dividing by the logarithm of the length (i.e., number of users) of a distribution. This coefficient measures the statistical dispersion of a distribution – i.e., the coefficient is one if all users contributed equally to the ontology, while it is zero in case of total inequality where a single user conducts all changes. The results indicate that ICD-11 (0.55) exhibits a low entropy value, i.e., the changes are dominated by only a few users. For NCIt (0.61), OPL (0.64) and ICTM (0.68) we receive medium normalized entropies indicating a more democratic contribution to the ontology by users. A high entropy can be observed for BRO (0.81), which indicates that it is a demographically edited ontology – even though there are only five users.[10]

Interpretation & Practical Implications: The transition probabilities for a first-order Markov chain unveil the roles of certain users and can help to identify users or even groups of users who frequently change the same classes. Users that frequently change classes after other users (i.e., exhibit high transition probabilities in their columns) were identified by us as actual gardeners, curators and administrators of the corresponding projects. If certain users always change the same classes after specific other users, it could be worthwhile for project administrators to investigate if these users are actually collaborating, for example by looking at the changed properties and property values, or if a single user is always cleaning up after the other user. In all datasets we were able to observe at least one user who contributed a high number of changes, with evenly distributed transition probabilities to all remaining users. This observation indicates that in all projects, gardeners, curators and administrators are assigned (directly or indirectly) certain parts of the ontology; otherwise the transition probabilities between the very active users would be higher.

The ability of understanding who is most likely going to change a specific class next, as well as the classes that a user is most likely to change next could be used by project administrators to help users in finding and identifying classes (and thus work) of interest. On the other hand, the information about the next, most probable contributor for a class, can even be used to create automatic class recommender systems to suggest work to users, which could help to increase participation. However, these two analyses are beyond the scope of this paper and are therefore subject to future work. In particular for projects the size of ICD-11 and NCIt, mechanisms to automatically identify and assign work are highly useful as it is still very time-consuming to find pending work and users with the necessary knowledge to address the identified work-tasks.

4.2. Structural Paths

The investigation of *Structural Paths* involves an analysis of different aspects regarding how and where users contribute to the ontology, such as the depth level of the class that users con-

---

[9] Additionally, we calculated the *Gini Coefficient* for each distribution confirming the results presented here.

[10] Note that we do not necessarily know whether the differences between these distributions are statistically significant as we are mainly interested in the behavior of single distributions.

tribute to next (Section 4.2.1) as well as looking at the relationship distances between consecutively changed classes (Section 4.2.2).

4.2.1. Depth-Level Paths

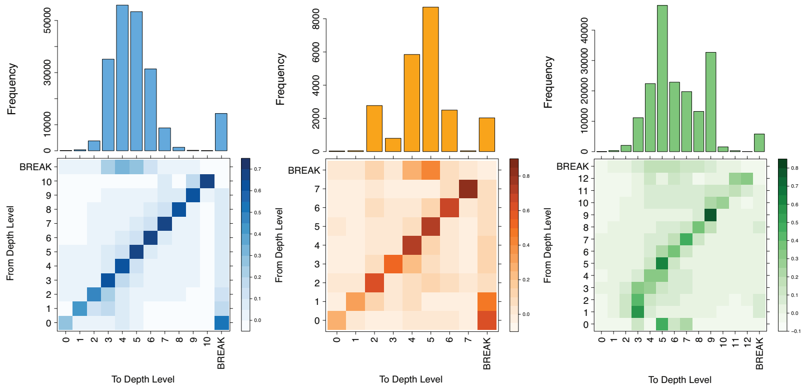
In this analysis, we investigate if users concentrate their efforts on specific depth levels of the ontology and if there are certain depth levels that are frequently consecutively changed and receive less concentrated workflows. The gathered results provide the necessary information to implement prefetching mechanisms, potentially helping to minimize the loading and waiting times for contributors. Furthermore, we can determine whether users move along the structure of the underlying ontology when editing classes.

Path & Model Description: For this analysis, we stored the chronologically ordered depth levels of each changed class for each user (user-based). The depth level of a class is the length of the shortest path between the *root node* of the ontology and the corresponding class. For example, a given path for a given user can look like the following: *Depth 3 (for class A), Depth 3 (for class A), Depth 3 (for class A), Depth 3 (for class B), Depth 4 (for class C).* We merged consecutive changes that were conducted by the same user on the same class into one single sequent change between the same depth levels. Hence, for our previous example we would merge the three successive changes of class A into just two consecutive ones which results in the following final depth-level path: *Depth 3, Depth 3, Depth 3, Depth 4.* This approach helps us to investigate patterns of changing distinct depth levels while still retaining the notion of users consecutively editing the same classes.

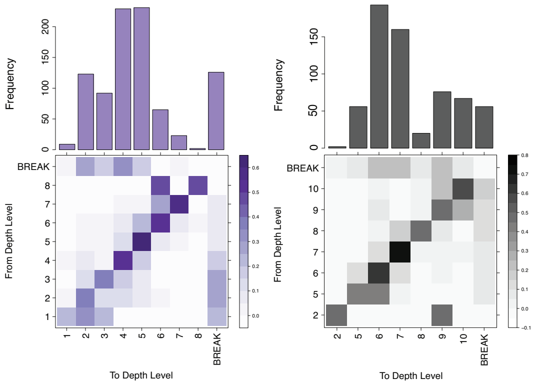
Consequently, we fit a first-order Markov chain model on these paths – each path represents a single user and each element of a path represents a corresponding depth level of a class the user has changed. The final transition probabilities give us information about consecutive depth levels that users change over time. For example, they might tell us the probability that users change a class belonging to the third depth level of the ontology after one that has a depth level of 2.

Results: First, the histograms (see top area of Figures 4(a) to 4(e)) show that work is concentrated on certain depth levels of the ontology, with the highest and lowest levels not receiving as much attention as the levels in-between.

As depicted in the transition maps (bottom area of Figures 4(a) to 4(e)), users have a high tendency to edit classes in the same depth levels, visible in the darker colored diagonal. In ICD-11, for the first five depth levels, users appear to have a tendency towards *top-down* editing, evident in the darker immediately right of the diagonal, while this tendency turns around into a *bottom-up* editing behavior, evident in the darker colored squares immediately left of the diagonal, at a depth level of 6 and higher, and appears to be strictly limited to surrounding depth levels. For ICTM (see Figure 4(b)), we can observe a similar trend, again with the tendency towards *top-down* editing appearing to be minimally more dominant. For NCIt, when only looking at the transition map, we can identify a trend towards *bottom-up* editing, evident in the squares directly left of the diagonal being darker than the ones right of the diagonal. However, when also considering the absolute number of changes, depicted in the histogram of Figure 4(c), we can infer that the levels with a higher frequency of occurrence, even though their transition probabilities are more evenly distributed, have a greater impact on the editing strategy. This means that while we can see a *bottom-up* editing behavior for levels 8 to 5 and a *top-down* editing behavior for levels 1 to 4, classes on levels 1 to 4 are more frequently changed than classes on the other levels, hence a tendency towards *top-down* editing can be observed. Thus, when users are not changing the same classes, they still exhibit a preference towards *top-down* editing. Given the short observation periods for BRO and OPL it is hard to infer edit strategies. However, similar to the other

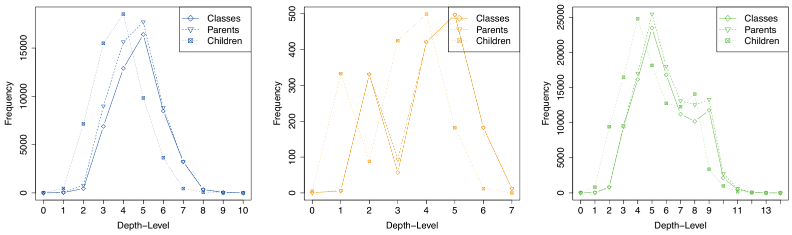


(a) International Classification of Diseases (ICD-11)
(b) International Classification of Traditional Medicine (ICTM)
(c) National Cancer Institute Thesaurus (NCIt)



(d) Biomedical Resource Ontology (BRO)
(e) Ontology for Parasite Lifecycle (OPL)

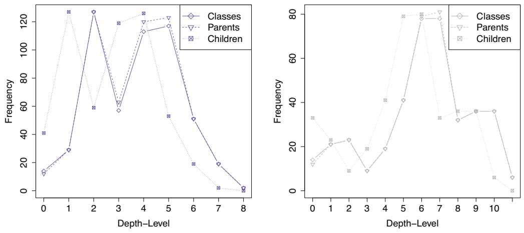
Figure 4: **Results for the *Depth-Level Paths* analysis:** The columns and rows of the transition maps (**bottom area** of Figures 4(a) to 4(e)) represent the transition probabilities of a first-order Markov chain between depth levels, where rows are *source depth levels* and columns are *target depth levels*. A sequence (or transition probability) is always read *from row to column*. Darker colors represent higher transition probabilities while lighter colors indicate lesser transition-probabilities. Absolute probability values are dependent on the number of investigated rows and columns, hence relative differences are of greater importance. For classes closer to root a *top-down* editing manner can be observed, while this is reversed for classes further away from root. The sum of all transition probabilities for one row in the transition maps is 1. For example, if *Depth-Level 6* exhibits a transition probability of 0.30 to another *Depth-Level 5* it means that a class on *Depth-Level 5* has a 30% probability of being changed after a class on *Depth-Level 6*. The histograms (**top area** of Figures 4(a) to 4(e)) show the number of changes performed in each depth level aggregated over all users of the respective projects (again for a first-order Markov chain). Throughout all projects, classes located between the first and last few depth levels (in the middle) are changed substantially more frequently than others, suggesting that work is concentrated on some depth levels while others receive none to very few changes at all. Note, that the y-axes for all histograms are scaled differently for each dataset. For the x-axes (and column/rows of the transition maps) we only display depth levels which exhibit at least one change, thus, the depth level sequences are not necessarily continuous from lowest to highest depth level.



(f) International Classification of Diseases (ICD-11)

(g) International Classification of Traditional Medicine (ICTM)

(h) National Cancer Institute Thesaurus (NCIt)



(i) Biomedical Resource Ontology (BRO)

(j) Ontology for Parasite Lifecycle (OPL)

Figure 5: The **Figures 5(f) to 5(j)** depict the absolute numbers (y-axis; Frequency) of classes as well as the number of edges (*isKindOf*) to classes on the immediate higher (*parents*; closer to root) and lower (*children*; further away from root) depth level for all depth levels (x-axis; Depth-Level). According to Figures 5(f) to 5(j) the transition probabilities depicted in the transition maps correlate with the total number of edges to children and parents for each depth level across all datasets.

projects, we can observe a concentration on the same depth levels with alternating preferences towards higher and lower depth levels. Similar to ICD-11, all datasets exhibit higher transition probabilities between the immediately surrounding depth levels.

Furthermore, we investigate whether the total number of classes as well as the total number of links to the immediate higher (children; edges to classes one level further away from root) and lower (parents; edges to classes one level closer to root) depth level correlate with our findings (Figures 5(f) to 5(j)). For example, the transition map for ICD-11 (see Figure 4(a)) shows that contributors exhibit a *top-down* editing behavior for the first five depth levels, with level 5 exhibiting first signs of *bottom-up* editing. Figure 5(f) shows a higher number of possible transitions from children than parents, indicating that users are in general likelier to follow *top-down* editing-strategies when changing classes, following relationships by chance, of the first four levels. This changes for ICD-11 at level 5, with a higher number of transitions to parents than to children, and continues until level 10. Resulting in a higher probability of users performing *bottom-up* editing-strategies when changing classes from levels 6 to 10. The same observations can be made for all other datasets, indicating that the class hierarchy influences the edit behavior of contributors.

In all datasets, after taking a *BREAK* (representing an artificially introduced session break when two consecutive changes of the same user are more than 5 minutes apart; for more information see Section 5.4), users exhibit a clear tendency towards changing classes on certain depth levels (e.g., levels 3 to 5 for ICD-11, levels 4 to 5 for ICTM, levels 4 to 7 for NCIt, levels 2 to 4 for BRO and levels 6 to 9 for OPL).

Interpretation & Practical Implications: The results of this analysis show if, to what extent and where (limited to locality being determined by *isKindOf* relationships) work is conducted and concentrated within the ontology. This information can potentially be used in a variety of ways, for example by ontology-engineering tool developers to adapt the interface of the ontology-engineering tool dynamically to display specific classes after users return from a *BREAK*. Project managers can adapt milestones and project progress reports to reflect the underlying editing strategies (e.g., *top-down* editing), for example by aligning progress with created branches (opposed to complete coverage). Another potential use-case for the results of this analysis involves the prefetching of content in certain environments (e.g., mobile or embedded systems) to minimize waiting times. Across all projects we can observe that classes close to and very far away from the *root* of the ontology are not edited as frequently as other classes. One explanation for this observation could be that classes in lower depth levels (closer to *root*) are mainly used as content dividers and are usually created in the beginning of a project. Thus, they may be more stable and less frequently updated. Classes at the higher depth levels (further away from *root*) on the other hand most likely require extensive expert knowledge. Hence, only a small number of users have the necessary expertise to contribute to these classes. Additionally, the absolute number of classes in the higher and lower depth levels is much lower in all investigated datasets. Note that absolute values of depth levels are less important for the interpretation of the results than their relative position (i.e., closest to root, furthest away from root etc.). For example, a class at level 6 can exhibit different behaviors in ontologies with 6 or 10 levels.

In all projects, except for NCIt, the depth levels where users start to edit the ontology after they return from a *BREAK* are similar to the ones where they stop editing before taking a *BREAK*. To be able to make that observation we have to take the absolute numbers of changes on each depth level (bottom area of Figure 4) into account when looking at the transition probabilities (top area of Figure 4). NCIt is the only dataset where users appear to be similarly likely to take a *BREAK* after changing classes across all depth levels, except for 0 and 12.

When we combine the results of this analysis with the results of the *User-Sequence Paths* (Section 4.1) we may be able to develop automatic mechanisms to curate and delegate work to users. For example, if we know that a specific user is most probably going to contribute to a class on level 3 and we have a set of classes on that level where that specific user is the most probable next user to contribute to, determined by the *User-Sequence Paths* analysis, we may combine these two observations to create class (and thus work) suggestions for users.

4.2.2. Hierarchical Relationship Paths

Given the high number of observed transitions between the same depth levels in the *Depth-Level Paths* analyses (Section 4.2.1; bottom area of Figure 4), we conducted an additional analysis investigating the relationships between the changed classes for all users. Hence, we wanted to know if all worked-on classes on the same depth-levels are siblings, cousins or any other kind of close relative? And in general, can we determine if users follow these hierarchical orders of an ontology when contributing to classes on the same depth level? To further strengthen our observation that users are actually moving along the ontological hierarchy when contributing to an ontology (see Section 4.2.1), we analyzed the relationships between the changed classes for each user. Note that whenever we talk about relationships for this analysis, we refer to the hierarchical *isKindOf* relationships between two classes, e.g., parent, child, sibling or cousin. For example, when traversing the shortest-path distance of 2, multiple different nodes can be reached, such as a grandparent (i.e., 2 times up), a grandchild (i.e., 2 times down), a sibling (i.e., 1 time up, 1 time down) or even some other relationship (e.g., 1 time down, 1 time up).

Path & Model Description: By combining the information from the *Depth-Level Paths* and the relative movement between depth levels, we inferred the hierarchical relationships between two consecutively changed classes of a single user (user-based). For example, if the difference between the depth levels of the investigated classes would be exactly the size of the shortest-path between them (with the shortest-path being > 0), the latter-changed class could either be a *Child*, a *Parent*, an *Ancestor* or a *Descendent* of the first-changed class. Given a relative *DOWN* movement (to a lower depth level) value, depending on the shortest-path value, the second class could be classified as *Child* (shortest-path of 1) or *Descendent* (shortest-path > 1). Analogously follows the definition of a *Parent* and *Ancestor* with a relative *UP* movement. A *Sibling* is defined as the two classes being (i) connected via the same parent with (ii) a shortest-path distance of 2 and (iii) both classes are located on the *SAME* depth level. A *Cousin* is used when two classes on the *SAME* depth level are connected by the same grand parent while exhibiting a shortest-path distance of 4. Every other possible combination of depth level and shortest-path was classified as *Other*. *Self* indicates that the same class that was changed last time was changed again. For example, a consecutive change of *Sibling* and *Self* means that a change was first performed on a class that is a sibling of the previous class (not displayed in this example) and then another change was performed on the same class, however now the relationship changed to *Self* as no new class was involved.

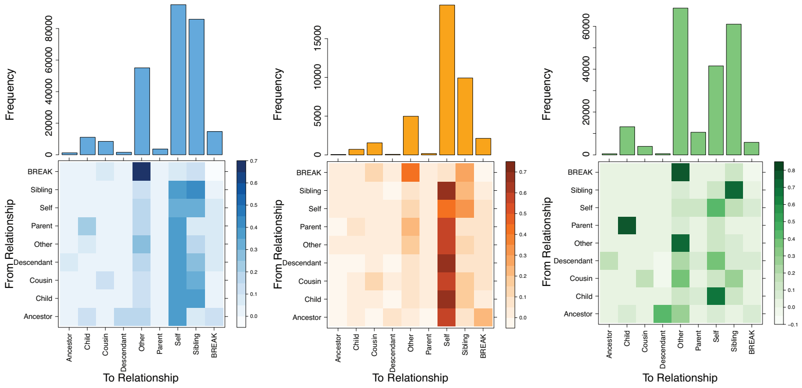
Again, consecutive changes on the same class by the same user have been merged into one single sequent change (c.f. Section 4.2.1), meaning that multiple (more than 2) consecutive changes of the same user on the same class have been merged into *Self* to *Self*. Hence, a given path for a single user can, e.g., look like the following: Sibling, Self, Self, Child.

We fit a first-order Markov chain model to the data – each path represents a single user and each element represents a hierarchical relationship between the classes changed by the user. The resulting transition probabilities of the fitted model can then give us insights into common emerging patterns. E.g., we can identify how probable it is that users change a *Sibling* after a *Child*.

Results: When looking at the histograms (see top area of Figures 6(a) to 6(e)), we can observe that the relationships *Self*, *Sibling* and *Other* are highly represented across all datasets. The transition maps (bottom area of Figures 6(a) to 6(e)) show that after a *BREAK*, across all five datasets, users tend to change classes "somewhere els" in the ontology, evident in the high transition probability from *BREAK* towards *Other*, and are likely not to resume work in the same area of the ontology that they stopped working on. For ICD-11, ICTM and OPL, no matter which relationship type occurs, users tend to edit the same class consecutively (dark colors in the *Self* column). From this *Self* relationship, which is also the one that occurs the most often in ICD-11, ICTM and OPL, users are very likely either to change the same class again (*Self*) or to change a *Sibling* of the current class.

For NCIt, BRO and OPL we can observe that users, when changing a *Parent* are very likely to change a *Child* of that parent afterwards. Note, that this *Child* does not necessarily have to be the same class that was changed prior to the traversal to *Parent*. In all datasets, except for OPL, very high transition probabilities towards *Other* can be observed for all not so frequently present relationships. In particular for NCIt we can observe that *Other* is the most frequently observed transition, even before *Self* and *Sibling*.

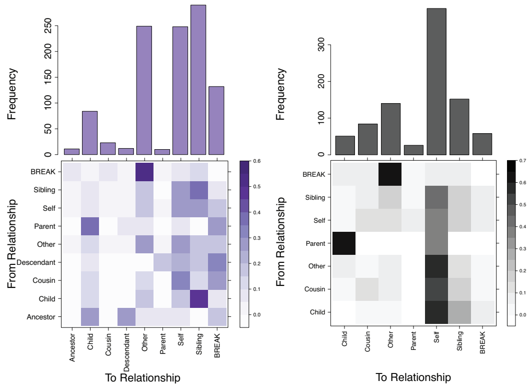
Interpretation & Practical Implications: By combining the results of this analysis with



(a) International Classification of Diseases (ICD-11)

(b) International Classification of Traditional Medicine (ICTM)

(c) National Cancer Institute Thesaurus (NCIt)



(d) Biomedical Resource Ontology (BRO)

(e) Ontology for Parasite Lifecycle (OPL)

Figure 6: **Results for the *Hierarchical-Relationship Paths* analysis:** The columns and rows of the transition maps (**bottom area** of Figures 6(a) to 6(e)) represent the transition-probabilities of a first-order Markov chain between hierarchical-relationship levels, where rows are *source relationships* and columns are *target relationships*. A sequence (or transition-probability) is always read *from row to column*. Darker colors represent higher transition-probabilities while lighter colors indicate lesser transition-probabilities. Absolute probability values are dependent on the number of investigated rows and columns, hence relative differences are of greater importance. Across all datasets, aside from *Self*, a very clear trend towards editing the ontology along *Siblings* can be observed. The histograms (**top area** of Figures 6(a) to 6(e)) show the total number of occurrences of each relationship in the corresponding datasets aggregated over all users (again for a first-order Markov chain). Note, that the *y*-axes for all histograms are scaled differently for each dataset. For the *x*-axes (and column/rows of the transition maps) we only relationships that occur at least once in the corresponding paths, thus the *x*-axes could be different from project to project. Given the very high amount of *Self* and *Sibling* transitions we can concur that users, when they contribute to classes on the same depth level follow a *breadth-first* strategy, meaning that they first concentrate their work on closely related classes (*Siblings*) on the same depth-level before switching to a different branch on the same or any other depth-level.

the results of the *Depth-Level Paths* analysis, we can infer that users exhibit a tendency towards *top-down* editing while contributing to the ontology, when only considering changes that occur on different depth levels. If they concentrate their efforts on the same depth levels, users exhibit a *breadth-first* editing behavior, meaning that they first concentrate their work on closely related classes (*Siblings*) on the same depth-level before switching to a different branch on the same or any other depth-level, either changing the same class multiple times or traversing along siblings of the current class. We can leverage this information not only to refine the previously suggested pre-fetching of classes but also to enhance possible class recommendations. Similarly, it is possible for ontology-engineering tool developers to minimize the necessary efforts of users to contribute to the ontology by implementing, for example, guided workflows that take the underlying edit strategies of the contributors into account.

As classes in ICD-11 and ICTM have a large number of properties and for ICTM certain properties have to be added in multiple languages, the high transition probabilities towards *Self* (dark colors in the *Self* column) are not surprising. One possible explanation for this observation for ICD-11 could be the special functionality available in iCAT (for ICD-11) that allows users to export parts of the ontology as spreadsheets for local editing and adding property values. Once contributors finished editing the spreadsheet they have to enter the data into the system manually, as no automatic import functionality is present. In the iCAT interface, users are simultaneously presented with the ontology tree for navigating through the classes and the corresponding properties and property values. When users select a property they can easily switch between classes, with the selected property staying selected, thus allowing to quickly enter the same properties for different classes.

A similar, yet not as dominant as in ICD-11 and ICTM, behavior can be observed for NCIt and BRO and even to some extent in OPL, which all do not use the export functionality. According to our observations, users travel along the underlying hierarchy when contributing to the ontology. Given the observations made for ICD-11 this behavior can be enforced by providing certain functionalities in the user-interface especially when they compliment the workflows of the contributors.

The results of this analysis have also shown that users are likely to pursue a certain strategy or intermediate goal for their edit sessions, for example changing all classes in a specific (narrow) area of the ontology. This is evident in the observation that after returning from a *BREAK*, users have a very high tendency to change the ontology "somewhere else" (see the transition probabilities from *BREAK* towards *Other* in the top-row of Figure 6), rather than picking up the work, where they left off. This discovery is very important for developing class-recommender, as we may use the results of this analysis to suggest closely related classes to the current class a user is working on, however when that user stays inactive for the duration defined for introducing *BREAK*s the recommendation strategy has to be changed.

4.3. Property Paths

Aside from analyzing different aspects of activity (Section 4.1) and the correlation between contribution patterns and the structure of an ontology (Section 4.2), we can use Markov chains to perform an analysis on the properties that are consecutively change by users in an ontology. This means that, for example, if a property value was edited by a user, we extracted the property (not the value) and created chronologically ordered lists of properties, whose values were changed by the corresponding users. For example, if a user changed the title of a specific class, we would extract *title*, rather than the value inserted into the title property. Now, we provide insights into emerging patterns from different viewing angles for the observations. Thus, we look at property sequences for (a) single users (user-based) and for (b) single classes (class-based) – see Section 3.2. We were not able to perform the *Property Paths* analysis on OPL and BRO as these datasets contain only a very limited number of unique property value changes during our observation periods. We also had to discard the results from NCIt, as the ontology-editing environment for NCIt provides a unique change-queuing mechanism that allows for multiple property values to be changed at the same time, making it impossible to extract chronologically ordered sequential property patterns.

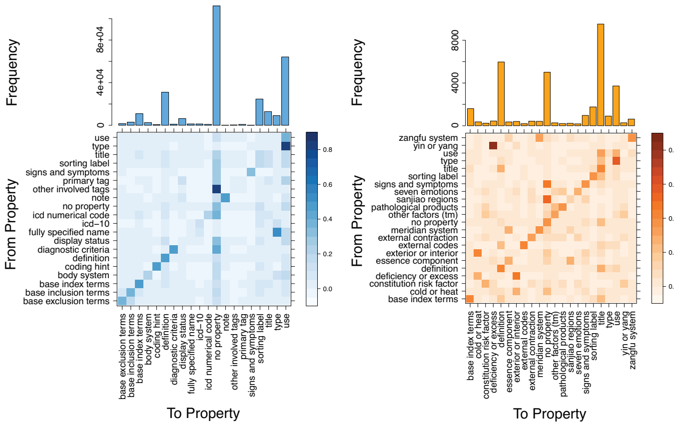
Path & Model Description: First, we extracted the properties whose values were changed in ICD-11 and ICTM, sorted either by user and timestamp or by class and timestamp. Finally, two different types of chronologically ordered property lists were extracted, one ordered per user and one ordered per class (for both datasets). The properties in *Property Paths* represent the ones which can be assigned a value for each class in ICD-11 and ICTM. Whenever a change did not modify a property (e.g., because the change action dealt with moving or creating a class) we added the element *no property* to the corresponding path. A potential path for a single user or class then may look like: *title, title, title, use*. Similar to previous analyses, if the same user has consecutively changed the same property (e.g., in the previous example *title*) on the same class, we merged these multiple changes into one successive change. Analogously, however without the restriction of the same user, if the same property was changed on the same class, we merged these changes into one sequent change. For previous example, if changes would have been performed editing the referenced properties for a single class, we would end up with the path: *title, title, use*.

Consequently, we fit a first-order Markov chain model on this set of paths (for users or classes). The final transition probabilities of the model then give us information about the probability of changing a value of one property Y after another property X either for users or for classes. For instance, we can find the property Y that most frequently has been changed after property X for classes.

Results: When looking at the histograms (top area in Figures 7(a) to 7(d)) we can see that even after removing not very frequently used properties[11], both datasets exhibit a few properties which have received a high number of changes, while the remaining majority of properties only received a very limited number of changes. For both datasets, aside from *no property*, the properties *use*, *title* and *definition* appear to be the most frequently used properties. As can be seen in the top area of Figures 7(a) and 7(b), multiple consecutive changes of the same property appear to be fairly common for both datasets. In contrast, when looking at Figures 7(c) and 7(d), which depict the transition probabilities between the sequences of properties changed by each user, we can see an even stronger trend towards consecutively changing the same properties across different classes, especially *definition*, *title* and *use*. For ICD-11 Figures 7(a) and 7(c) show that the class-based approach is less focused on consecutively changing the same property, evident in the brighter diagonal, when compared to the user-based approach. This is due to the export functionality available in iCAT combined with the manual process of inserting the same property for different classes by users of ICD-11. In contrast, such functionality is absent in ICTM, thus leading to similar behaviors for the class and user-based approaches for ICTM. The fact that a large portion of successive changes are conducted on the same property for both approaches analyzed for ICTM could also be due to the multilingual nature of the project, meaning that certain properties, such as *title* and *definition*, have to be entered multiple times in multiple languages. Similar results have been presented by Wang et al. [24], who used association rule mining techniques to analyze the change-logs of ICD-11 and ICTM.

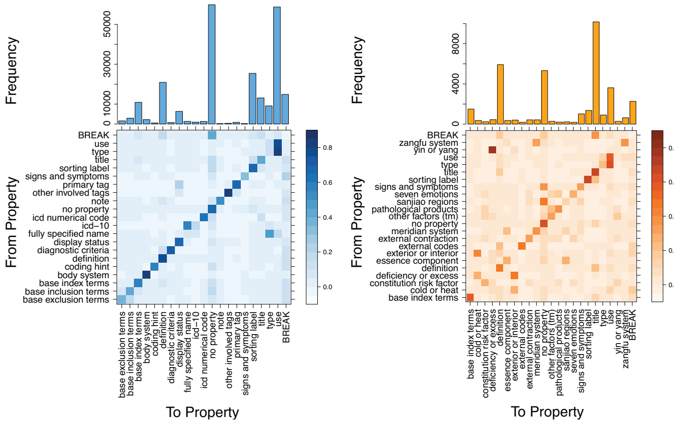
---

[11] All properties which where rarely edited have been removed from Figure 7 as they do not hold information but their removal increased the readability of the plots dramatically.



(a) International Classification of Diseases (ICD-11) (Class)

(b) International Classification of Traditional Medicine (ICTM) (Class)



(c) International Classification of Diseases (ICD-11) (User)

(d) International Classification of Traditional Medicine (ICTM) (User)

Figure 7: **Results for the *Property Paths* analysis:** The columns and rows of the transition maps (**bottom area** of Figures 7(a) to 7(d)) represent the transition-probabilities of a first-order Markov chain between consecutively changed properties, where rows are *source properties* and columns are *target properties*. Figures 7(a) and 7(c) represent class-based patterns while Figures 7(b) and 7(d) visualize user-based patterns. A sequence (or transition-probability) is always read *from row to column*. Darker colors represent higher transition-probabilities while lighter colors indicate lesser transition-probabilities. Absolute probability values are dependent on the number of investigated rows and columns, hence relative differences are of greater importance. Across all datasets a very clear trend towards consecutively editing the same properties can be observed. The histograms (**top area** of Figures 7(a) to 7(d)) show the total edits of each property in the corresponding datasets aggregated over all users and classes (again for a first-order Markov chain). Note, that the *y*-axes for all histograms are scaled differently for each dataset. As ICTM and ICD-11 only share a limited amount of properties the *x*-axes (and column/rows of the transition maps) are different from project to project. In both projects and across all 4 different approaches the *title*, *definition* and *use* properties are frequently used. Due to reasons of readability we were forced to remove properties from the plots, which exhibited only a very limited number of changes, thus did not provide substantial information for the purpose of this analysis.

Table 2: **A summary of all findings** applicable to all investigated biomedical ontologies. All listed findings are discussed in more detail in Section 5.

User-Sequence Paths (cf. Section 4.1)	**Users work in micro-workflows**	Information about which users successively change a class can be identified; i.e., information about who has edited classes in the past contains predictive information about who is going to change a class next.
	User-roles can be identified	Looking at historic data, we can identify different user roles, i.e., administrators and moderators, gardeners (a contributor focused on pruning ontology classes and fixing syntactical errors) and users that frequently interact with (collaborate/revert) each other.
Structural Paths (cf. Section 4.2)	**Users' edit behavior is influenced by the class hierarchy**	Contributors, when adding content to the ontology, are influenced by the class hierarchy.
	Users edit the ontology top-down and breadth-first	By and large, users exhibit a minor tendency towards top-down editing behavior when changing hierarchy levels while contributing. However, when staying in the same hierarchy level, contributors rather follow a *breadth-first* edit behavior, moving from one sibling of a class to the next sibling.
	Users edit closely related classes	Contributors have a very high tendency to consecutively change closely related classes, as opposed to randomly and distantly related classes.
Property Paths (cf. Section 4.3)	**Users perform property-based workflows**	Contributors, when adding content to the ontology, tend to concentrate their efforts on one single property, which is added and edited for multiple classes.

Contributors in ICD-11 have a high tendency of performing *no property* changes after they return from a *BREAK* followed by *use*, *title* and *definition*. In ICTM, users resume their work primarily by changing the *title* property, the *definition* property followed by *no property* changes.

Interpretation & Practical Implications: One of the main benefits of this analysis is the identification of commonly and consecutively changed properties for classes and users. In turn, this information might potentially be used to suggest work (e.g., prompting a user to check a certain property by combining the *User-Sequence Paths* analysis and the *Property Paths* analysis), or by ontology-engineering tool developers to potentially anticipate the property a user is most likely to change next. The fact that classes appear to exhibit more diverse property-contribution patterns when being changed than users could be a direct result of the multi-lingual nature of ICTM and the already mentioned export functionality present in iCAT. This means that given the most recent property of a class that was edited, we may predict which property is most likely to be changed next. Similarly, we can predict the property a user is going to edit next.

5. Findings and Discussion

In this section we first summarize our findings in Section 5.1 before we shortly discuss the potential applicability of higher order Markov chain models in Section 5.2. Next, we discuss differences between the investigated projects in Section 5.3 and finally, point out potential limitations of this work in Section 5.4.

5.1. Summary of findings

We will now discuss our main findings (Table 2) and explore their consequences.

Emergence of micro-workflows: By investigating whether sequential user-contribution patterns (see Section 4.1) can be identified in five different collaborative ontology-engineering projects, we have shown that users appear to work in micro-workflows, indicating that for all investigated projects, each user contains predictive information about the user, who is going to contribute to a specific class next.

Additionally, however not presented in this paper due to reasons of space, we have also conducted an analysis to determine the change type (e.g., adding a property value, moving a class, replacing a property value etc.) a user is most likely to perform next (as shown in Walk et al. [30] for ICD-11). In this analysis we were able to extract a first-order Markov chain for all datasets presented in this paper, meaning that the last change type that a user performed contains information about the next change type of that user. When combining the information about the user who is most likely to contribute to a class next and the specific change action that this user is most likely to conduct (or the change action that is most likely conducted on a class next), we can create specific tasks for contributors, asking them to perform a certain change on a specific class.

Our results could be used by project managers and ontology-engineering tool developers to identify classes for users and users for classes, helping editors to minimize the necessary efforts for finding and identifying classes to contribute to. Moreover, automatic means of curating and delegating work-tasks to users can be derived by ontology-engineering tool developers, which can help to potentially increase participation as discussed in Kittur and Kraut [31].

User roles can be identified: Across all datasets we were able to identify that a limited number of users have contributed to the majority of all changes. These highly active users are very likely to be *target users* for all other users, meaning that they are very likely to change the same class after another user. Across all five datasets, the roles of these *target users* could be identified by us as moderators or administrators of the corresponding projects performing maintenance tasks, such as gardening (e.g., pruning outdated classes, fixing errors etc.) or manual verification of newly added data.

Furthermore, we were able to show that moderators and administrators divide work among each other, as they are not very likely to change the same classes directly after another administrator or moderator, even though these users exhibit the highest absolute numbers of changes in the corresponding projects. Looking at the transition probabilities of Figure 3 it is possible to identify users or even groups of users who have a high tendency to work on the same classes, thus might be collaborators or reverting/correcting changes of each other.

Users edit the ontology top-down and breadth-first: The *Depth-Level Paths* analysis (see Section 4.2.1) demonstrated that users have a very high tendency of staying in the same depth level when contributing to the ontology. If editors change depth levels while editing the ontology they exhibit a minimal preference to do so in a *top-down* rather than a *bottom-up* manner. Furthermore, the results suggest that users move along the hierarchy as we were able to show that they follow a *top-down* editing strategy for classes that are closer to the root node while this changes to a *bottom-up* editing strategy for classes closer to the deepest depth levels and transitions are more likely to occur along the immediate higher or lower depth level.

To further investigate the distances between changed classes at the same depth levels we investigated the *Hierarchical Relationship Paths* (e.g., child, parent, sibling, cousin etc.) between these changed classes. We found that users, when they edit classes on the same depth level, follow a *breadth-first* manner, focusing on editing all the siblings of a class before switching to a completely different area of the ontology to continue their work after a *BREAK*.

Users edit closely related classes: Additionally to the *breadth-first* manner that users follow when editing classes in the same depth level, we discovered that users have a very high tendency to work on closely related classes (e.g., the sibling or cousin of the currently changed class). The information collected in Section 4.2 allows to potentially predict (or narrow down) the class a user is going to contribute to next, which, if accurate, is a very valuable information that could be used for a variety of improvements and adaptions. For example, project-administrators could adjust the milestones of the development-strategy to better reflect the way users contribute to the ontology while user-interface designers could emphasize certain areas of the ontology to direct users towards specific classes – especially after they return from a *BREAK* – or implement pre-fetching algorithms to minimize load-times. For contributors in particular, the task of identifying and finding classes that they (i) want and (ii) have the necessary expert knowledge to contribute to is a time-consuming task, which potentially can be minimized by implementing class recommender based on the results of the *Structural Paths Analysis* and *User-Sequence Paths Analysis*.

Users perform property-based workflows: The investigation of sequential patterns for property-contributions showed that in ICD-11, users have a very high tendency of consecutively changing the same property across multiple classes. We could also identify specific patterns that emerge when users successively change properties in collaborative ontology-engineering projects.

The results collected in the Section 4.3 provide new insights for administrators and ontology-engineering tool developers, as they allow the generation of work-tasks (e.g., Please verify the property *title* of the class *XII Diseases of the skin*!). So far, users are always presented first with the section of the interface that allows for changing or adding the *title* and *definition*, which could be one explanation for the high probabilities of users changing these properties when returning from a *BREAK*.

Note, that for this analysis we have used the data from ICD-11 and ICTM, which both share a very similar ontology-engineering tool, thus the results might be biased towards the used ontology-editor.

5.2. Higher Order Markov chains

Based on our proposed methodology of using first-order Markov chain models (see Section 3.3) resulting in the findings summarized in Section 5.1, we currently lay our focus on detecting patterns only derived from successive interactions within collaborative ontology-engineering projects. This means, that we identify how likely it is that one specific interaction follows another one (e.g., which user edits a class after another one). This is reasoned by the definition of a first-order Markov chain based on the Markovian property which postulates that the next interaction only depends on the current one.

Contrary, Markov chain models can also be defined on higher orders; this means that the next state of the model (or interaction in our case) depends on a series of preceding ones instead of only the current one. For example, a *second-order* Markov chain model postulates that the next state depends on the current state and also the previous one. Previous studies suggest that human navigation on the Web might be better modeled by using higher order models compared to first-order models (e.g., [32, 29]). Hence, we could assume that this might also be the case for our use-case. By also modeling our data with such higher order models, we would potentially be able to identify longer patterns (e.g., *User A* regularly edits a class after *User B and User C*). Also, possible recommender systems could benefit from the additional predictive power of such higher order chains[12]. While highly interesting, this analyses would be out-of-scope for this article which is why we leave this open for future work.

5.3. Differences between the investigated projects

Even though each project exhibits a different number of depth levels, which all receive a different amount of attention by the contributors, we can observe commonalities of edit strategies between them. For example, the levels 3 to 6 exhibit the highest number of changes in our observation period for ICD-11, while for OPL these levels are 6 and 7.

---

[12] Note that it is necessary to apply model selection techniques as described in [29] in order to identify the most appropriate Markov chain order based on statistical significant improvements of higher orders compared to lower orders

Regarding the hierarchical relationships we can see that consecutively changing the same class is very likely to happen in ICD-11, ICTM, BRO and OPL regardless of the source relationship (evident in the darker colored *Self* columns in Figures 6(a), 6(b), 6(d) and 6(e)). This *Self*-relationship is still very prominent, however the transition probabilities towards *Self* for NCIt are not as dominant as they are for the other datasets.

Another observation depicted in the transition maps is the clear focus on transitions from *Sibling* to *Sibling* across three out of five datasets, with the exception of ICTM and OPL. One explanation for ICTM could be the fact that some properties of the ontology are multi-lingual, thus require users to add multiple languages for the same property, which are all stored as a single change. For OPL, transitions, except towards *Self* are in general really scarce, indicating that users focused on editing and entering multiple property values (or one property value) of a single class before continuing to the next class.

When looking at the sequence of changed properties for each class (in contrast to: for each user) we can observe a concentration on consecutively changing the same property in ICTM, which is most likely a direct result of the multi-lingual nature of the properties used in this project. In ICD-11 on the other hand, transitions between changed properties of classes are much more diverse and less focused on transitions between the same properties. This observation indicates that either not all properties have received a substantial amount of values for all the possible properties and/or that users make use of this special export functionality of iCAT, thus successively changing the same property is less common as the content is only inserted once into the system.

In the *User-Interface Sections Paths* analysis we have mapped the changed properties to the corresponding sections of the user interface of the used ontology-engineering tools, which essentially represents a more abstract analysis of the *Property Paths* analysis. By investigating the sequences of user interface sections we could confirm that, for ICD-11, users have a very high tendency to consecutively change the same properties for multiple classes, evident in the scarce transitions between different sections and the high concentration on transitions between the same sections. For ICTM this behavior was not as distinctive as it was for ICD-11, which could be due to the missing export functionality and therefore the lack of the previously explained manual import sessions.

In general these observations indicate that the absence or presence of a given functionality of the ontology-engineering tool can produce (and influence) different editing behaviors when developing an ontology.

5.4. Limitations

We were not able to recreate the exact class hierarchy of the ontology for every single change across our observation periods for all datasets. This limitation is partly due to a lack of detail in the change-logs. Thus, we decided to focus our analysis, using all five ontologies *as is* at the latest point in time, which is also what would most likely be used in a *real-world* scenario.

For example, if a class was changed by a user while it was located on depth level 3 and at a later point in time moved to a different location where it now resides at depth level 5, we would assume that this class has always been on depth level 5. Please note that this bias is only present in the *Structural Paths* analyses (Section 4.2). To measure the extent of the potential bias, we counted all changes that were performed on a class before it was moved within in the ontology. Applying this rule to our change dataset, we collected a total of $116,204$ of $439,229$ changes for ICD-11 and $18,958$ of $67,522$ for ICTM. These numbers represent about $1/4$ and $1/3$ of all changes for ICD-11 and ICTM respectively. For BRO 276 of 2, 507 (ca. 1/10) and for OPL 2 of 1, 993 of all changes were performed on classes, which have been moved afterwards.

Note that an additional requirement for the identification of sequential patterns in collaborative ontology-engineering projects using Markov chains is the availability of rather large change-logs. In general, the less common entities (e.g., properties) are present in the change-log the more (exponentially) observations have to be available in order to detect more fine-grained patterns. Without enough observations (changes), the identification of sequential patterns is either very hard, and can only be approximated, or not possible at all. As can be seen in Table 1, we have selected all of our datasets to satisfy this requirement, as all chosen datasets exhibit a substantial number of changes.

Furthermore, we have included *artificial session breaks* into our analysis as described by Walk et al. [30] to analyze where or what users start to edit in the ontology and where or what users edit before they take a break. For all user-based analyses we have introduced a *BREAK* if two consecutive changes of the same user were apart longer than 5 minutes.

All analyses in this paper are based on *isKindOf* relationships for determining distances and locations within the ontology. We plan on further expanding this analysis by investigating the impact of other kinds of relationships and other features that are available in ontologies on our pattern detection approach.

Even though all datasets presented in this paper are created with WebProtégé or one of its derivatives, there is only one requirement that prevents practitioners from performing this analysis on other ontologies: The availability of a change-log (in the required granularity for the deemed analyses) that can be mapped onto the underlying ontology. Note that it would be possible to conduct this analysis for ontologies created by single individuals, meaning that “collaboration” is only a requirement when the nature of the analysis requires investigating transitions between multiple users.

Also, the kind of knowledge base (classification, taxonomy or ontology), the used representation language (e.g., OWL and OWL-DL expressivity, RDF, Turtle) or the development tool of a particular collaborative ontology-engineering project in question does not prohibit conducting a pattern analysis as presented in this paper, as long as the underlying knowledge base (and thus the change-log) exhibits the necessary granularity and the semantic properties of interest for the analysis.

However, this also means that the differences of the knowledge representation used languages (i.e., expressivity and types) are not considered by our analysis, with NCIt being a thesaurus and the rest of the investigated datasets being ontologies. Thus, whenever differences are observed between NCIt and the remaining datasets, further research is warranted to determine the origin of this observation.

Furthermore, the analysis presented relies on investigating usage logs of collaborative ontology-engineering projects by looking at changes, performed by users of the corresponding systems. As this only represents one possible way of interacting with the underlying ontology, albeit the most frequently used one, an extension of the conducted Markov chain investigation warrants future work to include, for example, discussions for consensus building, suggestions of terms by users or automatic imports.

6. Related Work

For the analysis and evaluation conducted in this paper, we identified relevant information and publications in the domains of (i) Markov chain models, (ii) collaborative authoring systems and (iii) sequential pattern mining.

6.1. Markov chain models

In the past, Markov chain models have been heavily applied for modeling Web navigation—some sample applications of Markov chains can be found in [33, 34, 35, 36, 37, 38]. Also, the Random Surfer model in Google's PageRank [39] can be seen as a special case of a Markov chain.

Previously, researchers investigated whether human navigation is memoryless (i.e., of first order) in a series of studies (e.g., [40, 36]). However, these studies mostly showed that the memoryless model seems to be a quite plausible abstraction (see e.g., [41, 42, 37, 38]). Recently, a study picked up on these investigations and suggested that the Markovian assumption (i.e., property) might be wrong [32]. However, this study did not reveal any statistically significant improvements of higher order models. Singer et al. [29] solved this problem by developing a framework for determining the appropriate order of a Markov chain for a given set of input data. In Walk et al. [30] we applied and mapped the presented framework onto structured logs of changes and provided an in-depth description of the requirements and steps necessary to use the framework in this setting.

In this paper we present a detailed analysis of sequential patterns by applying and analyzing Markov chains across the change-logs of five collaborative ontology-engineering projects in the biomedical domain. A more detailed explanation of the necessary steps to be able to apply Markov chains onto the change-logs of collaborative ontology-engineering projects is presented in Walk et al. [30]. Note that we focus on applying first-order Markov chain models in this work while we see the application of also higher order models as highly interesting future work as discussed in Section 5.2.

6.2. Collaborative Authoring Systems

Research on collaborative authoring systems such as Wikipedia has in part focused on developing methods and studying factors that improve article quality or increase user participation. These problems represent important facets of collaborative authoring systems and solutions to tackle these problems are of interest for collaborative ontology-engineering projects.

For example, Cabrera and Cabrera [43] demonstrated the effect of minimizing the costs and efforts necessary for users to contribute on potentially achieving higher contribution rates. Another approach, also presented by Cabrera and Cabrera [43], focuses on providing an environment where interactions and communication between contributors are encouraged and performed frequently over a long period of time to establish a group identity and to promote personal responsibility.

More recent research on collaborative authoring systems, such as Wikipedia, focuses on describing and defining not only the act of collaboration amongst strangers and uncertain situations that contribute to a digital good [44] but also on antagonism and sabotage of said systems [45]. It has also been discovered only recently that Wikipedia editors are slowly but steadily declining [46]. Therefore Halfaker et al. [47] have analyzed what impact reverts have on new editors of Wikipedia. Kittur and Kraut [31] showed that an increase in participation can be achieved by directly delegating specific tasks to contributors. As simple as this approach may appear, the identification of work (and thus specific tasks) is still a tedious and time-consuming process, which can only partly be automated due to its assigned complexity.

With the analysis that we described here, we provide new results that we can use to tackle some of the problems for collaborative authoring systems. These problems are also present in collaborative ontology-engineering projects. For example, we can identify new tasks by combining the results of the *User-Sequence Paths* (Section 4.1) and *Property Paths* (Section 4.3) analyses to suggest classes and the corresponding properties to work on to users.

6.3. Sequential Pattern Mining

In 1995 Agrawal and Srikant [48] have first addressed the problem of sequential pattern mining. They stated that given a collection of chronologically ordered sequences, sequential pattern mining is about discovering all sequential patterns weighted according to the number of sequences that contain these patterns. The presented algorithm represents one of the first *a priori* sequential pattern mining algorithms. This means that a specific pattern cannot occur more frequently (above a threshold) if a sub-pattern of this pattern occurs less often (below that threshold). Other examples of a priori algorithms are [49, 50].

One of the biggest problems assigned to the a priori based sequential pattern mining algorithms was (in the worst case) the exponential number of candidate generation. To tackle this problem Han et al. [51] developed the FP-Growth algorithm.

Many researchers have adapted different algorithms and approaches for different domains to anticipate changing requirements, such as Wang and Han [52] and Hsu et al. [53] who analyzed algorithms for sequential pattern mining in the biomedical domain.

In Walk et al. [30] the authors have presented a novel application of Markov chains to mine and determine sequential patterns from the structured logs of changes of collaborative ontology-engineering projects. Making use of this framework we investigate differences and commonalities across five different collaborative ontology-engineering projects from the biomedical domain.

7. Conclusions & Future Work

In this work, we discovered intriguing social and sequential patterns that suggest that large collaborative ontology-engineering projects are governed by a few general principles that determine and drive development. Specifically, our results indicate that patterns can be found in all investigated projects, even though the National Cancer Institute Thesaurus (NCIt), the International Classification of Diseases (ICD-11), the International Classification of Traditional Medicine (ICTM), the Ontology for Parasite Lifecycle (OPL) and the Biomedical Resource Ontology (BRO) (i) represent different projects with different goals, (ii) use variations of the same ontology-editors and tools for the engineering process and (iii) differ in the way the projects are coordinated. Using the presented Markov chain analysis, multiple different user-roles could be identified in all investigated datasets. We were also able to see that users work in micro-workflows, meaning that given a specific user, we can identify the most likely users that are editing a specific class next, again independent from the investigated project. When contributing to a project that is created using WebProtégé, iCAT, iCAT-TM or Collaborative Protégé, users exhibit a tendency to do so in a *top-down* and *breadth-first* manner, editing primarily closely related classes while moving along the ontological hierarchy. In ICD-11 and ICTM we were able to identify property-based workflows, meaning that users concentrate their efforts on adding and editing values for one specific property for multiple classes.

The analysis presented not only provides new insights about the engineering and development processes of each single project, but also shows that the analysis of sequential patterns potentially provides actionable insights for different stakeholders in collaborative ontology-engineering projects.

Furthermore, the information of the next possible action (e.g., a user, a change-type, a property, set of classes) or the combination of multiple of these next actions could be used by ontology-engineering tool developers to potentially augment users in collaboratively creating an ontology. For example, by making use of the *Property Paths* analysis to highlight, prefetch, rearrange or adjust sections and content of the interface dynamically, according to the user's needs.

The next logical step to further deepen our understanding of collaborative ontology-engineering projects involves applying the gathered results to productive and live environments, for example as plug-in for (Web)Protégé. Simultaneously, this would allow us to collect valuable data to quantify the usefulness and actionability of the results, generated with our presented approach, in real world scenarios.

Additionally, expanding the Markov chain analysis to take other types of interactions (e.g., discussions, automatic imports and term suggestions by users) into account, represents a potential topic of future work. This also includes a detailed analysis of human factors studies in terms of user-studies (e.g., with a heuristic evaluation or A/B testing) or more sophisticated approaches, such as eye tracking, to assess the usefulness of the presented results for augmenting users when collaboratively engineering an ontology.

Furthermore, as change tracking and click tracking data will likely become available more broadly in the future, we believe that the analysis of this paper and the possible benefits of putting the results into practical use represent an import step towards the development of better (and simpler) ontology editors, which can dynamically anticipate the editing-style of the users. Project administrators could make use of the results of the analysis, for example by allowing for easier delegation of work to the "right" users. This is even more emphasized when considering that the Markov chain analysis is not computationally intensive, making it highly suitable for productive use.

As biomedical ontologies play an increasingly critical role in acquiring, representing, and processing information about human health, we can use quantitative analysis of editing behavior to generate potentially useful insights for building better tools and infrastructures to support these tasks.

Acknowledgements

This work was generously funded by a Marshall Plan Scholarship with support from Graz University of Technology.

[1] T. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220.

[2] W. Borst, Construction of engineering ontologies for knowledge sharing and reuse (1997).

[3] R. Studer, V. R. Benjamins, D. Fensel, Knowledge engineering: Principles and methods, volume 25, 1998, pp. 161–197.

[4] N. F. Noy, T. Tudorache, Collaborative ontology development on the (semantic) web., in: AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering, AAAI, 2008, pp. 63–68.

[5] T. Groza, T. Tudorache, M. Dumontier, Commentary: State of the art and open challenges in community-driven knowledge curation, Journal of Biomedical Informatics 46 (2013) 1–4. URL: `http://dx.doi.org/10.1016/j.jbi.2012.11.007`. doi:`10.1016/j.jbi.2012.11.007`.

[6] M. Krötzsch, D. Vrandecic, M. Völkel, Semantic MediaWiki, in: Proceedings of the 5th International Semantic Web Conference 2006 (ISWC 2006), Springer, 2006, pp. 935–942.

[7] S. Auer, S. Dietzold, T. Riechert, OntoWiki–A Tool for Social, Semantic Collaboration, in: Proceedings of the 5th International Semantic Web Conference (ISWC 2006), volume LNCS 4273, Springer, Athens, GA, 2006.

[8] C. Ghidini, B. Kump, S. Lindstaedt, N. Mahbub, V. Pammer, M. Rospocher, L. Serafini, MoKi: The Enterprise Modelling Wiki, in: L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren,

M. Sabou, E. P. B. Simperl (Eds.), Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications 2009, Springer, Berlin, Heidelberg, 2009, pp. 831–835.

[9] T. Schandl, A. Blumauer, Poolparty: SKOS thesaurus management utilizing linked data, The Semantic Web: Research and Applications 6089 (2010) 421–425.

[10] T. Tudorache, C. Nyulas, N. F. Noy, M. A. Musen, WebProtégé: A Distributed Ontology Editor and Knowledge Acquisition Tool for the Web, Semantic Web Journal 4 (2013) 89–99.

[11] T. Tudorache, S. M. Falconer, C. I. Nyulas, N. F. Noy, M. A. Musen, Will Semantic Web technologies work for the development of ICD-11?, in: Proceedings of the 9th International Semantic Web Conference (ISWC 2010), ISWC (In-Use), Springer, Shanghai, China, 2010.

[12] J. Pöschko, M. Strohmaier, T. Tudorache, N. F. Noy, M. A. Musen, Pragmatic analysis of crowd-based knowledge production systems with icat analytics: Visualizing changes to the icd-11 ontology, in: Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium: Wisdom of the Crowd, Stanford, CA, USA, 2012.

[13] S. Walk, J. Pöschko, M. Strohmaier, K. Andrews, T. Tudorache, C. Nyulas, M. A. Musen, N. F. Noy, PragmatiX: An Interactive Tool for Visualizing the Creation Process Behind Collaboratively Engineered Ontologies, International Journal on Semantic Web and Information Systems (2013).

[14] S. M. Falconer, T. Tudorache, N. F. Noy, An analysis of collaborative patterns in large-scale ontology development projects., in: M. A. Musen, . Corcho (Eds.), K-CAP, ACM, 2011, pp. 25–32.

[15] C. Pesquita, F. M. Couto, Predicting the extension of biomedical ontologies, PLoS Comput Biol 8 (2012) e1002630. URL: `http://dx.doi.org/10.1371%2Fjournal.pcbi.1002630`. doi:`10.1371/journal.pcbi.1002630`.

[16] R. S. Goncalves, B. Parsia, U. Sattler, Analysing the evolution of the nci thesaurus, in: Proceedings of the 2011 24th International Symposium on Computer-Based Medical Systems, CBMS '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 1–6. URL: `http://dx.doi.org/10.1109/CBMS.2011.5999163`. doi:`10.1109/CBMS.2011.5999163`.

[17] R. S. Gonçalves, B. Parsia, U. Sattler, Facilitating the analysis of ontology differences, in: Proceedings of the Joint Workshop on Knowledge Evolution and Ontology Dynamics (EvoDyn), 2011.

[18] R. S. Gonçalves, B. Parsia, U. Sattler, Categorising logical differences between owl ontologies, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11, ACM, New York, NY, USA, 2011, pp. 1541–1546. URL: `http://doi.acm.org/10.1145/2063576.2063797`. doi:`10.1145/2063576.2063797`.

[19] N. F. Noy, A. Chugh, W. Liu, M. A. Musen, A framework for ontology evolution in collaborative environments, in: The Semantic Web-ISWC 2006, Springer, 2006, pp. 544–558.

[20] B. C. Grau, I. Horrocks, Y. Kazakov, U. Sattler, Just the right amount: extracting modules from ontologies, in: Proceedings of the 16th international conference on World Wide Web, ACM, 2007, pp. 717–726.

[21] B. C. Grau, I. Horrocks, Y. Kazakov, U. Sattler, A logical framework for modularity of ontologies, in: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, pp. 298–303. URL: `http://dl.acm.org/citation.cfm?id=1625275.1625322`.

[22] E. Mikroyannidi, L. Iannone, R. Stevens, A. Rector, Inspecting regularities in ontology design using clustering, in: Proceedings of the 10th International Conference on The Semantic Web - Volume Part I, ISWC'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 438–453. URL: `http://dl.acm.org/citation.cfm?id=2063016.2063045`.

[23] M. Strohmaier, S. Walk, J. Pöschko, D. Lamprecht, T. Tudorache, C. Nyulas, M. A. Musen, N. F. Noy, How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects, Web Semantics: Science, Services and Agents on the World Wide Web 20 (2013). URL: `http://www.websemanticsjournal.org/index.php/ps/article/view/333`.

[24] H. Wang, T. Tudorache, D. Dou, N. F. Noy, M. A. Musen, Analysis of user editing patterns in ontology development projects, in: On the Move to Meaningful Internet Systems: OTM 2013 Conferences, Springer, 2013, pp. 470–487.

[25] S. Staab, R. Studer, Handbook on Ontologies, 2nd ed., Springer Publishing Company, Incorporated, 2009.

[26] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, New York, NY, USA, 2003.

[27] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, L. W. Wright, NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information, Journal of Biomedical Informatics 40 (2007) 30–43.

[28] J. D. Tenenbaum, P. L. Whetzel, K. Anderson, C. D. Borromeo, I. D. Dinov, D. Gabriel, B. A. Kirschner, B. Mirel, T. D. Morris, N. F. Noy, C. Nyulas, D. Rubenson, P. R. Saxman, H. Singh, N. Whelan, Z. Wright, B. D. Athey, M. J. Becich, G. S. Ginsburg, M. A. Musen, K. A. Smith, A. F. Tarantal, D. L. Rubin, P. Lyster, The Biomedical Resource Ontology (BRO) to enable resource discovery in clinical and translational research, Journal of Biomedical Informatics 44 (2011) 137–145.

[29] P. Singer, D. Helic, B. Taraghi, M. Strohmaier, Memory and structure in human navigation patterns, arXiv preprint

arXiv:1402.0790 (2014).

[30] S. Walk, P. Singer, M. Strohmaier, D. Helic, N. F. Noy, M. A. Musen, Sequential usage patterns in collaborative ontology-engineering projects, arXiv preprint arXiv:1403.1070 (2014).

[31] A. Kittur, R. E. Kraut, Harnessing the wisdom of crowds in wikipedia: quality through coordination, in: Proceedings of the 2008 ACM conference on Computer supported cooperative work, CSCW '08, ACM, New York, NY, USA, 2008, pp. 37–46.

[32] F. Chierichetti, R. Kumar, P. Raghavan, T. Sarlos, Are web users really markovian?, in: Proceedings of the 21st international conference on World Wide Web, WWW '12, ACM, New York, NY, USA, 2012, pp. 609–618. URL: `http://doi.acm.org/10.1145/2187836.2187919`. doi:`10.1145/2187836.2187919`.

[33] J. Borges, M. Levene, Evaluating variable-length markov chain models for analysis of user web navigation sessions, IEEE Trans. on Knowl. and Data Eng. 19 (2007) 441–452. URL: `http://dx.doi.org/10.1109/TKDE.2007.1012`. doi:`10.1109/TKDE.2007.1012`.

[34] M. Deshpande, G. Karypis, Selective markov models for predicting web page accesses, ACM Trans. Internet Technol. 4 (2004) 163–184. URL: `http://doi.acm.org/10.1145/990301.990304`. doi:`10.1145/990301.990304`.

[35] R. Lempel, S. Moran, The stochastic approach for link-structure analysis (salsa) and the tkc effect, Comput. Netw. 33 (2000) 387–401. URL: `http://dx.doi.org/10.1016/S1389-1286(00)00034-7`. doi:`10.1016/S1389-1286(00)00034-7`.

[36] P. L. T. Pirolli, J. E. Pitkow, Distributions of surfers' paths through the world wide web: Empirical characterizations, World Wide Web 2 (1999) 29–45. URL: `http://dx.doi.org/10.1023/A:1019288403823`. doi:`10.1023/A:1019288403823`.

[37] R. Sen, M. Hansen, Predicting a web user's next access based on log data, Journal of Computational Graphics and Statistics 12 (2003) 143–155. URL: `http://citeseer.ist.psu.edu/sen03predicting.html`.

[38] I. Zukerman, D. W. Albrecht, A. E. Nicholson, Predicting users' requests on the www, Proceedings of the Seventh International Conference on User Modeling, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999, pp. 275–284. URL: `http://dl.acm.org/citation.cfm?id=317328.317370`.

[39] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in: Proceedings of the seventh international conference on World Wide Web 7, WWW7, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 1998, pp. 107–117.

[40] J. Borges, M. Levene, Data mining of user navigation patterns, in: Revised Papers from the International Workshop on Web Usage Analysis and User Profiling, WEBKDD '99, Springer-Verlag, London, UK, UK, 2000, pp. 92–111. URL: `http://dl.acm.org/citation.cfm?id=648036.744399`.

[41] I. Cadez, D. Heckerman, C. Meek, P. Smyth, S. White, Model-based clustering and visualization of navigation patterns on a web site, Data Min. Knowl. Discov. 7 (2003) 399–424. URL: `http://dx.doi.org/10.1023/A:1024992613384`. doi:`10.1023/A:1024992613384`.

[42] R. R. Sarukkai, Link prediction and path analysis using markov chains, Proceedings of the 9th international World Wide Web conference on Computer networks: the international journal of computer and telecommunications netowrking, North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2000, pp. 377–386. URL: `http://dl.acm.org/citation.cfm?id=347319.346322`.

[43] A. Cabrera, E. F. Cabrera, Knowledge-Sharing Dilemmas, Organization Studies 23 (2002) 687–710.

[44] B. Keegan, D. Gergle, N. S. Contractor, Hot off the wiki: dynamics, practices, and structures in Wikipedia's coverage of the Tohoku catastrophes., in: F. Ortega, A. Forte (Eds.), Int. Sym. Wikis, ACM, 2011, pp. 105–113.

[45] N. Shachaf, Beyond vandalism: Wikipedia trolls., Journal of Information Science; Jun2010, Vol. 36 Issue 3, p357-370, 14p, 2 Charts (2010).

[46] B. Suh, G. Convertino, E. H. Chi, P. Pirolli, The singularity is not near: slowing growth of wikipedia, in: WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration, ACM, New York, NY, USA, 2009, pp. 1–10.

[47] A. Halfaker, A. Kittur, J. Riedl, Don't bite the newbies: how reverts affect the quantity and quality of Wikipedia work., in: F. Ortega, A. Forte (Eds.), Int. Sym. Wikis, ACM, 2011, pp. 163–172.

[48] R. Agrawal, R. Srikant, Mining sequential patterns, in: Proceedings of the Eleventh International Conference on Data Engineering, ICDE '95, IEEE Computer Society, Washington, DC, USA, 1995, pp. 3–14. URL: `http://dl.acm.org/citation.cfm?id=645480.655281`.

[49] R. T. Ng, L. V. S. Lakshmanan, J. Han, A. Pang, Exploratory mining and pruning optimizations of constrained associations rules, in: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD '98, ACM, New York, NY, USA, 1998, pp. 13–24. URL: `http://doi.acm.org/10.1145/276304.276307`. doi:`10.1145/276304.276307`.

[50] S. Sarawagi, S. Thomas, R. Agrawal, Integrating association rule mining with relational database systems: Alternatives and implications, in: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, SIGMOD '98, ACM, New York, NY, USA, 1998, pp. 343–354. URL:

`http://doi.acm.org/10.1145/276304.276335`. doi:`10.1145/276304.276335`.

[51] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, ACM, New York, NY, USA, 2000, pp. 1–12. URL: `http://doi.acm.org/10.1145/342009.335372`. doi:`10.1145/342009.335372`.

[52] J. Wang, J. Han, Bide: Efficient mining of frequent closed sequences, in: Proceedings of the 20th International Conference on Data Engineering, ICDE '04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 79–. URL: `http://dl.acm.org/citation.cfm?id=977401.978142`.

[53] C.-M. Hsu, C.-Y. Chen, B.-J. Liu, C.-C. Huang, M.-H. Laio, C.-C. Lin, T.-L. Wu, Identification of hot regions in protein-protein interactions by sequential pattern mining, BMC bioinformatics 8 (2007) S8.

3.4 Sequential Action Patterns in Collaborative Ontology-Engineering Projects: A Case-Study in the Biomedical Domain

This article tackles the second research question. In particular, it studies if and to what extent actions of edit sequences in collaborative ontology-engineering projects can be predicted using Markov chain models of varying orders.

First, this article provides a detailed analysis regarding the existence of regularities and sequential patterns in the change-logs of five different collaborative ontology-engineering projects from the biomedical domain. To that end, I first applied PrefixSpan, a well-established pattern mining algorithm, on the logs of changes of a large-scale collaborative ontology-engineering project to identify frequent edit patterns of larger size and support.

For modeling and predicting future edit actions of contributors, the Markov chain framework, which was presented in Section 3.2, was applied on the structured logs of changes of five collaborative ontology-engineering projects. In contrast to the analyses conducted in Walk et al. [2014b], colleagues and I were now interested in achieving the highest accuracy for predicting the edit action a user is most likely to conduct next. For several of the prediction experiments, analyses and datasets, higher-order Markov chain models performed better than first-order Markov chain models.

In practice, these predictive models represent a very valuable resource for practitioners to generate, for example, recommender systems or to assess the impact of potential changes to the underlying ontology. In the context of this thesis, the article further expands on the analysis of sequential edit patterns in the change-logs of collaborative ontology-engineering projects in the biomedical domain and leverages these models for predicting different aspects of future actions of contributors.

Sequential Action Patterns in Collaborative Ontology-Engineering Projects: A Case-Study in the Biomedical Domain

Simon Walk*
Graz University of Technology
Graz, Austria
simon.walk@tugraz.at

Philipp Singer*
GESIS - Leibniz Institute for the Social Sciences
Cologne, Germany
philipp.singer@gesis.org

Markus Strohmaier
GESIS & Univ. of Koblenz
Cologne & Koblenz, Germany
markus.strohmaier@gesis.org

ABSTRACT

Within the last few years the importance of collaborative ontology-engineering projects, especially in the biomedical domain, has drastically increased. This recent trend is a direct consequence of the growing complexity of these structured data representations, which no single individual is able to handle anymore. For example, the World Health Organization is currently actively developing the next revision of the International Classification of Diseases (ICD), using an OWL-based core for data representation and Web 2.0 technologies to augment collaboration. This new revision of ICD consists of roughly $50,000$ diseases and causes of death and is used in many countries around the world to encode patient history, to compile health-related statistics and spendings. Hence, it is crucial for practitioners to better understand and steer the underlying processes of how users collaboratively edit an ontology. Particularly, generating predictive models is a pressing issue as these models may be leveraged for generating recommendations in collaborative ontology-engineering projects and to determine the implications of potential actions on the ontology and community. In this paper we approach this task by (i) *exploring* whether regularities and common patterns in user action sequences, derived from change-logs of five different collaborative ontology-engineering projects from the biomedical domain, exist. Based on this information we (ii) *model* the data using Markov chains of varying order, which are then used to (iii) *predict* user actions in the sequences at hand.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Medical information systems; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Web-based interaction*

Keywords

Markov Chain; Sequential Pattern; State Prediction; Collaborative Ontology-Engineering

1. INTRODUCTION

The complexity of structured knowledge representations, especially in the biomedical domain, has dramatically increased over the last decade. This recent trend is the direct result of the increasing requirements for these ontologies to satisfy, due to a growing field of application. For example, the International Classification of Diseases in its 10^{th} revision (ICD-10) is used to encode patient history data and to compile health-related spending and morbidity as well as mortality statistics for international comparison. To increase the utility of ICD, the World Health Organization (WHO) is currently developing the 11^{th} revision of this classification (ICD-11), using the Internet and Web 2.0 technologies as collaboration platform and an OWL-based core for knowledge representation. This change in knowledge representation will allow for additional information to be stored inside ICD-11. For example, diseases will have (among others) explicitly defined related/affected body parts and diagnostic criteria. Compared to ICD-10, the new revision now contains around $50,000$ diseases and causes of death, thus has roughly tripled in size and is to be developed until 2017.

Due to this increase in complexity, ontologies, such as ICD-11, can no longer be developed by single authorities. Instead, WHO decided to open-up the development process of ICD-11, allowing everyone with access to the Internet to contribute and discuss changes made to the ontology. However, this open and collaborative ontology-engineering process poses many, yet unidentified, problems to tackle and anticipate. For instance, tracking and monitoring user actions or the overall progress of the underlying ontology as well as helping users to identify work tasks, which they have the required expertise to contribute to, are two either computationally expensive or very time consuming tasks. In particular, administrators of collaborative ontology-engineering projects are in need of better tools to understand and augment users when contributing to these projects.

Objective. Our main objective is to predict user actions in collaborative ontology-engineering projects; e.g., the property a user is most likely to edit next. We want to achieve this task by first exploring whether regularities and sequential patterns exist, then building upon these observations for modeling the data and finally, evaluating the prediction accuracy of each model.

Approach. Specifically, we will approach this objective as follows in subsequent order:
(i) Exploring action sequences: First, we investigate whether action sequences based on several dimensions (e.g., sequential properties changed by users as illustrated in Figure 1) exhibit regularities or are emerging in random fashion before we mine and study common sequential patterns in our data.

---

[*] Both authors contributed equally to this work.

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2662049 .

(ii) Modeling action sequences: Next, we establish our model approach using Markov chains of varying order, allowing us to incorporate our insights from the first research approach. We also present model selection techniques that can be used for testing and evaluating the accuracy of these models.
(iii) Predicting user actions: Subsequently, we fit these models to our data and evaluate each model, giving insights into their predictive power. The models may be leveraged for generating recommendations in collaborative ontology-engineering projects and to determine the implications of potential actions on the ontology and community.

We perform our experiments on five datasets stemming from different biomedical projects (ICD-11, The International Classification of Traditional Medicine (ICTM), The National Cancer Institute Thesaurus (NCIt), The Biomedical Resource Ontology (BRO) and The Ontology of Parasite Lifecycle (OPL); for more details see Section 2).

Contributions. To the best of our knowledge, this paper presents the most detailed analysis of sequential user actions in collaborative ontology-engineering projects in the biomedical domain for predicting future actions. We find (significant) evidence that (i) regularities and (long) sequential patterns do exist and (ii) demonstrate their utility for predicting the action that is most likely to occur next in our datasets.

Our insights not only improve our understanding of how users engage in collaborative ontology-engineering projects but can also potentially improve the workflow of collaborators by, e.g., recommending properties to contributors to edit next. By doing so, we may be able to better leverage the expertise of contributors by steering them into the right direction. Apart from that, practitioners may also be able to enhance the quality of specific parts of the ontology by promoting them to the right users. Having predictive models for user actions will also allow collaborative ontology-engineering project administrators to assess potential actions regarding their implications on the underlying ontology and community.

Structure of this article. We introduce our experimental setup in Section 2 before we explore action sequences in Section 3. We introduce our model approach in Section 4 and apply and evaluate these models in Section 5. We discuss (Section 6) our findings and related work (Section 7) next and conclude our work in Section 8.

2. EXPERIMENTAL SETUP

In this section we first briefly introduce our five datasets, stemming from the biomedical domain, before we elaborate on our specific dataset preparation steps.

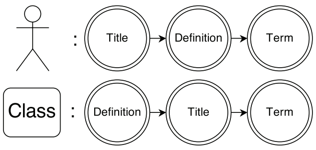


Figure 1: The top row of the figure depicts an exemplary user-based property sequence with properties *Title*, *Definition* and *Term* for a user. This means that the first property that was changed by the user is *Title*, then *Definition* and last *Term*. The bottom row of the figure shows the class-based sequential property path for a class and the same properties *Title*, *Definition* and *Term*. Analogously, the first property that was changed for the class was *Definition*, then *Title* and last *Term*.

2.1 Dataset Description

Table 1 lists the detailed features and observation periods for all datasets used in our analysis. The two largest datasets are ICD-11[1] and the National Cancer Institute Thesaurus (NCIt) [28] with 48,771 and 102,865 classes and 439,299 and 294,471 changes respectively. NCIt is a reference vocabulary for clinical care, translational, basic research and cancer biology. The International Classification of Traditional Medicine (ICTM), which was first intended to be a stand-alone biomedical ontology but was merged with ICD-11 after our observation period, represents a collaborative ontology-engineering project of medium size, with 1,506 classes and a total of 67,522 changes. ICTM is developed by WHO and tries to unify knowledge from traditional medicine practices from China, Japan and Korea. The Biomedical Resource Ontology (BRO) and the Ontology for Parasite Lifecycle (OPL) are two smaller sized collaborative ontology-engineering projects with only 528 and 393 classes and 2,507 and 1,993 changes respectively. BRO is a controlled terminology for describing the source type, areas of research, and activity of biomedical related resources. OPL models the life cycle of a parasite, which is responsible for a number of human diseases.

2.2 Dataset preparation

We extracted sequences from activity logs of the five collaborative ontology-engineering datasets to perform our experiments on. All extracted sequences are either *class- or user-based* (see Figure 1). A class-based sequence depicts a chronology of a specific feature of all changes that were performed *by any user on a single class*. A user-based sequence, analogously, captures the ordered list

---

[1] `http://www.who.int/classifications/icd/ICDRevision/`

Table 1: Characteristics of the investigated datasets. Note that all datasets differ in size (number of classes and users), activity (number of changes) and observation periods. ICD-11 and ICTM both exhibit changes that were performed automatically and are denoted as *# of bots (changes)* in the table. For our analysis we removed these changes.

		ICD-11	ICTM	NCIt	BRO	OPL
Ontology	# of classes	48,771	1,506	102,865	528	393
	# of changes	439,229	67,522	294,471	2,507	1,993
Users	# of users	109	27	17	5	3
	# of bots (changes)	1 (935)	1 (1)	0 (0)	0 (0)	0 (0)
Duration	first change	18.11.2009	02.02.2011	01.06.2010	12.02.2010	09.06.2011
	last change	29.08.2013	17.7.2013	19.08.2013	06.03.2010	23.09.2011
	observation period (ca.)	4 years	2.5 years	3 years	1 month	3 months

of specific features of changes that were performed *on any class by a single user* for each dataset. Note that we are interested in studying collaborative behavior in this paper and hence, provide an aggregated view on the data based on all users or all classes. Thus, we always work with a set of distinct sequences where each sequence corresponds to one single user (user-based) or one single class (class-based). In a preprocessing step, we pruned all sequences that exhibit less than two elements, for example, if a class was only ever changed by one user, we removed this specific entry from our training set. Note that we have removed all automatic changes performed in ICD-11 and ICTM for our analyses (see Table 1). In Sections 3 and 4, we will closely investigate the following aspects (and thus sequences) of the activity logs:

(i) Users for Classes. These, solely class-based, sequences consist of chronologically ordered lists, where each list captures one class, of users that changed a specific class.

(ii) Change-Types for Classes and Users. Such a sequence contains a chronology of change-types of the performed changes by a specific user on any class (user-based) or the change-types of the performed changes for a specific class by any user (class-based). We aggregated the performed change-types into abstract classes of changes, which was necessary due to the large variety of different change-types present in our investigated datasets. All changes that edit the value of a property of a class have been aggregated (i.e., added property, edited property, deleted property). Analogously, we have aggregated the changes performed on classes (i.e., added class, moved class, removed class, deleted class).

(iii) Properties for Classes and Users. These sequences consist of chronologically ordered lists of properties changed by a specific user of any class (user-based) or the properties changed for a specific class by any user (class-based).

Note that we were not able to conduct the *Change-Types for Classes and Users* and *Properties for Classes and Users* analyses for NCIt. The reason for this is the existence of a specific feature in the ontology-editor that is used to develop NCIt, which allows contributors to queue changes and commit batches of changes simultaneously to the ontology.

3. EXPLORING ACTION SEQUENCES

In this section we explore the nature of our action sequences at hand. We first investigate randomness and regularities in Section 3.1 and then continue to extract common sequential patterns in Section 3.2.

3.1 Randomness and Regularities

To begin with, we are interested in determining whether our data sequences are produced in random fashion or based on some regularities. One common way to investigate randomness in such sequences or time series is to use *autocorrelation* with varying lags [6]. This method builds on Pearson's product-moment correlation coefficient which determines linear relationships between lagged variables. Contrary, in our paper, we work with categorical data in our sequences (e.g., properties) which is why the autocorrelation method is not directly applicable to our problem at hand.

Another way of determining randomness in data sequences is the so-called *runs test* which is also more specifically entitled *Wald-Wolfowitz runs test* [35, 7]. It is a non-parametric test in which the null hypothesis (the sequence was produced randomly; the elements of the sequence are independent to each other) is tested against the alternative hypothesis stating that the sequence was not produced randomly. In particular, the null hypothesis gets rejected if the total number of runs – a run is a series of identical values (e.g., the sequence "AABA" has three runs "AA", "B" and "A") – is too small leading to a clustered arrangement or too large resulting in a systematic arrangement [21]. Predominantly, the test is only suited for sequences with binary or dichotomous observations. O'Brien and Dyck [21] adapted the initial method by proposing a test that is based on a linear combination of the weighted variances of run lengths. This approach can now be extended to also work with categorical observations which is required for our analyses.[2] We exemplarily applied this method on our individual ICD-11 sequences, and can clearly see that a significant proportion of sequences is produced in a non-random way. This is imminent as the null hypotheses regularly gets rejected (p-value below 0.05) – e.g., the null hypotheses gets rejected for more than 60% of all user property sequences. Our observations in this section warrant further investigations of patterns and structural properties in these sequences. Hence, we next focus on investigating how these present regularities in our sequential patterns look like; i.e., we focus on mining common sequential patterns.

3.2 Sequential Pattern Mining

Given our observations made in Section 3.1, we are now interested in actual sequential patterns that account for the regularities in the activity logs. There do exist a variety of algorithms to extract the most frequently used sequential patterns from a set of sequences. We make use of PrefixSpan [22] to investigate commonly used sequential patterns in collaborative ontology-engineering project change-logs, as the algorithm concentrates on expanding (or growing) frequently used patterns and strictly matches only patterns to sequences that are completely identical (i.e., do not exhibit gaps or skipped elements). Support for sequential pattern mining algorithms, a measure to determine how frequent certain patterns are observed in the data, is usually defined as the percentage of all investigated paths that contain a given pattern. Note that all paths have to be chronologically sorted and patterns only consist of succeeding states. For example, the pattern "AB" is *not* present in the sequence "ACBA", as "B" never immediately succeeds "A".

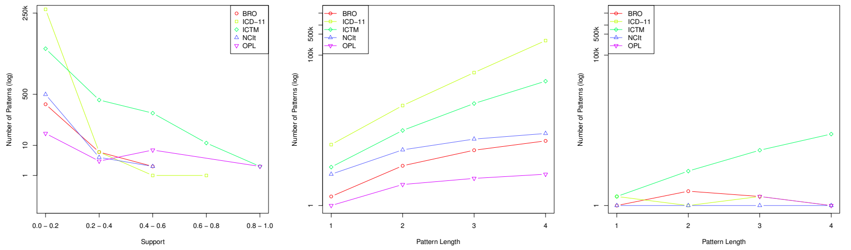
PrefixSpan first scans all available sequences and denotes the number of occurrences for each element in all sequences. It then stores the occurrences and the remainder of the sequences (the suffix) and uses the most frequently used sequential patterns as prefix requirement for the next iteration. Analogously, the prefix is again expanded until a certain level (minimum) support is reached.

We have applied PrefixSpan on the five collaborative ontology-engineering project datasets to see if and to what extent such sequential patterns are present. As can be seen in Figure 2(a), PrefixSpan was able to extract between 5 to 500 patterns for the *Predicting Users for Classes* analysis across all five datasets with a support of 0.2 to 0.4. This means that the identified sequential patterns are present in 20 to 40 percent of all investigated sequences. Figures 2(b) and 2(c) show the number of identified patterns of lengths 1 to 4 for support levels of 0.0 to 0.2 and 0.2 to 0.4. Similar observations could be made for the other analyses.

Given the high number of sequential patterns of lengths 2 to 4 we argue that such patterns play a crucial role in the contributor logs of collaborative ontology-engineering projects at hand. Hence, we believe that there might be some dependence between subsequent

---

[2] We make an implementation of this method available online at `https://github.com/psinger/RunsTest`. Note though that the method has some limitations. For example, there have to be more than one distinct run length for an element, more than one success run and the number of successes minus the number of success runs of an element has to exceed one. For more details please refer to [21] and the source on github. Hence, we only recommend to perform the test on "somewhat" longer sequences with more runs which is the case for our data at hand.



(a) Patterns per Support

(b) Number of Patterns with length n for Support 0.0 - 0.2

(c) Number of Patterns with length n for Support 0.2 - 0.4

Figure 2: Results of the PrefixSpan analysis on the *Predicting Users for Classes Sequences*: Figure 2(a) shows the number of extracted patterns (y-axis; log-scale) by PrefixSpan for a given support range (x-axis). Support is defined as the percentage of paths that exhibit a certain pattern. For example, the roughly 500 sequential patterns extracted for ICTM with a support level of 0.2 - 0.4 are all present in 20 to 40 percent of all analyzed sequences. Furthermore, Figures 2(b) and 2(c) depict the length (x-axis) and number (y-axis; log-scale) of patterns found for each dataset for support levels 0.0 - 0.2 and 0.2 - 0.4.

elements in a sequence – i.e., memory effects might be in play (see also Rosvall et al. [25] for a discussion surrounding memory in networks). Consequently, we want to incorporate these potential memory effects into our model approach in the next section, in which we resort to Markov chain models of varying order. The goal is to find a model that can describe action sequences and predict user actions in a sound way.

4. MODELING ACTION SEQUENCES

As our main goal of this work is to predict user actions in collaborative ontology-engineering projects, we need to find an appropriate model that we can fit to the data and leverage for prediction. Our choice falls on Markov chain models which are suitable for modeling categorical sequences. Specific variations of model parameters allow us to incorporate our findings of Section 3; i.e., that regularities and specifically, serial dependence seems to play a role in the action sequences at hand. Consequently, we first give a brief introduction into Markov chain models in Section 4.1 also elaborating a way to incorporate our observations about regularities and patterns in the action sequences. Finally, we will explain two model selection techniques in Section 4.2, which is crucial for deciding between different models, which will help us to evaluate the performance of our models. We then apply the methods established in this section in Section 5.

4.1 Markov Chains

A Markov chain is a stochastic process that models transitions from one state to another based on a given state space S. It usually is referred to as *memoryless* which constitutes the so-called *Markov property* stating that the next state only depends on the current state and not on a series of preceding ones. We now briefly provide an introduction to Markov chains; we point the interested reader to a more thorough introduction in previous work [27, 37].

For such a *first-order* Markov chain[3] – a sequence of random variables $X_1, X_2, ..., X_n$ – the following holds:

$$P(X_{n+1} = x_{n+1} | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = P(X_{n+1} = x_{n+1} | X_n = x_n) \quad (1)$$

Motivated by our observations in Section 3, where we could see that at least some sequences are arranged in a non-random way – i.e., dependence between elements in a sequence – as well as where we could identify longer sequential patterns to be present in our sequences, we are now also interested in extending this notion of memorylessness of Markov chains to also include memory effects. This means, that we not only want to model the next state as being dependent on the current state, but also on a sequence of preceding states (memory effect). Hence, we now also look at Markov chain models of order k where the future depends on the past k states. We can define a Markov chain model of order k as a process that satisfies:

$$P(X_{n+1} = x_{n+1} | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = P(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, ..., X_{n-k+1} = x_{n-k+1}) \quad (2)$$

Such higher order chains can be modified to a first-order Markov chain by using a state space of compound states of size k[4]; i.e., the state state includes all sequences of length k which finally leads to a set of size $|S|^k|S|$ (see [27] for details). Additionally, we also introduce a so-called *zero-order* Markov chain model where $k = 0$. In such a model the next state does not depend on any other one but we can see this as a *weighted random selection* that should serve as a baseline for our Markov chain models of varying order.

A Markov chain model is represented by a stochastic transition matrix P if the state space is finite (which it is in our case). This matrix contains the transition probabilities of a state x_i to another state x_j for all possible combinations; the probabilities of each row sum to one. The elements of this matrix represent the parameters

---

[3] For our chains we assume *time-homogeneity*, i.e., the probability of transitions is independent of n.

[4] We prepend k reset states and append one reset state to each sequence so that we "forget" the history of other sequences in the dataset [9].

θ that we have to determine. For doing so we resort to Bayesian inference (see [30, 27] for details). We use a Laplace prior for the inference process – i.e., we set each $\alpha_{ij} = 1$.

4.2 Markov Chain Model Selection

As we are interested in modeling memory in the process, we model the data with a set of models with varying orders k and consequently, have to evaluate the performance of each model leading to a determination of the most appropriate order out of this set. We need to note that lower order models are always nested within higher order ones by definition and hence, higher order models will always fit at least as good as lower order ones. Nonetheless, such higher order Markov chain models need exponentially more parameters and thus may result in severe overfitting.

First, we apply Bayesian model selection [30, 27] giving us a tool to decide between an array of models. The benefit of this method is that it naturally includes a *Occam's razor*, which means that higher order models receive a penalty due too much higher complexity, which can help us to avoid overfitting and give us insights into significance [17].

As a second method for evaluating varying order Markov chain models we use a stratified[5] k-fold cross-fold validation[6]. Following the concepts of Singer et al. [27] and Walk et al. [37] we train the Markov chain models on each training set and validate the predictive power on the test set. First, we rank the probabilities of each row in the transition matrix – which are the expectations of the Bayesian posterior – using *modified competition ranking* that includes a natural *Occam's razor* for higher orders. Next, we determine the rank of each transition of the test set – i.e., from each *start state* to each *target state* – and henceforth, average over all transitions in the test set. Finally, we average over all folds and visualize the results. Note that the best accuracy to be achieved would be one as this would mean that each transition in the test set would be the highest probability of the transition matrix learned from the training set. This method also directly gives us a prediction accuracy of each model that can provide us with insights into the general prediction performance of a model.

5. PREDICTING USER ACTIONS

In this section we present results for fitting and evaluating (via prediction) the Markov chain models of varying order for all conducted analyses (see Section 4.2). We were not able to conduct all analyses for NCIt, as the ontology editor used for developing NCIt exhibits some special functionality, which makes it impossible to extract chronologically ordered change-types and properties (cf. Section 2).

5.1 Predicting Users for Classes

The Bayesian model selections (see Table 2) mostly suggest first- or second-order Markov chain models to be appropriate fits for the underlying data. Only for NCIt a higher order – i.e., a fifth-order – is suggested. In order to study the predictive power of these varying order Markov chain models, we conducted a stratified 3-fold cross-fold validation task (see Figure 3(a) and Table 2) which mostly agrees with our Bayesian model selection results in terms of order appropriateness. This means, that a first- (ICD-11, ICTM and BRO) or second-order (NCIt and OPL) model are shown to have the best predictive power throughout all datasets (accounting for overfitting).

The results indicate that the next event in a sequence seems to be dependent on at least the previous one; partly, also on a sequence of previous states (memory effects). Such Markov chain models (of first or second order) can be used for predicting the next contributor for a class while simultaneously compensating for overfitting. An average position of mostly below two can be achieved with the corresponding best working model.

This tells us that we have a well-working tool for predicting the user that is most likely changing a class next. We may leverage this for recommending classes to users which are eligible for change. By doing so we may manage to severely improve the workflow of users as they may not need to tap into their own intuitions about which class to change next. Also, this process could improve the quality of some classes by automatically finding experts who should edit the class.

5.2 Predicting Change Types for Users

The Bayesian model selection (see Table 2) suggests a fourth-order Markov chain model for ICD-11 and ICTM, a second-order model for BRO and a first-order model for OPL. Subsequently, we conducted a 3-fold stratified cross-fold validation for ICD-11 and ICTM and a 2-fold stratified cross-fold validation for OPL and BRO, due to the smaller number of users available in the latter two datasets (see Figure 3(b) and Table 2). The results suggest that a third-order Markov chain model performed best for predicting the change-type a user is going to perform next for ICD-11. For ICTM and OPL a second-order yielded the best prediction results, while a first-order Markov chain model performed best for BRO. The

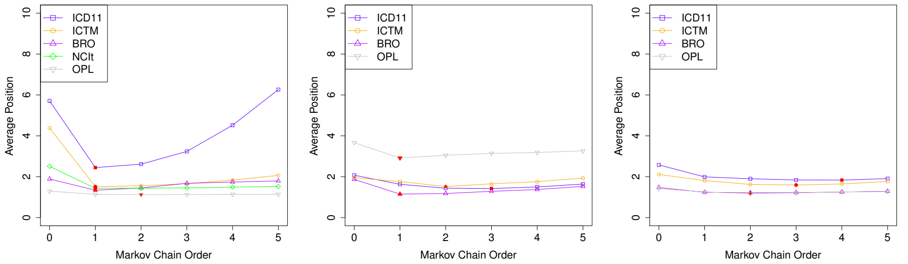
Table 2: The results for all datasets and all analyses conducted in Section 5. Rows marked with *CV* indicate the order of the best-performing Markov chain models of our stratified cross-fold validation task (Section 4.2). Rows marked with *Bayes* depict the order of the Markov chain models determined by the Bayesian model selection task (Section 4.2).

		ICD-11	ICTM	NCIt	BRO	OPL
Predicting Users for Classes (**Section 5.1**)	Bayes	2	1	5	2	2
	CV	1	1	2	1	2
Predicting Change Types for Users (**Section 5.2**)	Bayes	4	4	-	2	1
	CV	3	2	-	1	2
Predicting Change Types for Classes (**Section 5.3**)	Bayes	4	3	-	2	2
	CV	4	3	-	2	2
Predicting Properties for Users (**Section 5.4**)	Bayes	2	1	-	3	4
	CV	1	1	-	1	0
Predicting Properties for Classes (**Section 5.5**)	Bayes	2	1	-	3	5
	CV	1	1	-	3	5

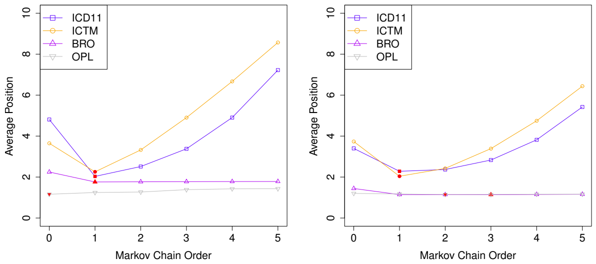
---

[5] Stratified refers to the fact that we try to keep the number of observations equal in each fold.

[6] Note that the number of folds is determined individually for each evaluation due to their stratified nature.



(a) Users for Classes Cross-Fold Validation (b) Change-Types for Users Cross-Fold Validation (c) Change-Types for Classes Cross-Fold Validation



(d) Properties for Users Cross-Fold Validation (e) Properties for Classes Cross-Fold Validation

Figure 3: Results for the *Stratified Cross-Fold Validation* analysis: The plots depict the results of the stratified cross-fold validation for all five datasets for the conducted analyses. The filled elements represent the Markov chain model for each dataset, which achieved the best (lowest) average accuracy (position) score in the prediction tasks. The position score is calculated by determining the position of the next most likely state to occur in a test path given k previous states, where k represents the investigated Markov chain order. Probabilities to select the next most likely state are created using the training set to calculate the transition maps for all datasets and Markov chain orders. The figures show that we can model activity sequences for all of our analyses as first- or higher-order Markov chain models perform best in our prediction task for all datasets, with the only exception of OPL for the *Predicting Properties for Users* analysis (see Figure 3(d)).

cross-fold prediction task also yielded an average accuracy (position) between roughly 1.8 and 3.5.

This indicates that higher-order Markov chains can be used for predicting the change-type a user is most likely to perform next. Practitioners may use this information for recommending change types users should edit next. By doing so we may help to improve the overall progress and quality of the ontology; e.g., if we know that several areas of the ontology or classes lack certain changes, we can steer contributors, which exhibit a preference to perform these kinds of changes, into a specific direction and enforce their contributions in certain branches of the underlying knowledge representation.

5.3 Predicting Change Types for Classes

As depicted in Table 2 the Bayesian model selection suggests a second-order Markov chain model for BRO and OPL, while a third-order model for ICTM and a fourth-order Markov chain model for ICD-11 work best. A stratified 3-fold cross-fold validation (see Figure 3(c) and Table 2) completely agrees with these results for all datasets. The best fitting Markov chain models allow for an average prediction accuracy (position) between 1.8 and 2.0.

The presented results indicate that we can predict the change-type that is most likely conducted on a class next, given at least the two most recent changes on said class as input for our trained Markov chain models. Similar to predicting change types for users, practitioners can use this information for recommending change types that may be useful to change next on a given class. For example, if a class is most likely to receive a certain change type next, we can combine this information with the change types for users and identify a suitable contributor to recommend this class for editing.

5.4 Predicting Properties for Users

The Bayesian model selection yields a second- and first-order Markov chain model for ICD-11 and ICTM and a third- and fourth-order model for BRO and OPL (see Table 2). The conducted 3-fold stratified cross-fold validation, to predict the property a specific user is most likely to change next, yielded a first-order Markov chain model for ICD-11 and ICTM (see Figure 3(d) and Table 2). Due to a limited number of users, a stratified 2-fold cross-fold validation was conducted for BRO and OPL, which showed that a first- and zero-order Markov chain model performs best for predicting the next property for a given user respectively. This means that there was no difference between the Markov chain models trained for OPL and randomly (weighted) choosing (zero-order) the property a user is most likely to change next.

This also means, that for ICD-11, ICTM and BRO we were able to show that subsequent properties users change are dependent on each other; at least for an order of one, which allows for an average prediction accuracy between 1.9 and 2.2. For OPL, the Bayesian model selection and the cross validation approaches do not directly agree with each other; i.e., the Bayesian method suggest an order of four while, interestingly, cross validation would prefer an order of zero (weighted random selection).

In general, by using at least first-order Markov chains it is possible to predict the property a user is most likely to change next for all datasets, except OPL. For steering users into the right direction, we may recommend appropriate properties to change next to contributors.

5.5 Predicting Properties for Classes

Our Bayesian model selection results (see Table 2) suggests for ICD-11 and ICTM a second- and first-order Markov chain model respectively. Furthermore, the results indicate that for BRO a third- and for OPL a fifth-order seem to be appropriate. A stratified 3-fold cross-fold validation (see Figure 3(e) and Table 2) yielded the same results, except for ICD-11, where a first-order model, instead of a second-order model, represents the best predictive accuracy for the underlying data. The conducted cross-fold validation prediction task yielded an accuracy (average position) between roughly 1.8 and 2.4.

Again, our results indicate that we can predict the property that is changed next for a given class reasonably well by using at least a first-order Markov chain. Similar to predicting properties for users, we may now enhance the overall quality of the ontology in an automatic way by aligning the gained information with the properties derived from our user analysis results and recommend users to change specific suitable properties of classes next.

6. SUMMARY AND DISCUSSION

In the previous sections we have studied action sequences of five collaborative ontology-engineering projects from the biomedical domain (see Section 2). To begin with, we provided an initial analysis regarding regularities and sequential patterns in Section 3 to give a basic insight into the processes underlying the user action sequences at hand. First, we started by looking at randomness and regularities by applying an adopted version of the so-called *runs test* exemplary to the ICD-11 dataset in Section 3.1. Our results clearly indicated that a significant array of sequences, based on different features, are produced in a non-random way; this means that at least a portion of sequences is produced in a clustered or systematic arrangement. These observations warranted further studies regarding detailed insights into how these potential regularities look like; hence, we focused on mining sequential patterns next (see Section 3.2). We applied *PrefixSpan* on our User sequences and could identify numerous sequential patterns of longer length – specifically lengths 2 to 4. This lead us to the conclusion that longer patterns seem to play a crucial role in contributor logs of collaborative ontology-engineering projects and that there might be a dependence between subsequent elements in the sequences at hand. Consequently, we hypothesized that it would be beneficial to consider memory effects when modeling our data, and thus user actions. This means, that we wanted to incorporate information of the past into deriving future information – for example, it might be useful to check the two past properties a user has changed for predicting the property she will most likely change next.

For doing so we resorted to Markov chain models of varying order (see Section 4.1) that we applied to our data. We used a Bayesian model selection method for finding the appropriate order for each set of sequences at interest. Supplementary, we were interested in investigating the predictive power of such models, which we evaluated using a cross validation task as described in Section 4.2. The results, as shown in Section 5, confirm our hypotheses: It is indeed useful to incorporate memory effects into the process of modeling user contribution in collaborative ontology-engineering projects. This is particularly imminent as several higher order models are to be preferred throughout all investigations, as can be seen in Table 2. For example, an order of three means that we can best model or predict the next event (e.g., property) by looking at the past three events in a sequence – hence, memory effects are in play. We need to note that all our applied methods compensate the goodness of fit with the corresponding complexity of a model, thus, we penalize higher orders (Occam's razor) which is a necessary step for accounting for potential overfitting.

We can see that both the Bayesian model selection as well as the cross validation prediction task mostly result in similar order suggestion even though they are based on distinct approaches. If the outcome of both methods differ, we can for the most part observe that the cross validation method ensues slightly lower orders than the Bayesian method. This can be explained by the different ways both methods work. The Bayesian method always learns the Markov chain model on the complete model and then performs a model selection strategy which is based on comparing the posterior probabilities of varying order models. Contrary, the cross validation technique learns the Markov chain on a different set (training) compared to where it is evaluated (testing). These differences also account for the drastic mismatch observed between the cross-fold validation prediction task and the Bayesian model selection for OPL in our *Predicting Properties for Users* analysis, where only a very limited number of sequences (three) with unevenly distributed properties across these sequences, is available. Also, the way we rank the probabilities in the cross validation evaluation influences the outcome. Currently, we use modified competition ranking which assigns the worst rank to ties and hence, we very strictly penalize higher orders. Hence, it comes to no surprise for us that if different, the cross validation mostly suggest lower orders than the Bayesian approach. One advantage of the Bayesian approach though is that we could further incorporate penalizations of higher orders when working with model selection; e.g., using an exponential prior [27].

In general, the application of Markov chains on the activity logs of five collaborative ontology-engineering projects has shown that regularities exist. These regularities can potentially be used and exploited by project and community managers to augment and assist users in contributing to the underlying structured knowledge representation. For example, knowing which property a user is most likely to change next and which user is most likely to change a specific concept next could be used to automatically adjust and modify the interface to allow for quicker and personalized workflows. This is especially important for projects the size of ICD-11 or NCIt with thousands of potential classes to contribute to.

We also need to note that the corresponding orders that get suggested might also be – at least to some extent – influenced by how the sequences are shaped; i.e., potential influence factors might be: the distribution of the length of sequences or the number of sequences in a dataset. However, we can argue that these are also properties emerging from how users behave in such systems. Yet, if we are specifically interested in comparing the models of different datasets we need to look deeper into these factors which we leave open for future work. Furthermore, we only work with limited data which also influences the choice of order. Precisely, the number of distinct states as well as the number of observations affect the appropriate order. Basically, the more states one works with, the more difficult it is to compensate the much higher complexity of higher order models with the goodness of fit. Also, we do not necessarily know what would happen if we would perform our investigations on an unlimited number of observations; most likely higher orders will then statistically significantly outperform lower ones (that we e.g., found in our studies) – notwithstanding, working with limited data is a common scenario for researchers and practitioners warranting our experiments and findings.

7. RELATED WORK

The work presented in this paper was inspired by work of the following research areas: Collaborative ontology-engineering, Markov chains and sequential pattern mining.

7.1 Collaborative Ontology Engineering

An ontology represents an explicit specification of a shared conceptualization [14, 5, 32]. In computer-science, this definition usually refers to a construct (formalization) that is automatically processable by a machine representing an abstraction of the real world (shared conceptualization). Ontologies allow computers to "understand" relationships between entities and objects that are modeled in an ontology.

On the other hand, collaborative ontology engineering represents a new field of research with many new problems, risks and challenges. Contributors of such projects, similar to Wikipedia, engage remotely (e.g., via the Internet or a client–server architecture) in the development process to create and maintain an ontology. As mentioned, an ontology represents a formalized and abstract representation of a specific domain; thus, disagreements between authors on certain subjects can occur and tools are needed that augment collaboration and help contributors in reaching consensus when modeling these (and other) topics. Indeed, the majority of the literature about collaborative ontology engineering sets its focus on surveying, finding and defining requirements for the tools used in these projects [20, 13]. Various tools have been developed, specifically aiming at supporting the collaborative development of ontologies. For example, Semantic MediaWikis [18] and its derivatives [2, 12, 26] add semantic, ontology modeling and collaborative features to traditional MediaWiki systems.

Protégé, WebProtégé [34] and its extensions and derivatives for collaborative development are prominent stand-alone tools that are used by a large community worldwide to develop ontologies in a variety of different projects. Both WebProtégé (and its derivatives) and Collaborative Protégé have shown to provide a robust and scalable environment for collaboration and are used in several large-scale projects, including the development of ICD-11 [33].

For analyzing and visualizing the collaborative processes that occur during these projects, Pöschko et al. [24] and Walk et al. [36] have developed *PragmatiX*, a tool that allows to visualize and analyze aspects of the history of collaboratively engineered ontologies. The tool also provides quantitative insights into the ongoing collaborative development processes. Falconer et al. [11] investigated the change-logs of collaborative ontology-engineering projects, showing that users exhibit regularities in their contribution behavior when editing to the ontology. Strohmaier et al. [31] analyzed the collaborative processes in a number of different collaborative ontology-engineering projects by investigating hidden social dynamics and provide new metrics to quantify various aspects of these engineering processes. Wang et al. [39] used association-rule mining to analyze user editing patterns in collaborative ontology-engineering projects.

7.2 Markov chain models

In previous Web studies, Markov chain models have been frequently applied for understanding and modeling Web navigation (e.g., [23, 10, 42]). Mostly, the used Markov chain models were memoryless following the Markovian assumption which is e.g., also modeled in the *random surfer model* in Google's PageRank[8]. Nonetheless, various researchers were also interested in studying the appropriateness of modeling memory effects into models of human navigation – i.e., using higher order chains (e.g., [4, 23]). Yet, the studies revealed that the benefit of higher orders can frequently not compensate the higher complexity and the first-order Markov chain model seems to be a plausible choice. Recently, Chierichetti et al. [9] turned towards again questioning the choice of a first-order chain for modeling human navigation and suggested that the Markovian assumption might be wrong. Consequently, Singer et al. [27] introduced a series of precise model selection techniques for choosing the appropriate Markov chain order. They applied the framework to a series of human navigational datasets and again showed that the memoryless model indeed seems to be a plausible abstraction for human navigation based on the lack of statistically significant improvements of higher order models mostly due to the much higher complexity as already pointed out several years ago. However, the authors also showed that human navigation on a topical level reveals memory effects. Walk et al. [37] adopted this framework to be applicable to structured logs of changes in collaborative ontology-engineering projects and investigated the structure of first-order Markov chains for the change-logs of five different collaborative ontology-engineering projects [38].

7.3 Sequential Pattern Mining

In 1995, Agrawal and Srikant [1] have first addressed the problem of sequential pattern mining. They stated that given a collection of chronologically ordered sequences, sequential pattern mining is about discovering all sequential (chronologically ordered) patterns weighted according to the number of sequences that contain these patterns. The algorithms presented in Agrawal and Srikant [1], in particular AprioriAll and AprioriScale, represent the first *a priori* sequential pattern mining algorithm. In 1996, Srikant and Agrawal [29] further included time-constraints and sliding windows to the definition of sequential patterns and introduced the generalized sequential pattern algorithm (GSP). This means that a specific pattern cannot occur more frequently (above a threshold) if a sub-pattern of this pattern occurs less often (below that threshold). Many other examples of a priori algorithms have been discussed in literature [19, 40, 3], with SPADE [41] being one of the most prominently used and referred to algorithms. One major problem assigned to the a priori based sequential pattern mining algorithms was (in the worst case) the exponential number of candidate generation. To tackle this problem so called pattern-growth approaches have been developed [15, 22].

Many researchers have adapted different algorithms and approaches for different domains to anticipate changing requirements, such as

[16] who analyzed algorithms for sequential pattern mining in the biomedical domain. In Walk et al. [37] the authors have presented a novel application of Markov chains to mine and determine sequential patterns from the structured logs of changes of collaborative ontology-engineering projects.

For the analysis presented in this paper we made use of *PrefixSpan* [22] to investigate if the change-logs of collaborative ontology-engineering projects exhibit commonly used, sequential patterns – we thoroughly introduced this algorithm in Section 3.2.

8. CONCLUSIONS & FUTURE WORK

In this paper our main objective was to predict user actions in collaborative ontology-engineering projects. To that end, we first *explored* if and to what extent regularities and sequential patterns can be extracted from the change-logs of our five datasets. We found that at least a set of sequences were produced in a non-random way and that frequent (longer) patterns can be extracted. We then *modeled* user actions by using Markov chain models which allowed us to incorporate our findings about regularities and patterns. We fitted the models to our sequence data and evaluated them with a specific focus on prediction accuracy. We found that incorporating memory effects (serial dependence) into our models can indeed be useful. The generated predictive models for user actions can not only be used for various recommendation purposes, but also provide project administrators and managers with the means to assess the impact of potential changes on the ontology and the community. For example, knowing which user is most likely to change a specific concept next combined with the information of what kind of change that user is most likely to perform next can potentially be exploited to create personalized task recommendations or to adapt the user-interface to allow for dynamically assisted and faster workflows.

In future work, we first want to extend our choice of models for predicting user action by exploring, for example, varying order Markov chain models, Hidden Markov chain models or Semi Markov chain models. When fitting these models to the data, we plan on providing further evaluation comparisons between these distinct models and consequently, also want to explore the potential of incorporating memory into these alternative models. Furthermore, we want to look at other data sources (e.g., Semantic MediaWikis) to be able to produce more general statements, independent from the datasource, and also closely investigate the influence of different data properties as discussed in Section 6.

We strongly believe that the analysis and predictive models presented in this paper represents an important step towards a better understanding of collaborative ontology-engineering projects in the biomedical domain.

Acknowledgements

We want to thank Natalya F. Noy, Tania Tudorache and Mark A. Musen from the Stanford Center for Biomedical Informatics Research (BMIR) for providing access to the data. This work was partially funded by the DFG German Science Fund grant STR 1191/3-2.

9. REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society.

[2] S. Auer, S. Dietzold, and T. Riechert. Ontowiki - a tool for social, semantic collaboration. In *Proceedings of the 5th International Conference on The Semantic Web*, ISWC'06, pages 736–749, Berlin, Heidelberg, 2006. Springer-Verlag.

[3] C. Bettini, X. S. Wang, and S. Jajodia. Testing complex temporal relationships involving multiple granularities and its application to data mining (extended abstract). In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '96, pages 68–78, New York, NY, USA, 1996. ACM.

[4] J. Borges and M. Levene. Data mining of user navigation patterns. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, WEBKDD '99, pages 92–111, London, UK, UK, 2000. Springer-Verlag.

[5] W. Borst. Construction of engineering ontologies for knowledge sharing and reuse. 1997.

[6] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*. John Wiley & Sons, 2013.

[7] J. V. Bradley. Distribution-free statistical tests. 1968.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[9] F. Chierichetti, R. Kumar, P. Raghavan, and T. Sarlos. Are web users really markovian? In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 609–618, New York, NY, USA, 2012. ACM.

[10] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Trans. Internet Technol.*, 4(2):163–184, May 2004.

[11] S. Falconer, T. Tudorache, and N. F. Noy. An analysis of collaborative patterns in large-scale ontology development projects. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP '11, pages 25–32. ACM, 2011.

[12] C. Ghidini, B. Kump, S. Lindstaedt, N. Mahbub, V. Pammer, M. Rospocher, and L. Serafini. Moki: The enterprise modelling wiki. In *The Semantic Web: Research and Applications*, pages 831–835. Springer, 2009.

[13] T. Groza, T. Tudorache, and M. Dumontier. Commentary: State of the art and open challenges in community-driven knowledge curation. *Journal of Biomedical Informatics*, 46(1):1–4, Feb. 2013.

[14] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[15] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 1–12, New York, NY, USA, 2000. ACM.

[16] C.-M. Hsu, C.-Y. Chen, B.-J. Liu, C.-C. Huang, M.-H. Laio, C.-C. Lin, and T.-L. Wu. Identification of hot regions in protein-protein interactions by sequential pattern mining. *BMC bioinformatics*, 8(Suppl 5):S8, 2007.

[17] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[18] M. Krötzsch, D. Vrandečić, and M. Völkel. Semantic mediawiki. In *The Semantic Web-ISWC 2006*, pages 935–942. Springer, 2006.

[19] H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.

[20] N. F. Noy and T. Tudorache. Collaborative ontology development on the (semantic) web. In *AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering*, pages 63–68. AAAI, 2008.

[21] P. C. O'Brien and P. J. Dyck. A runs test based on run lengths. *Biometrics*, pages 237–244, 1985.

[22] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th International Conference on Data Engineering*, ICDE '01, pages 215–224, Washington, DC, USA, 2001. IEEE Computer Society.

[23] P. L. T. Pirolli and J. E. Pitkow. Distributions of surfers' paths through the world wide web: Empirical characterizations. *World Wide Web*, 2(1-2):29–45, Jan 1999.

[24] J. Pöschko, M. Strohmaier, T. Tudorache, N. F. Noy, and M. A. Musen. Pragmatic analysis of crowd-based knowledge production systems with iCAT Analytics: Visualizing changes to the ICD-11 ontology. In *Proceedings of the AAAI Spring Symposium 2012: Wisdom of the Crowd*, 2012.

[25] M. Rosvall, A. V. Esquivel, A. Lancichinetti, J. D. West, and R. Lambiotte. Memory in network flows and its effects on spreading dynamics and community detection. *Nature Communications*, 5, 2014.

[26] T. Schandl and A. Blumauer. Poolparty: SKOS thesaurus management utilizing linked data. *The Semantic Web: Research and Applications*, 6089:421–425, 2010.

[27] P. Singer, D. Helic, B. Taraghi, and M. Strohmaier. Detecting memory and structure in human navigation patterns using markov chain models of varying order. *PloS one*, 9(7):e102070, 2014.

[28] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, and L. W. Wright. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, February 2007.

[29] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK, 1996. Springer-Verlag.

[30] C. C. Strelioff, J. P. Crutchfield, and A. W. Hübler. Inferring markov chains: Bayesian estimation, model comparison, entropy rate, and out-of-class modeling. *Physical Review E*, 76:011106, Jul 2007.

[31] M. Strohmaier, S. Walk, J. Pöschko, D. Lamprecht, T. Tudorache, C. Nyulas, M. A. Musen, and N. F. Noy. How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects. *Web Semantics: Science, Services and Agents on the World Wide Web*, 20(0), 2013.

[32] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. volume 25, pages 161–197, 1998.

[33] T. Tudorache, S. Falconer, C. Nyulas, N. F. Noy, and M. A. Musen. Will semantic web technologies work for the development of icd-11? In *The Semantic Web–ISWC 2010*, pages 257–272. Springer, 2010.

[34] T. Tudorache, C. Nyulas, N. F. Noy, and M. A. Musen. WebProtégé: A Distributed Ontology Editor and Knowledge Acquisition Tool for the Web. *Semantic Web Journal*, 4(1/2013):89–99, 2013.

[35] A. Wald and J. Wolfowitz. On a test whether two samples are from the same population. *The Annals of Mathematical Statistics*, 11(2):147–162, 1940.

[36] S. Walk, J. Pöschko, M. Strohmaier, K. Andrews, T. Tudorache, C. Nyulas, N. F. Noy, and M. A. Musen. PragmatiX: An Interactive Tool for Visualizing the Creation Process Behind Collaboratively Engineered Ontologies. *International Journal on Semantic Web and Information Systems*, 9(1):45–78, 2013.

[37] S. Walk, P. Singer, M. Strohmaier, D. Helic, N. F. Noy, and M. A. Musen. Sequential usage patterns in collaborative ontology-engineering projects. *arXiv preprint arXiv:1403.1070*, 2014.

[38] S. Walk, P. Singer, M. Strohmaier, T. Tudorache, M. A. Musen, and N. F. Noy. Discovering beaten paths in collaborative ontology-engineering projects. *Journal of Biomedical Informatics*, 2014.

[39] H. Wang, T. Tudorache, D. Dou, N. F. Noy, and M. A. Musen. Analysis of user editing patterns in ontology development projects. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, OTM '13, pages 470–487. Springer, 2013.

[40] J. T.-L. Wang, G.-W. Chirn, T. G. Marr, B. Shapiro, D. Shasha, and K. Zhang. Combinatorial pattern discovery for scientific data: Some preliminary results. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, SIGMOD '94, pages 115–125, New York, NY, USA, 1994. ACM.

[41] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60, 2001.

[42] I. Zukerman, D. W. Albrecht, and A. E. Nicholson. *Predicting usersâĂŹ requests on the WWW*. Springer, 1999.

3.5 Understanding How Users Edit Ontologies: Comparing Hypotheses About Four Real-World Projects

The last paper of my thesis tackles the third research question. In particular, colleagues and I were interested to formally define, express and evaluate the edit patterns identified in previous empirical studies. To that end, colleagues and I first formulated a set of different hypotheses, which represent beliefs about how users collaboratively edit and develop ontologies and are inferred from the results of previous empirical studies. To be precise, each Hypothesis is represented as a weighted transition matrix. The weights for each of the transition matrices represent the belief in specific transitions to occur. To be able to compare and rank the generated hypotheses, colleagues and I make us of the HypTrails framework [Singer et al., 2015].

The analysis conducted in this article not only describes how to formulate different hypotheses about how users edit ontologies, but also confirms the results of previous empirical analyses and represents a very important step towards a better understanding and evaluation of how users engineer ontologies "in the wild".

Understanding How Users Edit Ontologies: Comparing Hypotheses About Four Real-World Projects

Simon Walk[1], Philipp Singer[2], Lisette Espín Noboa[2], Tania Tudorache[3],
Mark A. Musen[3], and Markus Strohmaier[2,4]

[1] Graz University of Technology
`simon.walk@tugraz.at`
[2] GESIS - Leibniz Institute for the Social Sciences
`firstname.lastname@gesis.org`
[3] Stanford Center for Biomedical Informatics Research
`lastname@stanford.edu`
[4] University of Koblenz-Landau
`lastname@uni-koblenz.de`

Abstract. Ontologies are complex intellectual artifacts and creating them requires significant expertise and effort. While existing ontology-editing tools and methodologies propose ways of building ontologies in a normative way, empirical investigations of how experts *actually* construct ontologies "in the wild" are rare. Yet, understanding actual user behavior can play an important role in the design of effective tool support. Although previous empirical investigations have produced a series of interesting insights, they were exploratory in nature and aimed at gauging the problem space only. In this work, we aim to advance the state of knowledge in this domain by systematically defining and comparing a set of hypotheses about how users edit ontologies. Towards that end, we study the user editing trails of four real-world ontology-engineering projects. Using a coherent research framework, called HypTrails, we derive formal definitions of hypotheses from the literature, and systematically compare them with each other. Our findings suggest that the *hierarchical structure* of an ontology exercises the strongest influence on user editing behavior, followed by the *entity similarity*, and the *semantic distance* of classes in the ontology. Moreover, these findings are strikingly consistent across all ontology-engineering projects in our study, with only minor exceptions for one of the smaller datasets. We believe that our results are important for ontology tools builders and for project managers, who can potentially leverage this information to create user interfaces and processes that better support the observed editing patterns of users.

1 Introduction

Large real-world ontologies are intellectual artifacts that are inherently complex and hard to build. Most such ontologies are found in the biomedical domain. For example, SNOMED-CT,[5] a comprehensive clinical health terminology, has over $300,000$ classes, the National Cancer Institute Thesaurus (NCIT)[6] has more than $100,000$ classes, and the

---

[5] `http://www.ihtsdo.org/snomed-ct`
[6] `http://ncit.nci.nih.gov`

11th revision of the International Classification of Diseases (ICD-11)[7] has over 50,000 classes. The development of such large ontologies usually takes place in distributed teams, and requires a significant effort both in the ontological modeling and coordination of the entire process.

One of the biggest challenges in developing large real-world ontologies is proper tool support. While existing ontology-editing tools and methodologies prescribe certain ways of building ontologies, there is very little research on how users actually use these tools. Empirical analyses of how users develop ontologies “in the wild” are very rare. We address this gap with this paper, by aiming to broaden our understanding of editing behaviors in large ontology-engineering projects. It is the ultimate vision of our work to lay a more solid foundation for creating tools that better support ontology authors based on their actual authoring behavior.

We define a *sequential edit trail* as a chronologically sorted list of all actions a user takes while editing an ontology. We derive such editing trails from the change logs recorded by the ontology-editing tools. In previous work, we have conducted exploratory empirical analyses of various types of edit trails in several ontology-engineering projects [21, 22], and we have discussed our findings and potential implications [23]. In these works, we have been able to explore different editing patterns and potential explanations via manual inspection and qualitative interpretation. For example, we have speculated that users edit ontologies in a top-down fashion or that users navigate along similar concepts. However, it is still unclear how such hypotheses can best be expressed formally, or how they can be systematically compared with each other in order to explain the production of edit trails, and hence an ontology, at hand.

Thus, in this paper, we systematically investigate previous, mostly exploratory, results using HypTrails [11]—a generic methodology for comparing hypotheses about human trails in ontology-engineering projects. This allows us to (i) formally define, (ii) systematically study, and (iii) rank different hypotheses about ontology-editing behavior within a coherent research framework. By using HypTrails, we approach this problem by modeling edit trails as first-order Markov chains (see Section 3.2) and hypotheses as priors. From our analyses, we find that the *hierarchical structure* of an ontology exercises the strongest influence on observed user behaviors, followed by the *similarity* of entities, and the *distance* of classes in the ontology. These findings are strikingly consistent across the four real-world ontology-engineering projects used in our study, with only minor exceptions for one of the smaller datasets. We believe that our results are important for ontology tools builders and for project managers, who can potentially leverage this information to create user interfaces and processes that better support the observed editing patterns of users.

The main research contributions of this work are:

- A formal way to define hypotheses about how users edit an ontology (e.g., top-down vs. bottom-up editing strategies).
- A detailed systematic comparison of such hypotheses across four real-world ontology-engineering projects.
- A ranking of all investigated hypotheses according to their relative plausibility for each dataset by adopting a coherent research approach.

---

[7] `http://who.int/classifications/icd/revision/en/`

The remainder of the paper is structured as follows: In Section 2, we discuss the related work. The methodology and datasets are described in Section 3, followed by a detailed formal description of all investigated hypotheses in Section 4. We present the results of our analysis in Section 5, discuss implications and limitations of our findings in Section 6 and conclude our work and discuss opportunities for future work in Section 7.

2 Related Work

The related work relevant for this paper is covered by two different research fields: *Human Trails on the Web* and *Analysis of Ontology Editing Behavior*.

2.1 Human Trails on the Web

Previous research has studied human trails on the Web in various settings. Modeling trails has received a lot of attention [3, 12], as well as the detection of regularities, patterns and strategies in trails of interest [6, 25]. Most prominently, researchers have focused on studying human navigational trails on the Web—capturing the subsequent websites that humans navigate to [6, 12, 25]. This research on navigational trails has inspired other works in the effort to improve the Web, e.g., better website design (usability) [4], identifying related links [18] or constructing an e-learning Semantic Web [2]. Researchers have also investigated other kinds of human trails, e.g., search trails [13, 26], diffusion trails [1] or song listening trails [11]. Our work directly connects to these studies as we are interested in shedding more light on the production of human trails on the Web; however, in our case, we look at human edit trails in ontology-engineering projects by using the approach presented in [11].

2.2 Analysis of Ontology Editing Behavior

In this line of research, a large part of the literature has focused on analyzing the editing behavior or identifying editing patterns in collaborative ontology-engineering. To perform these types of analyses, researchers have used the change logs recorded by the different ontology-editing environments, similar to our approach.

Strohmaier et al. [14] conducted an empirical analysis to investigate the hidden social dynamics that take place when editors develop an ontology, and provided new metrics to quantify various aspects of the engineering processes. Falconer et al. [5] did a change-log analysis of different ontology-engineering projects, showing that contributors exhibit specific roles, which can be used to group and classify these users. Pesquita and Couto [9] analyzed the influence of the location and specific structural features to determine if and where the next change will be conducted in the Gene Ontology[8]. The work by Wang et al. [24] presents an analysis of user editing patterns derived from change logs of several real-world ontology-engineering projects utilizing association-rule mining. The results suggest that users tend to edit in a vertical way, i.e., users edit the same properties for different classes in a sequential way. Rospocher et al [10] analyzed the

---

[8] `http://www.geneontology.org`

change logs for two different Web-based collaborative ontology-editing tools and found similar collaboration and editing patterns. For example, they found that users tend to edit in the local neighborhood of an entity. Van Laere et al. [19] analyzed behavior-based user profiles in collaborative ontology-engineering projects using K-means clustering to group similar users.

In contrast to our previous research [21–23], this work represents a systematic and comparative study of different hypotheses in a coherent mathematical research framework, whereas our previous analyses have mostly been exploratory. We can thereby—for the first time—make relative, empirically grounded statements about the plausibility of different hypotheses given data.

3 Materials & Methodology

We present the four datasets used in our research (Section 3.1), and the HypTrails framework (Section 3.2) that forms the basis of the methodology used in this work.

3.1 Datasets

We used the change logs of four real-world ontology-engineering projects to conduct the analyses presented in this work. These projects use WebProtégé [17] as the editing platform, a Web-based generic ontology-editing tool, which records a log of all changes performed by each user. Each change record stores metadata about the change, such as the user who performed the change, a textual description of the change, the timestamp, and the entity on which the change occurred.

To extract the editing trails from the change logs, we performed a pre-processing step in which we merged consecutive changes on the same entity by the same user (i.e., *self-loops*) into one change. Such changes occurred when users would edit different properties of the same entity. For the purpose of this work, we have not been interested in such changes, but rather in the ones which occurred on different entities. Further, we have limited all our analyses on *isA* relationships and removed equivalence links. However, multiple *isA* inheritances have been kept “as-is”. We provide a brief description of the four datasets used in our research below.

The International Classification of Diseases (ICD),[9] developed by the World Health Organization (WHO), is the international standard for diagnostic classification used to encode information relevant to epidemiology, health management, and clinical use in over one hundred United Nations countries. WHO regularly publishes new revisions of the classifications. The 11th revision of the classification, **ICD-11**,[10] is currently in progress, and is planned to be finalized in 2017. In contrast to previous revisions, ICD-11 is developed as a rich OWL ontology [16]. Over 100 domain experts are using a customized version of WebProtégé to author the ontology collaboratively.

The International Classification of Traditional Medicine (ICTM)[11] is a WHO-led project that aimed to produce an international standard terminology and classification

---

[9] `http://who.int/classifications/icd/en/`

[10] `http://who.int/classifications/icd/ICDRevision/`

[11] `http://who.int/mediacentre/news/notes/2010/trad_medicine_20101207/en/`

for diagnoses and interventions in Traditional Medicine. ICTM was developed collaboratively as an OWL ontology with the goal to unify the knowledge from the traditional medicine practices from China, Japan and Korea. Its content is authored in 4 languages: English, Chinese, Japanese and Korean. More than 20 domain experts from the three countries developed ICTM using a customized version of WebProtégé. The development of ICTM ended in 2012.

The Biomedical Resource Ontology (BRO) [15] was developed as part of the Biositemaps project. Biositemaps is a mechanism for researchers working in biomedicine to publish metadata about biomedical data, tools, and services. Applications can then aggregate this information for tasks such as semantic search. BRO is the enabling technology used in Biositemaps; a controlled terminology for describing the resource types, areas of research, and activity of a biomedical related resource. A small group of editors authored BRO using WebProtégé to modify the ontology and to carry out discussions.

The Ontology for Parasite Lifecycle (OPL) models the life cycle of the *T.cruzi*, a protozoan parasite, which is responsible for a number of human diseases [8]. OPL uses expressive OWL (SHOIF) to represent its knowledge base, and extends several other OWL ontologies. Several users from different institutions collaborate on OPL development using WebProtégé as a collaborative platform.

Table 1 provides some characteristics about each of the datasets used in our analysis. The average trail length ranges from 1,637.13 transitions for ICD-11 to 136.60 transitions for BRO. Trails refer to the number of different human edit trails per dataset, where each trail represents a chronologically ordered list of all the classes a user has edited. Users with less than 2 distinct changes have been removed from our analysis.

Table 1. Characteristics of the four datasets.

	ICD-11	ICTM	BRO	OPL
Classes	48,771	1,506	528	393
Changes	439,229	67,522	2,507	1,993
Users	109	27	5	3
Trails	102	26	5	3
Avrg. trail length	1,637.13	673.54	136.60	152.00
Transitions	361,491	66,708	2,388	2,668
Self-Loops	194,504	49,196	1,705	2,212
First change	18.11.2009	02.02.2011	12.02.2010	09.06.2011
Last change	29.08.2013	17.7.2013	06.03.2010	23.09.2011
Period (ca.)	4 years	2.5 years	1 month	3 months

3.2 Methodology

By and large, HypTrails [11] is an approach that allows us to compare hypotheses about human trails. In our case, we are interested in studying: (i) the human edit trails in ontology-engineering projects, and (ii) the relative plausibility of hypotheses about the production of these trails that have been manifested in previous studies. In Section 1, we used the hypothesis that users edit ontologies in a top-down manner as an example. Using HypTrails, we are able to compare this hypothesis to other such hypotheses, and determine which one is more plausible to describe the production of the corresponding editing trails, and hence the ontology at hand. Section 4 provides a formal description of all hypotheses that we have compared as part of this research. Figure 1 shows a graphical representation of the editing patterns represented by each hypothesis. Next, we introduce the core concepts of HypTrails; for a more thorough introduction please refer to [11].

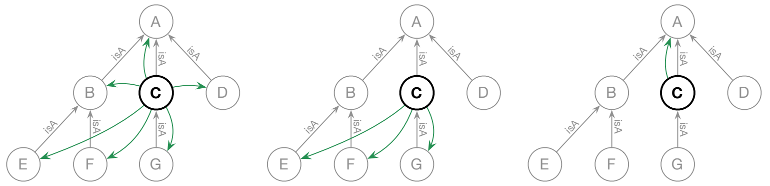
Technically, HypTrails models trails with first-order Markov chain models, and compares hypotheses using Bayesian inference, and more specifically, the *marginal likelihood* which can also be referred to as the *evidence* (we use both terms throughout this work synonymously). The marginal likelihood $P(H|D)$ describes the probability of a hypothesis H (e.g., uniform hypothesis) given the data (trails). For expressing generic hypotheses and being able to compare them, HypTrails uses the sensitivity of the marginal likelihood on the prior. Thus, hypotheses are expresses as different priors—in case of a Markov chain model the conjugate prior is the *Dirichlet distribution*. The hyperparameters of Dirichlet distributions can be interpreted as *pseudo counts*. Thus, simply put, higher pseudo counts refer to higher beliefs in corresponding transition for a given hypothesis.

Consequently, we have to provide HypTrails with matrices that capture our generic hypotheses and corresponding beliefs in transitions (see Section 4). Based on these matrices, HypTrails internally elicits proper Dirichlet priors for given hypotheses by setting the pseudo counts accordingly, based on a parameter k which steers the total number of pseudo counts assigned. Basically, the higher we set k, the stronger we believe in a given hypothesis. Analogously, this means that with higher k, we expect to see less transitions contradicting the corresponding hypothesis (e.g., only transitions from higher level classes to lower level classes in the top-down hypothesis). For fairness, we always want to compare hypotheses with each other for the same values of k.

Finally, by using different priors for different hypotheses, we get different marginal likelihoods when combined with empirical trail data. Based on these evidences, we can compare the relative plausibility of hypotheses—higher evidences indicate higher plausibility. In theory, we need to further calculate *Bayes factors* [7] between the marginal likelihoods of two hypotheses, so that we would be able to judge the strength of the evidence for one hypothesis over the other. However, as all Bayes factors are decisive, we resort from presenting them individually throughout this paper. Thus, we can produce a partial ordering of hypotheses based on their relative plausibility by ranking their marginal likelihoods from largest to smallest for single values of k.

4 Hypotheses

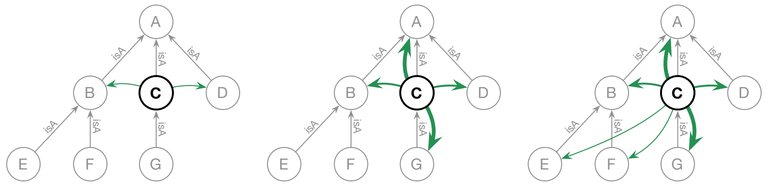
HypTrails allows us to compare hypotheses about the production of human edit trails in ontology-engineering projects, and helps us to understand how an ontology is produced in an ontology-development tool. Hypotheses are *beliefs about transitions* (see Figures 1(a)–1(h)) opposed to actual empirical transitional observations (see Figure 1(i)). With HypTrails, we express these transitional beliefs as our assumptions about Markov



(a) Uniform hypothesis: all transitions are equally likely

(b) Top-down hypothesis: transitions to lower classes are most likely

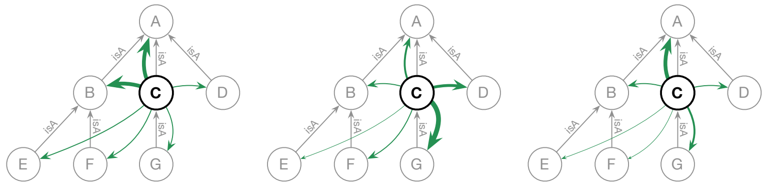
(c) Bottom-up hypothesis: transitions to higher classes are most likely



(d) Breadth-first hypothesis: transitions to same level classes are most likely

(e) Hierarchy hypothesis: transitions along hierarchical relations (parent, child, sibling, cousin) are most likely

(f) Shortest path hypothesis: transitions to close classes are most likely



(g) Connectivity hypothesis: transitions to popular classes are most likely

(h) Similarity hypothesis: transitions to similar (title and definition) classes are most likely

(i) Empirical transitions: obtained from real world data

Fig. 1. Sample-Hypotheses. This figure depicts eight hypotheses about how humans consecutively edit classes in ontology-engineering projects derived from our previous research (a-h), as well as empirical observations (i). The curved arrows represent transitions we believe in for a given hypothesis (a-h), or observed transition probabilities from data (i). The thicker an arrow, the higher our belief in the corresponding transition for a given hypothesis (a-h), or the higher the number of transitions we observed in the data (i). For simplicity, we always only visualize the transitions for class *C*; all other classes follow analogously.

chain transitions. In detail, we specify hypotheses as matrices that reflect our assumptions about transitions between states where higher values correspond to higher beliefs.

Thus, for each hypothesis, we need to specify the *hypothesis matrix* Q with elements $q_{i,j}$ that represent the belief in the transition between states s_i and s_j. A *state* corresponds to a class in the ontology that users are editing. A *transition* between states s_i and s_j corresponds to a two sequential user edit: first of the class represented by s_i, and then of the class represented by s_j. In order to express our hypotheses as beliefs in Markov transitions, and to have a better interpretation capability, we directly set $q_{i,j}$ as row probabilities $P(s_j|s_i)$. Thus, for each row i of Q it holds that $\sum_j q_{i,j} = 1$.

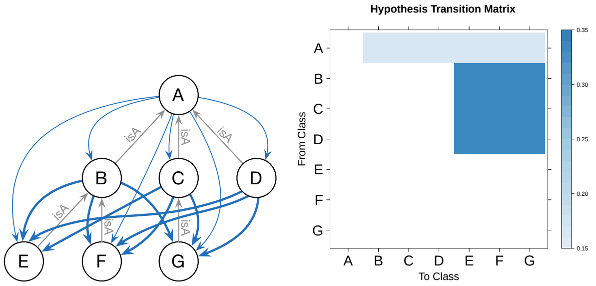
For example, Figure 1(e) depicts the *hierarchy-based hypothesis*, which postulates the belief that users are likelier to edit classes along the hierarchical (*isA*) structure of the ontology and the shortest distance. In this example, if a user has just previously changed class C, this hypothesis believes that the user is most likely to change class A (the *parent*) or G (the *child*) next. Classes B and D are both *siblings* (and two steps away) of C, which is why this hypothesis expresses a smaller belief in these transitions. Other hierarchical transitions, *ancestors*, *descendants* and *cousins*, follow analogously with less belief (i.e., lower proabability; not depicted in Figure 1(e)).

Figure 2 shows an exemplary illustration of the transition graph and the corresponding matrix for the *top-down* hypothesis, which believes that users consecutively edit classes at deeper levels in the hierarchy. In this example, our state space consists of seven classes $S = \{A,B,C,D,E,F,G\}$. The beliefs in the transitions between states are shown in Figure 2(a). As this hypothesis has stronger beliefs in top-down transitions, the graph and matrix will only contain beliefs in transitions from higher-level classes to lower-level classes, such as, from C to E, F and G. Figure 2(b) shows the corresponding representation of the beliefs in the hypothesis matrix. For example, for the row corresponding to the transitions from class C, we may set $q_{C,E} = 1/3$, $q_{C,F} = 1/3$ and $q_{C,G} = 1/3$. For all other classes, we can proceed analogously.

In the remainder of this section, we thoroughly describe the hypotheses used in this research, and provide formal descriptions of how we built the corresponding hypothesis matrices Q. Note that for each hypothesis and equation, we always calculate $q_{i,j}$, for all i and j. We set the diagonal of each hypothesis matrix Q to 0 as we do not consider self-loops in our data. As it is not always possible to express our beliefs with direct probabilities, we additionally normalize each row of Q using the ℓ_1-norm.

Figure 1 shows a graphical representation of the hypotheses investigated in our research. The *top-down*, *bottom-up*, *breadth-first* and *hierarchy hypotheses* resulted as part of our prior research from a manual inspection of Markov chains of different orders [21–23]. Additionally, we are also considering the *shortest path*, *connectivity*, and *similarity* hypotheses to also investigate further “strategies” of how users edit an ontology that could provide plausible explanations for the underlying data.

Uniform hypothesis. This hypothesis believes that each transition from one state to any other state is equally likely (cf. Figure 1(a)). Thus, it assumes that humans edit ontologies at random. We can see this hypothesis as a baseline. If other hypotheses are not more plausible than this uniform one, we cannot expect them to provide good explanations about the production of the trails (and the ontology) at hand. The elements of matrix Q for this hypothesis are defined as follows:



(a) Top-down hypothesis: transitions to classes on lower levels are most likely.

(b) Hypothesis matrix: the transition matrix representing the top-down hypothesis.

Fig. 2. Top-down hypothesis. This figure depicts (a) the top-down hypothesis and (b) its corresponding hypothesis matrix Q that is generated from its formal definition. Darker transitions between classes represent a strong belief in these transitions, while white transitions represent a disbelief in a transition. Note that the matrix is normalized per row, hence the sum of all beliefs for each row is 1.

$$q_{i,j} = \frac{1}{|S-1|} \tag{1}$$

Top-down hypothesis. For the top-down hypothesis, we express the belief that classes that are deeper in the hierarchy (further away from the root class) than the previously edited class, are likelier to be changed next. For expressing this hypothesis, we measure the depth level of each class (the distance to the root); classes deeper in the hierarchy have larger depth levels. In this hypothesis, we have stronger beliefs in transitions to classes that have a *larger depth level* than the current class (cf. Figure 1(b)). We express this hypothesis according to the following definition with $depth_i$ and $depth_j$ representing the depth-levels of the corresponding classes s_i and s_j.

$$q_{i,j} = \begin{cases} 1, & \text{if } depth_i < depth_j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Bottom-up hypothesis. Analogously to the top-down hypothesis, this hypothesis believes that classes that are closer to the root class (i.e., they have lower depth levels) than the previously edited class, are likelier to be changed next (cf. Figure 1(c)).

$$q_{i,j} = \begin{cases} 1, & \text{if } depth_i > depth_j, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Breadth-first hypothesis. Similar to the top-down and bottom-up hypotheses, we express the belief that classes are likelier to be changed next, if they are on the same depth levels (cf. Figure 1(d)).

$$q_{i,j} = \begin{cases} \mathit{1}, & \text{if } depth_i = depth_j, \\ \mathit{0}, & \text{otherwise.} \end{cases} \tag{4}$$

Shortest path hypothesis. With this hypothesis, we express the belief that users consecutively edit classes in an ontology that are close to each other in the class hierarchy (cf. Figure 1(f)). In detail, we look at the shortest path distances $d(i,j)$ between pairs of classes—the shorter the distance, the stronger we believe in the corresponding transition. To invert the shortest path length, we subtract it from the diameter $\max_{x,y}(d(x,y))$ of the whole hierarchy.

$$q_{i,j} = \max_{x,y}(d(x,y)) - d(i,j) \tag{5}$$

Hierarchy hypothesis. The hierarchy hypothesis represents our belief that users edit classes along the hierarchical structure of the ontology (i.e., *isA* links). In particular, the next edit operation is likelier to occur on close relatives than on relatives that are further away (cf. Figure 1(e)). This hypothesis has the following weight initialization of our belief matrix:

$$q_{i,j} = \begin{cases} \mathit{4}, & \text{if } d(i,j) = 1 \text{ and } depth_i \neq depth_j, \\ \mathit{3}, & \text{if } d(i,j) = 2 \text{ and } depth_i = depth_j \text{ and } check\_siblings(i,j) > 0, \\ \mathit{2}, & \text{if } d(i,j) = 4 \text{ and } depth_i = depth_j \text{ and } check\_cousins(i,j) > 0, \\ \mathit{1}, & \text{if } sp(i,j) = |depth_i - depth_j|, \\ \mathit{0}, & \text{otherwise.} \end{cases} \tag{6}$$

Where $sp(i,j)$ is the shortest path between pairs (i,j). It holds that $check\_siblings(i,j) = |parents(i) \cap parents(j)|$ and $check\_cousins(i,j) = |grandparents(i) \cap grandparents(j)|$. Hence, both functions are larger than zero, if classes i and j share at least one parent or grandparent, respectively.

Connectivity hypothesis. In this hypothesis, we believe that the next edit operation will likelier occur on a class that is better connected in the class hierarchy. We define the *connectivity level* of a class as the number of *isA* relationships a class has to and from other classes. We represent the connectivity level of class j as k_j. The higher the connectivity level of a class, the higher our belief in a given transition (cf. Figure 1(g)). Note that for this hypothesis, each row of Q is the same—it can be seen as a zero-order Markov chain hypothesis that is weighted by the connectivity of nodes.

$$q_{i,j} = k_j \tag{7}$$

Similarity hypothesis. In this hypothesis, we believe that transitions between similar classes are likelier to occur than between less similar classes (cf. Figure 1(h)). To calculate the similarity between classes i and j, we first generate *tf-idf* vectors, v_i and v_j, consisting of the values of the annotation properties corresponding to the label of a class, and the textual definition. Using these *tf-idf* vectors, we compute the cosine similarity between classes.

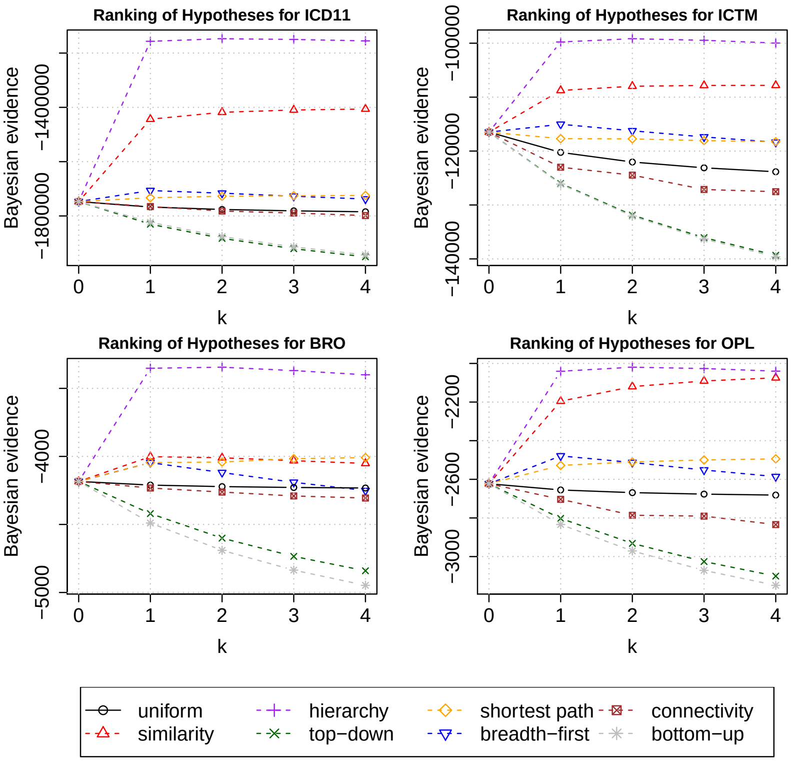


Fig. 3. Hypotheses ranking. Results for comparing hypotheses for the four datasets using HypTrails. The x-axes represent the hypothesis weighting factor k representing the "strength" of our belief in a hypothesis. In general, the stronger we believe in a hypothesis (i.e., the higher we set k), the less we expect to see transitions opposing the parametric beliefs of the corresponding hypothesis. The y-axes depict the Bayesian evidences. The higher the evidence for a given hypothesis, the better it is suited for describing the production of the extracted human edit trails (see Section 3).

$$q_{i,j} = cos\_sim(v_i, v_j) \tag{8}$$

$cos\_sim(v_i, v_j)$ is the cosine similarity between the *tf-idf* vectors of the property values corresponding to the labels and textual definitions of classes i and j.

5 Results

By applying HypTrails, we are able to gain insights into the relative plausibility of the hypotheses of interest based on the empirical data at hand. We illustrate the results in Figure 3. As mentioned in Section 3, we can compare the plausibility of hypotheses by comparing their marginal likelihoods—the higher, the more plausible. The hypothesis weighting factor k describes the "strength" of our belief in a given hypothesis; for fairness, we compare the plausibility of hypotheses by comparing their Bayesian evidences for the same values of k. For tractability, we report and interpret results for $0 <= k <= 4$; for higher values of k the results might slightly vary. Next, we highlight the main results for each ontology-engineering project (see Table 2 for a comparison of all hypotheses and datasets). We thoroughly discuss the results in Section 6.

International Classification of Diseases (ICD-11). The results for ICD-11, our biggest dataset, are depicted in the top-left part of Figure 3. The top-down and bottom-up hypotheses indicate lower evidences than the uniform hypothesis, suggesting that users are likelier to randomly change classes in the ontology than strictly follow a top-down or bottom-up approach. The connectivity hypothesis starts out to be nearly as plausible as the uniform hypothesis, but looses in Bayesian evidence faster with increasing k. The breadth-first and shortest-path hypotheses indicate higher evidences than the uniform hypothesis for our $k > 0$ at interest and thus, seem to be plausible explanations for the creation of the given human edit trails. Clearly, for ICD-11, the hierarchy hypothesis represents the most plausible explanation for the production of the trails, and thus the ontology at hand, followed by the similarity hypothesis.

International Classification of Traditional Medicine (ICTM). Similarly to ICD-11, the top-down, bottom-up and connectivity hypotheses exhibit lower evidences than the uniform hypothesis for all analyzed values of $k > 0$ (see top-right part of Figure 3). According to our experiments, the most plausible hypothesis for explaining the production of the edit trails of ICTM is the hierarchy hypothesis as it exhibits the highest Bayesian evidences for all $k > 0$. Further, the similarity hypothesis, as well as the breadth-first and shortest path hypotheses, are also better suited for describing the production of the human edit trails in ontology-engineering projects than the uniform hypothesis. For $k > 2$, we can also observe that the shortest-path hypothesis is increasing in plausibility and takes over the breadth-first hypothesis at $k = 4$.

Biomedical Resource Ontology (BRO). For BRO, the hypothesis with the highest Bayesian evidences for $k > 0$ is, again, the hierarchy hypothesis. Similarly to ICTM, the connectivity, top-down and bottom-up hypotheses are less plausible for explaining the

Table 2. Results. The table depicts the relative ranking of each hypothesis for the corresponding datasets at $k = 4$. The best performing hypotheses are highlighted bold-face. If a hypothesis is less likely to explain the production of the corresponding edit trails than the uniform hypothesis, we have marked them with "-" for the corresponding dataset.

	ICD-11	ICTM	BRO	OPL
Hierarchy Hypothesis	**1**	**1**	**1**	**1**
Similarity Hypothesis	2	2	3	2
Shortest Path Hypothesis	3	3	2	3
Breadth-First Hypothesis	4	4	-	4
Uniform Hypothesis	5	5	4	5
Connectivity Hypothesis	-	-	-	-
Bottom-Up Hypothesis	-	-	-	-
Top-Down Hypothesis	-	-	-	-

production of the human edit trails in ontology-engineering projects than the uniform hypothesis. In contrast to ICD-11 and ICTM, the similarity hypothesis is less likely to be a plausible explanation for the trails than the shortest path hypotheses. Further, the shortest path hypothesis gains evidence with growing k, while the breadth-first hypothesis drops below the uniform hypothesis at $k = 4$.

Ontology for Parasite Lifecycle (OPL). Similarly to all other projects, the most plausible hypothesis for explaining the production of the trails at hand for OPL is the hierarchy hypothesis, followed by the similarity hypothesis (especially for higher k). The top-down, bottom-up and connectivity hypotheses are again, less plausible than the uniform hypothesis at $k > 0$. Analogously to ICTM, the breadth-first and shortest path hypotheses are more plausible for explaining the creation of the human edit trails than the uniform hypothesis, and switch ranks with growing k.

6 Discussions

The results of comparing the different hypotheses for the four datasets with HypTrails are surprisingly consistent. In all of the four ontology-engineering projects, the hierarchy hypothesis represents the most plausible hypothesis to explain the production of the human edit trails in ontology-engineering projects, and therefore the corresponding ontology at hand. The similarity hypothesis is the second most plausible hypothesis for explaining the production of the human edit trails in ontology-engineering projects for ICD-11, ICTM and OPL (at $k = 4$). The reason for the high Bayesian evidences of the similarity hypothesis is most probably due to the fact that (semantically) similar classes are usually grouped into the same parts of an ontology, hence the similarity calculations are likely to reflect our beliefs of the hierarchy hypothesis. For example, in a biomedical ontology, similar classes are grouped together as siblings or cousins, sharing at least one common parent or grandparent among them. Hence, additional adaptions to further distinguish the similarity hypothesis from the hierarchy hypothesis are warranted. In particular, we plan on investigating correlation between the similarity of classes and existing hierarchical links in future work.

In Walk et al. [23], we have been arguing that users are editing the ontology in a combined top-down and breadth-first fashion. The results of our analysis confirm the results from our exploratory analysis. In particular, the hierarchy hypothesis emphasizes transitions along top-down and breadth-first hierarchical relations (i.e., children, siblings and cousins opposed to uncles and aunts). This finding is also supported by the empirical research conducted by Vigo et al. [20], which shows that the class hierarchy is the central focus of user activity in an ontology-editing session. Users spend more than 45% of their time navigating or editing the class hierarchy, which serves as an index and external memory of the ontology. The authors have identified the class hierarchy as the central component of the user interface, which also explains very well our findings.

Thus, these observations reinforce our initial belief that the ontological hierarchy influences the selection of which class to edit next. Among other potential scenarios, this information can be leveraged by ontology-engineering tools creators to minimize the efforts required by users to create new, or edit existing content in an ontology. For example, ontology-editing tools may visually highlight the corresponding classes in the user interface, and provide keyboard shortcuts that allow for quicker and more productive editing sessions. Vigo et al. [20] also make the recommendation to place editing features close to the class hierarchy to better support the users in their editing patterns.

In our investigations, we have also identified hypotheses that were weak, and potentially not useful for the purpose of improving the user interface or editing process: the top-down, bottom-up and connectivity hypotheses are less plausible than the uniform hypothesis, meaning that randomly selecting classes to work on is likelier to produce the corresponding edit trails than specifically editing highly connected classes, or editing classes in a top-down or bottom-up fashion.

Our study also has limitations, for example, all investigated ontologies are authored with the same tool, WebProtégé (or its customizations), which may biases some of our findings. However, we believe that the bias is attenuated by the fact that the projects are completely different efforts by different teams, and they also use different customizations of the user interface. Furthermore, Rospocher et al. [10], who have analyzed the change logs of two different ontology-editing platforms (WebProtégé and a Wiki system), have come to the conclusion that users tend to edit around the hierarchy, indifferent of the tool that they used. One difficulty in overcoming this limitation is the fact that obtaining change logs for real-world projects from different platforms is almost impossible. Another limitation is the fact that HypTrails focuses on comparing the relative plausibility of hypotheses. Hence, we can say that the hierarchy hypothesis is the most plausible one for explaining the production of the edit trails at hand. However, we do not know if another hypothesis, other than the ones compared, is more plausible than the hierarchy hypothesis. For example, calculating the actual transition probabilities directly from the trails yields highest Bayesian evidences. However, understanding and interpreting this empirical “hypothesis” is very hard. Also, to be able to conduct an analysis using HypTrails, we need to have detailed change-tracking information, which WebProtégé provides, but might not be as easily obtained for other projects and tools.

7 Conclusions

In this paper, we have formally defined several hypotheses of how users edit an ontology, and systematically investigated, analyzed, and ranked these hypotheses according to their relative plausibility for describing edit trails of four real-world ontology-engineering projects using HypTrails, a coherent research approach. We have found that the hierarchical structure of an ontology exercises the strongest influence on the observed user behavior, followed by the similarity of concepts. These findings are remarkably consistent across four different real-world projects, with some minor exception for the BRO dataset. We have also discussed how these findings may be used to improve ontology-editing tools. We think that our findings represent an advancement of the empirical research on how ontologies are created, which is a field that has been chronically lacking in our community.

We believe that the insights, uncovered in this paper, into how users *actually* edit real-world ontologies, represent a great opportunity for ontology-tools builders and for project managers, who can potentially leverage this information to create user interfaces and processes that better support the editing patterns of the users.

For future work, we plan to extend our set of formally defined hypotheses by including theories on how users edit properties (current work only considers class-based trails) and include different types of relationships for the analyses presented in this paper. In particular, studying individual (clustered) user behavior to automatically detect subsets of users that behave differently to other subsets of users represents a very promising opportunity for future work. On the longer term, we would like to create a recommendation module for ontology-editing tools, which would be informed by the editing patterns that we identify through our empirical research. We believe that the recommendation module and an adapted user interface will vastly improve the editing experience of the users.

Acknowledgements This work is supported in part by grants GM086587 and GM103316 from NIH, and grant STR1191/2-1 from the German Research Foundation (DFG).

References

1. An, J., Quercia, D., Crowcroft, J.: Partisan sharing: Facebook evidence and societal consequences. In: Conference on Online Social Networks. pp. 13–24. ACM (2014)
2. Beydoun, G.: Formal concept analysis for an e-learning semantic web. Expert Systems with Applications 36(8), 10952–10961 (2009)
3. Borges, J., Levene, M.: Evaluating Variable-Length Markov Chain Models for Analysis of User Web Navigation Sessions. IEEE Transactions on Knowledge and Data Engineering 19(4), 441–452 (Apr 2007), `http://dx.doi.org/10.1109/TKDE.2007.1012`
4. Chi, E.H., Pirolli, P., Pitkow, J.: The scent of a site: A system for analyzing and predicting information scent, usage, and usability of a web site. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems. pp. 161–168. ACM (2000)
5. Falconer, S.M., Tudorache, T., Noy, N.F.: An analysis of collaborative patterns in large-scale ontology development projects. In: K-CAP. pp. 25–32. ACM (2011)
6. Huberman, B.A., Pirolli, P.L.T., Pitkow, J.E., Lukose, R.M.: Strong Regularities in World Wide Web Surfing. Science 280(5360), 95–97 (Mar 1998), `http://www.sciencemag.org/content/280/5360/95.abstract`
7. Kass, R.E., Raftery, A.E.: Bayes factors. Journal of the American Statistical Association 90(430), 773–795 (1995)
8. Parikh, P., Zheng, J., Logan-Klumpler, F.J., Stoeckert Jr, C.J., Louis, C., Topalis, P., Protasio, A.V., Sheth, A.P., Carrington, M., Berriman, M., et al.: The Ontology for Parasite Lifecycle (OPL): towards a consistent vocabulary of lifecycle stages in parasitic organisms. J. Biomedical Semantics 3, 5 (2012)
9. Pesquita, C., Couto, F.M.: Predicting the Extension of Biomedical Ontologies. PLoS Comput Biol 8(9), e1002630 (09 2012)
10. Rospocher, M., Tudorache, T., Musen, M.A.: Investigating Collaboration Dynamics in Different Ontology Development Environments. In: Knowledge Science, Engineering and Management - 7th International Conference, KSEM 2014, Sibiu, Romania, October 16-18, 2014. Proceedings. pp. 302–313 (2014)
11. Singer, P., Helic, D., Hotho, A., Strohmaier, M.: HypTrails: A Bayesian Approach for Comparing Hypotheses about Human Trails on the Web. In: International Conference on World Wide Web (2015)
12. Singer, P., Helic, D., Taraghi, B., Strohmaier, M.: Detecting Memory and Structure in Human Navigation Patterns Using Markov Chain Models of Varying Order. PloS one 9(7), e102070 (2014)

13. Singla, A., White, R., Huang, J.: Studying trailfinding algorithms for enhanced web search. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 443–450. ACM (2010)
14. Strohmaier, M., Walk, S., Pöschko, J., Lamprecht, D., Tudorache, T., Nyulas, C., Musen, M.A., Noy, N.F.: How Ontologies are Made: Studying the Hidden Social Dynamics Behind Collaborative Ontology Engineering Projects. Web Semantics: Science, Services and Agents on the World Wide Web 20(0) (2013)
15. Tenenbaum, J.D., Whetzel, P.L., Anderson, K., Borromeo, C.D., Dinov, I.D., Gabriel, D., Kirschner, B.A., Mirel, B., Morris, T.D., Noy, N.F., Nyulas, C., Rubenson, D., Saxman, P.R., Singh, H., Whelan, N., Wright, Z., Athey, B.D., Becich, M.J., Ginsburg, G.S., Musen, M.A., Smith, K.A., Tarantal, A.F., Rubin, D.L., Lyster, P.: The Biomedical Resource Ontology (BRO) to enable resource discovery in clinical and translational research. Journal of Biomedical Informatics 44(1), 137–145 (February 2011)
16. Tudorache, T., Falconer, S., Nyulas, C., Noy, N.F., Musen, M.A.: Will semantic web technologies work for the development of ICD-11. In: The Semantic Web–ISWC 2010, pp. 257–272. Springer (2010)
17. Tudorache, T., Nyulas, C., Noy, N.F., Musen, M.A.: WebProtégé: A Distributed Ontology Editor and Knowledge Acquisition Tool for the Web. Semantic Web Journal 11-165 (2011)
18. Tufts, P.: Use of web usage trail data to identify related links (Feb 10 2004), `https://www.google.com/patents/US6691163`, uS Patent 6,691,163
19. Van Laere, S., Buyl, R., Nyssen, M.: A Method for Detecting Behavior-Based User Profiles in Collaborative Ontology Engineering. In: On the Move to Meaningful Internet Systems: OTM 2014 Conferences. pp. 657–673. Springer (2014)
20. Vigo, M., Jay, C., Stevens, R.: Constructing Conceptual Knowledge Artefacts: Activity Patterns in the Ontology Authoring Process. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, Seoul, Republic of Korea, April 18-23, 2015. pp. 3385–3394 (2015)
21. Walk, S., Singer, P., Strohmaier, M.: Sequential Action Patterns in Collaborative Ontology-Engineering Projects: A Case-Study in the Biomedical Domain. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 1349–1358. ACM (2014)
22. Walk, S., Singer, P., Strohmaier, M., Helic, D., Noy, N.F., Musen, M.A.: Sequential Usage Patterns in Collaborative Ontology-Engineering Projects. arXiv preprint arXiv:1403.1070 (2014)
23. Walk, S., Singer, P., Strohmaier, M., Tudorache, T., Musen, M.A., Noy, N.F.: Discovering beaten paths in collaborative ontology-engineering projects using markov chains. Journal of biomedical informatics 51, 254–271 (2014)
24. Wang, H., Tudorache, T., Dou, D., Noy, N.F., Musen, M.A.: Analysis of User Editing Patterns in Ontology Development Projects. In: On the Move to Meaningful Internet Systems: OTM 2013 Conferences. pp. 470–487. Springer (2013)
25. West, R., Leskovec, J.: Human Wayfinding in Information Networks. In: International Conference on World Wide Web. pp. 619–628. ACM (2012), `http://doi.acm.org/10.1145/2187836.2187920`
26. White, R.W., Huang, J.: Assessing the scenic route: measuring the value of search trails in web logs. In: Conference on Research and Development in Information Retrieval. pp. 587–594. ACM (2010)

4 Conclusions

Collaborative ontology engineering—a young and still mostly unexplored field of research—naturally emerged over the course of the last years due to the increasing complexity of structured knowledge representations. Particularly in the biomedical domain, where highly specialized ontologies are continuously extended to increase and complement the domain specific coverage, analyzing and modeling ongoing collaborative processes represents a very important first step towards better tool support and, hopefully, overall improved ontologies in general. Until today, evaluating an ontology involves assessing the quality of the developed ontology for a given task or by comparing it to a golden standard. However, when collaboratively engineering such highly specialized structured data representations in a distributed manner, it is important to not only evaluate the resulting ontology but also to consider and manage the encompassing social and engineering processes for quality assessment. Several empirical studies contained in this thesis indicate that contributors of collaborative ontology-engineering projects exhibit regularities and (sequential) patterns while adding or editing the structured knowledge base. In addition to investigating and interpreting these patterns, I have presented first analyses that showcase the potential of modeling and leveraging these sequential patterns to predict different aspects about future actions of users. Finally, to evaluate the results of the empirical analyses, I have formulated and compared hypotheses—beliefs, inferred from previous empirical analyses—about how users collaboratively engineer ontologies using HypTrails [Singer et al., 2015].

Before discussing limitations in Section 4.3 and future work in Section 4.4, I summarize the results and the contributions of this thesis in Section 4.1 and elaborate on the implications and potential applications of the presented analyses in Section 4.2.

4.1 Results and Contributions

To conclude and summarize the results of this work, I provide answers to the research questions defined in Section 1.4.

How can we extract and identify edit patterns in collaborative ontology-engineering projects?

Traditionally, evaluating the quality of an ontology is mainly based on assessing the utility of different aspects of the resulting ontology either for performing specific tasks or by comparison to a golden standard. However, given the increasing complexity of the data structures and the fact that large-scale ontologies, particularly in the biomedical domain, are created in a collaborative fashion, additional factors have to be considered for proper quality assessment of an ontology. Similar to traditional software development processes, the engineering processes of an ontology have to be managed and maintained and contain valuable information that should be considered for quality assessment purposes in various ways.

Hence, I present an article [Walk et al., 2015b], which describes the process of extracting sequential usage patterns from the change-logs of collaborative ontology-engineering projects. Further, the article showcases results for a large-scale collaborative ontology-engineering project and describes how to interpret the obtained results. Analyzing user behavior represents a very important first step towards a better understanding of collaborative ontology-engineering processes and hopefully towards better ontology tools and easier maintainable ontologies in general.

Do edit patterns in collaborative ontology-engineering projects exist and, if so, how do they look like?

In this thesis several empirical studies [Walk et al., 2014a,b] are presented, which not only investigate if and to what extent regularities and sequential patterns can be detected and leveraged in the change-logs of collaborative ontology-engineering projects, but also analyze the structure of these patterns in the context of different projects individually.

In Walk et al. [2014b], I investigate user behavior in collaborative ontology-engineering projects by investigating the transition matrices of fitted first-order Markov chain models. Subsequently, I was able to uncover that users are influenced by the ontological structure and proximity between classes when editing content. There also appear to be different roles of users, which can be identified by analyzing and interpreting the fitted transition matrices. Further, users appear to concentrate their edit sessions on specific edit actions and conduct their work in micro-workflows.

I also present a case study [Walk et al., 2014a] of collaborative ontology-engineering projects from the biomedical domain, which describes if and to what extent sequential usage patterns can be leveraged for predicting aspects of future actions of contributors. The results of this case study not only indicate that higher-order Markov chain models can be identified within the change-logs of collaborative ontology-engineering projects but also demonstrate that these models can be leveraged for prediction tasks reasonably well.

More generally speaking, by fitting Markov chain models on the edit sequences of contributors from collaborative ontology-engineering projects and detecting the appropriate order of these Markov chains, I was able to show (i) that users exhibit regularities and patterns in their edit sequences, (ii) that the patterns themselves already contain valuable information and (iii) that the patterns can be leveraged to predict different aspects about future changes of users.

How can we explain sequential edit patterns in collaborative ontology-engineering projects?

After manually investigating and inspecting the regularities and different sequential patterns identified in the empirical analyses presented in this thesis, the third research question tackles the problem of explaining the obtained patterns by formulating and comparing different hypotheses, representing beliefs in certain behaviors, about how users create ontologies.

Hence, colleagues and I have devised a total of 8 hypotheses, all inferred from empirical insights obtained in Walk et al. [2015b, 2014b,a], about how users actually engineer ontologies. The analysis presented in Walk et al. [2015a] makes use of the HypTrails framework, which first computes Bayesian evidences—representing the plausibility of a hypothesis to describe the production of the empirical data at hand—for each hypothesis and then uses these evidences for creating a relative ranking. The generated hypotheses were tested on four different collaborative ontology-engineering projects from the biomedical domain and show consistent results across the datasets. In particular, hypotheses that take the ontological structure of the datasets (in terms of *isA* relationships) into account, best describe the creation of the change-logs, and hence the creation of the corresponding patterns and ontologies.

4.2 Implications of this Work

The empirical analyses presented in this work represent a very first stepping stone for researchers and practitioners towards a better understanding about and an improvement of various aspects of collaborative ontology-engineering projects. Hence, I strongly believe that the results presented in this thesis will benefit future research to gather new insights into the intricate and ongoing dynamic processes that occur when users collaboratively engineer an ontology.

A method for analyzing edit patterns. The detailed description and discussion of the results presented in Walk et al. [2015b] enable researchers interested in investigating regularities and sequential patterns in their own projects. Additionally, the process description further highlights and elaborates on potential deviations due to different setups and requirements for other ontology-engineering projects, including differences regarding the interpretation and identification of such patterns. For example, when investigating transitions and patterns in the user-interface of the tool used to engineer the underlying ontology, the state space (e.g., tabs of the interface) might be different or a different level of granularity for the analysis is desired.

Empirical insights. Even when limiting the analysis only on investigating the transition matrices of fitted first-order Markov chain models, new insights can be gained, as shown in Walk et al. [2014b]. For example, by manually inspecting such transition matrices, user-roles and common edit-workflows can be identified. In that regard, project administrators and contributors are provided with potentially actionable information, allowing them to make informed decision about changes and adaptions to the engineering process to ultimately better reflect to ongoing social dynamics that occur in such projects.

Future ontology-engineering tool support. Additionally, when modeling sequential patterns using Markov chains, the resulting models can be used to predict various aspects—depending on the sequential patterns under investigation—of future actions in collaborative ontology-engineering projects. For example, when modeling transitions between user interface sections, ontology tool developers could use these models to dynamically adapt the user-interface, minimizing the amounts of clicks (and thus effort) required to add or edit the predicted/desired content. Hence, analyzing and modeling sequential patterns in collaborative ontology-engineering projects have become a useful addition to the analysis process, especially when combined and used to improve existing ontology engineering tools or processes.

Future ontology-engineering guidelines. Further, the empirical analyses presented in this work have led to a number of potential beliefs, directly inferred from manual inspections, about how users develop and maintain ontologies. These beliefs or hypotheses are formulated and represented as first-order Markov chains and ranked according to their (relative) plausibility to explain the creation of the corresponding empirical data at hand, using HypTrails [Walk et al., 2015a]. The results of this analysis suggest that users are influenced by the ontological structure when editing content in collaborative ontology-engineering projects. However, the presented analysis only represents a very first step towards a better understanding of how users collaboratively edit ontologies. By using HypTrails and further expanding on the presented analysis, it is possible to analyze if users follow specific guidelines when engineering an ontology and if not, where and how they deviate from the suggested approach.

Finally, the analyses presented in this thesis are all conducted in the context of collaborative ontology-engineering projects, but are not necessarily limited to this specific field of application. For example, it is also possible to model sequential patterns by applying Markov chains to gain new insights into how users interact with any given tool or website, as long as the corresponding edit or interaction sequences can be extracted.

4.3 Limitations

The analyses conducted in this thesis come with certain limitations that I want to discuss next.

Generality of empirical findings. All the empirical analyses of this thesis were conducted on the change-logs of collaborative ontology-engineering projects. All investigated datasets were created using Protégé or derivatives of Protégé, such as WebProtégé or Collaborative Protégé. To generalize the empirical findings of this thesis, further analyses are necessary, which not only include additional ontology tools, such as OntoStudio or OntoEdit, but also investigate and discuss the commonalities and differences between the results of these tools. To that end, I provide a detailed description in Walk et al. [2015b] on how to interpret the obtained results and how the analyses can be modified to account for different tools or settings.

Availability of datasets. For all analyses presented in this work, a very detailed log of changes is required. Each entry in the change-logs has to be mapped unambiguously to the corresponding users and entities of the ontology. Such highly detailed logs are rarely publicly available for collaborative ontology-engineering projects and limit the availability of datasets to investigate. If such logs are available, the findings presented in this thesis demonstrate how they can be used to gather new insights and how these insights can be used, for example, for predicting aspects of future actions of contributors.

Limited interaction data. The types of interactions that have been investigated in the sequential pattern analyses represent another limitation. Due to a lack of click-data, only interactions, which generate additional entries in the change-logs, are considered. In some cases, this lack of additional information might lead to biased interpretations. For example, click-based data might yield different patterns for the investigation of transitions between sections of the user-interface. For the analyses presented in this thesis, I was mainly interested in all interactions between users and the underlying structured data, which actually modify parts of the ontology. This is particularly useful when studying the impact of the ontology tool on the observed interactions. However, future work should further expand on this limitation and explore additional tools with more fine-grained change-logs, which also include click-based data.

Generality of tested hypotheses. I have used the HypTrails framework to determine which of the tested hypotheses about how users collaboratively edit an ontology best describes the creation of the corresponding empirical change-logs at hand. However, when using HypTrails it is only possible to determine the most plausible hypothesis for given empirical data out of the set of tested hypotheses. All tested hypotheses are inferred from previous empirical analyses, however, it is still possible that there exists a hypothesis that is more plausible for describing the logs of changes than the ones formulated in Walk et al. [2015a].

First-order Markov models for hypotheses. Further, HypTrails uses first-order Markov chains to model edit sequences. As a consequence, hypotheses that would require a larger memory (i.e., second-order Markov chain models or higher) can not be investigated with HypTrails without further adaptions. In the light of these limitations, the tested hypotheses have generally been chosen and formulated carefully to alleviate the impact of this limitation by concentrating on the definition of single state transitions rather than the modeling of processes containing interdependent state transitions.

4.4 Future Work

In the last section of this thesis I want to highlight potential future work.

Evaluating the engineering process. The analyses presented in this thesis represent a very first stepping stone towards bridging the gap between a better understanding of the intricate and ongoing processes in collaborative ontology-engineering projects and using these newly obtained results to improve the overall quality of the resulting ontology. In particular, the study of sequential usage patterns and hypotheses about how users collaboratively create ontologies already provide actionable results for project administrators and contributors. However, given that there exist many guidelines and best practices for (collaboratively) engineering an ontology (see Section 2.1), I am particularly interested in analyzing and modeling these as (higher-order Markov chain) hypotheses to determine which of these guidelines (if any at all) best describes how users actually create ontologies “in the wild”. Further, it might be possible to infer changes or adaptions for existing best practices, which might or might not differ for each project or domain individually.

Analyzing ontology-engineering tools. When investigating regularities and sequential patterns in the interaction logs between users and the ontology tool used to create an ontology, this thesis has shown that there is potential that can be leveraged to improve the design of the used tools. Hence, further analyses and studies are warranted that measure this potential for improvement, for example by implementing and evaluating adaptive user interfaces. Using the Markov chain framework, it is possible to conduct these analyses by analyzing the logs of changes (or click-based logs), without the explicit need for dedicated and continuous user studies.

Expanding on empirical analyses. For the empirical analyses I have concentrated my efforts on the identification and modeling of sequential usage patterns in collaborative ontology-engineering projects. However, there is still the open question if and to what extent the fitted Markov chain models can be used to predict different aspects of future actions of users in live-lab environments. One particular approach should involve not only the comparison of existing recommender approaches with the fitted Markov chain models, but also hybrid approaches that make use of the sequential information stored inside the Markov chain models for the prediction task. Additionally, one particular problem of large-scale ontology-engineering projects is related to the vast amount of classes and content available in these ontologies and the retrieval of the correct ones to work on. To that end, it would be interesting to further investigate if the fitted Markov chain models can be used reasonably well—either alone or in combination with other recommender approaches—to identify and suggest classes to users, eliminating the need to manually search and identify content to work on. Further, to mitigate the limitation regarding the generality of the obtained results, future work should also involve the analysis of additional collaborative ontology-engineering projects, such as Wikidata or datasets included in the Linked Open Data cloud, and ontology-engineering tools (e.g., OntoStudio or OntoEdit). In that regard, additional analyses to discern attributes and aspects of collaborative ontology-engineering projects from traditional online peer production systems such as Wikipedia are warranted.

In general, it is the hope of the author of this work that the presented results are used as a foundation for future work to further investigate the complex social dynamics that occur when users collaboratively engineer an ontology. In particular, the analyses and results presented in this thesis can be seen as a very first stepping stone towards improved ontology tool support and easier maintainable ontologies. It is the long-term vision of the author of this thesis that this work will serve as a very first stepping stone towards further advancing ontology evaluation techniques to include the ongoing complex social interactions in collaborative ontology-engineering projects.

List of Figures

List of Tables

Bibliography

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA. IEEE Computer Society.

Auer, S. (2006). The rapidowl methodology–towards agile knowledge engineering. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on*, pages 352–357. IEEE.

Auer, S., Dietzold, S., and Riechert, T. (2006). OntoWiki–A Tool for Social, Semantic Collaboration. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, volume LNCS 4273, Athens, GA. Springer.

Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, us edition.

Beck, K. (2000). *Extreme programming explained: embrace change*. Addison-Wesley Professional.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved Feb 02, 2012, from `http://www.agilemanifesto.org/`.

Bettini, C., Wang, X. S., and Jajodia, S. (1996). Testing complex temporal relationships involving multiple granularities and its application to data mining (extended abstract). In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '96, pages 68–78, New York, NY, USA. ACM.

Borgelt, C. (2005). An implementation of the fp-growth algorithm. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pages 1–5. ACM.

Borges, J. and Levene, M. (2000). Data mining of user navigation patterns. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, WEBKDD '99, pages 92–111, London, UK, UK. Springer-Verlag.

Borges, J. and Levene, M. (2007). Evaluating variable-length markov chain models for analysis of user web navigation sessions. *IEEE Trans. on Knowl. and Data Eng.*, 19(4):441–452.

Borst, W. (1997). Construction of engineering ontologies for knowledge sharing and reuse.

Brank, J., Grobelnik, M., and Mladenić, D. (2005). A Survey of Ontology Evaluation Techniques. In *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, pages 166–170.

Braun, S., Schmidt, A. P., Walter, A., Nagypal, G., and Zacharias, V. (2007). Ontology maturing: a collaborative web 2.0 approach to ontology engineering. *Ckc*, 273.

Brewster, C., Alani, H., Dasmahapatra, S., and Wilks, Y. (2004). Data Driven Ontology Evaluation. In *Proceedings of the International Conference on Language Resources and Evaluation 2004 (LREC)*, Lisbon, Portugal.

Brickley, D. and Guha, R. (2004). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands. Elsevier Science Publishers B. V.

Buehrer, G. and Chellapilla, K. (2008). A scalable pattern mining approach to web graph compression with communities. In *Proceedings of the 2008*

International Conference on Web Search and Data Mining, pages 95–106. ACM.

Cabrera, A. and Cabrera, E. F. (2002). Knowledge-Sharing Dilemmas. *Organization Studies*, 23(5):687–710.

Chierichetti, F., Kumar, R., Raghavan, P., and Sarlos, T. (2012a). Are web users really markovian? In *Proceedings of the 21st international conference on World Wide Web*, pages 609–618. ACM.

Chierichetti, F., Kumar, R., Raghavan, P., and Sarlos, T. (2012b). Are web users really markovian? In *Proceedings of the 21st international conference on World Wide Web*, pages 609–618. ACM.

Coad, P., Lefebvre, E., and DeLuca, E. (1999). *Java Modeling in Color with UML: Enterprise Components and Process.* Prentice Hall, Upper Saddle River, NJ.

Cockburn, A. (2004). *Crystal-Clear a Human-Powered Methodology for Small Teams.* Addison-Wesley Professional, first edition.

Davies, J., Sure, Y., Vrandecic, D., Pinto, S., Tempich, C., and Sure, Y. (2005). The diligent knowledge processes. *Journal of Knowledge Management*, 9(5):85–96.

De Leenheer, P., Debruyne, C., and Peeters, J. (2009). Towards social performance indicators for community-based ontology evolution. In *Proceedings of Workshop on Collaborative Construction, Management and Linking of Structured Knowledge (CK2008), collocated with the 8th International Semantic Web Conference (ISWC 2009), Washington DC, CEUR-WS*, pages 38–48. Citeseer.

Debruyne, C. and Nijs, N. (2013). Using a reputation framework to identify community leaders in ontology engineering. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pages 677–684. Springer.

Debruyne, C., Reul, Q., and Meersman, R. (2010). Gospl: Grounding ontologies with social processes and natural language. In *2010 Seventh*

International Conference on Information Technology, pages 1255–1256. IEEE.

Dellschaft, K., Engelbrecht, H., Barreto, J. M., Rutenbeck, S., and Staab, S. (2008). Cicero: Tracking design rationale in collaborative ontology engineering. In Bechhofer, S., Hauswirth, M., Hoffmann, J., and Koubarakis, M., editors, *Proceedings of the 5th European Semantic Web Conference*, volume 5021 of *Lecture Notes in Computer Science*, pages 782–786. Springer.

Deshpande, M. and Karypis, G. (2004). Selective markov models for predicting web page accesses. *ACM Trans. Internet Technol.*, 4(2):163–184.

Di Francescomarino, C., Ghidini, C., and Rospocher, M. (2014). Evaluating Wiki Collaborative Features in Ontology Authoring. *IEEE Transactions on Knowledge & Data Engineering*, (12):2997–3011.

Domingue, J. (1998). Tadzebao and webonto: Discussing, browsing, and editing ontologies on the web.

Falconer, S. M., Tudorache, T., and Noy, N. F. (2011). An analysis of collaborative patterns in large-scale ontology development projects. In Musen, M. A. and Corcho, O., editors, *K-CAP*, pages 25–32. ACM.

Farquhar, A., Fikes, R., and Rice, J. (1997). The ontolingua server: A tool for collaborative ontology construction. *International journal of human-computer studies*, 46(6):707–727.

Fernández-López, M., Gómez-Pérez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering.

Garofalakis, M. N., Rastogi, R., and Shim, K. (1999). Spirit: Sequential pattern mining with regular expression constraints. In *VLDB*, volume 99, pages 7–10.

Ghidini, C., Kump, B., Lindstaedt, S., Mahbub, N., Pammer, V., Rospocher, M., and Serafini, L. (2009). MoKi: The Enterprise Modelling Wiki. In Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., and Simperl, E. P. B., editors, *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications 2009*, pages 831–835, Berlin, Heidelberg. Springer.

Gil, Y., Kapoor, D., Markham, R., and Ratnakar, V. (2015). The provenance bee wiki: Tracking the growth of semantic wiki communities. In *Proceedings of the 8th International Conference on Knowledge Capture*, page 10. ACM.

Gil, Y., Knight, A., Zhang, K., Zhang, L., and Sethi, R. J. (2013). An initial analysis of semantic wikis. In *IUI Companion*, pages 109–110.

Gil, Y. and Ratnakar, V. (2013). Knowledge capture in the wild: a perspective from semantic wiki communities. In *K-CAP*, pages 49–56.

Gomez, A., Fernandez, M., and Corcho, O. (2004). Ontological engineering. *2 nd Edition, Springer-Verlag*.

Gómez-Pérez, A. and Suárez-Figueroa, M. C. (2009). Scenarios for building ontology networks within the neon methodology. In *Proceedings of the fifth international conference on Knowledge capture*, pages 183–184. ACM.

Groza, T., Tudorache, T., and Dumontier, M. (2013). Commentary: State of the art and open challenges in community-driven knowledge curation. *Journal of Biomedical Informatics*, 46(1):1–4.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Grüninger, M. and Fox, M. S. (1995). Methodology for the design and evaluation of ontologies.

Haase, P. and Qi, G. (2007). An analysis of approaches to resolving inconsistencies in dl-based ontologies. In *Proceedings of International Workshop on Ontology Dynamics (IWOD'07)*.

Halfaker, A., Kittur, A., and Riedl, J. (2011). Don't bite the newbies: how reverts affect the quantity and quality of Wikipedia work. In Ortega, F. and Forte, A., editors, *Int. Sym. Wikis*, pages 163–172. ACM.

Han, J., Pei, J., and Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 1–12, New York, NY, USA. ACM.

Holsapple, C. W. and Joshi, K. D. (2002). Knowledge manipulation activities: results of a delphi study. *Information & Management*, 39(6):477–490.

Hsu, C.-M., Chen, C.-Y., Liu, B.-J., Huang, C.-C., Laio, M.-H., Lin, C.-C., and Wu, T.-L. (2007). Identification of hot regions in protein-protein interactions by sequential pattern mining. *BMC bioinformatics*, 8(Suppl 5):S8.

Keegan, B., Gergle, D., and Contractor, N. S. (2011). Hot off the wiki: dynamics, practices, and structures in Wikipedia's coverage of the Tohoku catastrophes. In Ortega, F. and Forte, A., editors, *Int. Sym. Wikis*, pages 105–113. ACM.

Kittur, A., Chi, E., Pendleton, B., Suh, B., and Mytkowicz, T. (2007). Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. *World Wide Web*, 1(2):1–9.

Kittur, A. and Kraut, R. E. (2008). Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, CSCW '08, pages 37–46, New York, NY, USA. ACM.

Kotis, K. and Vouros, G. A. (2006). Human-centered ontology engineering: The hcome methodology. *Knowledge and Information Systems*, 10(1):109–131.

Krötzsch, M., Vrandecic, D., and Völkel, M. (2006). Semantic MediaWiki. In *Proceedings of the 5th International Semantic Web Conference 2006 (ISWC 2006)*, pages 935–942. Springer.

Lam, J. (2007). *Methods for resolving inconsistencies in ontologies.* PhD thesis, University of Aberdeen.

Lamprechta, D., Strohmaier, M., Helica, D., Nyulasb, C., Tudoracheb, T., Noyb, N. F., and Musenb, M. A. (2014). Using ontologies to model human navigation behavior in information networks: A study based on wikipedia. *Semantic Web Journal*.

Lempel, R. and Moran, S. (2000). The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Comput. Netw.*, 33(1-6):387–401.

Maedche, A. and Staab, S. (2002). Measuring Similarity between Ontologies. In Gómez-Pérez, A. and Benjamins, V. R., editors, *Proceedings of the European Conference on Knowledge Acquisition and Management 2002(EKAW)*, volume 2473 of *Lecture Notes in Computer Science*, pages 251–263. Springer.

Mannila, H., Toivonen, H., and Verkamo, A. I. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289.

Markov, A. A. (2006). An example of statistical investigation of the text eugene onegin concerning the connection of samples in chains. *Science in Context*, 19(04):591–600.

Masseglia, F., Cathala, F., and Poncelet, P. (1998). The psp approach for mining sequential patterns. In *Principles of Data Mining and Knowledge Discovery*, pages 176–184. Springer.

Mika, P. and Alani, H. (2005). Ontologies are us: A unified model of social networks and semantics. In Gil, Y., Motta, E., Benjamins, R. V., and Musen, M., editors, *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, Lecture Notes in Computer Science no. 3729, pages 122–136, Galway, Ireland. Springer.

Mortensen, J. M., Musen, M. A., and Noy, N. F. (2013). Crowdsourcing the verification of relationships in biomedical ontologies. In *AMIA Annual Symposium Proceedings*, volume 2013, page 1020. American Medical Informatics Association.

Neuhaus, F., Vizedom, A., Baclawski, K., Bennett, M., Dean, M., Denny, M., Grüninger, M., Hashemi, A., Longstreth, T., Obrst, L., et al. (2013).

Towards ontology evaluation across the life cycle. *Applied Ontology*, 8(3):179–194.

Ng, R. T., Lakshmanan, L. V. S., Han, J., and Pang, A. (1998). Exploratory mining and pruning optimizations of constrained associations rules. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD '98, pages 13–24, New York, NY, USA. ACM.

Noy, N., Chugh, A., Liu, W., and Musen, M. E. (2006). A framework for ontology evolution in collaborative environments. *The Semantic Web - ISWC 2006*, pages 544–558.

Noy, N. F., McGuinness, D. L., et al. (2001). Ontology development 101: A guide to creating your first ontology.

Noy, N. F., Mortensen, J., Musen, M. A., and Alexander, P. R. (2013). Mechanical turk as an ontology engineer?: using microtasks as a component of an ontology-engineering workflow. In *Proceedings of the 5th Annual ACM Web Science Conference*, pages 262–271. ACM.

Noy, N. F. and Tudorache, T. (2008). Collaborative ontology development on the (semantic) web. In *AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering*, pages 63–68. AAAI.

Obrst, L., Ceusters, W., Mani, I., Ray, S., and Smith, B. (2007). The evaluation of ontologies. In Baker, C. J. and Cheung, K.-H., editors, *Revolutionizing Knowledge Discovery in the Life Sciences*, chapter 7, pages 139–158. Springer.

Palmer, S. R. and Felsing, J. M. (2002). *A Practical Guide to Feature-Driven Development.* Prentice Hall, Upper Saddle River, NJ.

Patel-Schneider, P. F., Hayes, P., Horrocks, I., et al. (2004). Owl web ontology language semantics and abstract syntax. *W3C recommendation*, 10.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. (2001). PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In *Proceedings of the 17th International Conference on Data Engineering*, ICDE '01, pages 215–224, Washington, DC, USA. IEEE Computer Society.

Pei, J., Han, J., Mortazavi-Asl, B., and Zhu, H. (2000). Mining access patterns efficiently from web logs. In *Knowledge Discovery and Data Mining. Current Issues and New Applications*, pages 396–407. Springer.

Perera, D., Kay, J., Koprinska, I., Yacef, K., and Zaïane, O. R. (2009). Clustering and sequential pattern mining of online collaborative learning data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(6):759–772.

Pesquita, C. and Couto, F. M. (2012). Predicting the extension of biomedical ontologies. *PLoS Comput Biol*, 8(9):e1002630.

Pinto, H. S., Tempich, C., and Staab, S. (2004). DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engingeering of oNTologies. In de Mantaras, R. L. and Saitta, L., editors, *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 393–397. IOS Press.

Pirolli, P. L. T. and Pitkow, J. E. (1999). Distributions of surfers' paths through the world wide web: Empirical characterizations. *World Wide Web*, 2(1-2):29–45.

Porzel, R. and Malaka, R. (2004). A Task-based Approach for Ontology Evaluation. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI) 2004 Workshop on Ontology Learning and Population*, Valencia, Spain. Citeseer.

Pöschko, J., Strohmaier, M., Tudorache, T., Noy, N. F., and Musen, M. A. (2012). Pragmatic analysis of crowd-based knowledge production systems with icat analytics: Visualizing changes to the icd-11 ontology. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium: Wisdom of the Crowd*, Stanford, CA, USA.

Poveda-Villalón, M., Gómez-Pérez, A., and Suárez-Figueroa, M. C. (2014). Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):7–34.

Ribeiro, B. (2014). Modeling and predicting the growth and death of membership-based websites. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 653–664, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Royce, W. W. (1970). Managing the development of large software systems. In *Proceedings of the IEEE (Institute of Electrical and Electronics Engineers) Wescon 1970*, volume 26, pages 1–9. Los Angeles.

Sabou, M., Gracia, J., Angeletou, S., d'Aquin, M., and Motta, E. (2007). Evaluating the semantic web: A task-based approach. In *ISWC/ASWC*, pages 423–437.

Sarasua, C., Simperl, E., and Noy, N. F. (2012). Crowdmap: Crowdsourcing ontology alignment with microtasks. In *The Semantic Web–ISWC 2012*, pages 525–541. Springer.

Sarawagi, S., Thomas, S., and Agrawal, R. (1998). Integrating association rule mining with relational database systems: Alternatives and implications. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD '98, pages 343–354, New York, NY, USA. ACM.

Schaffert, S. (2006). Ikewiki: A semantic wiki for collaborative knowledge management. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on*, pages 388–396. IEEE.

Schandl, T. and Blumauer, A. (2010). Poolparty: SKOS thesaurus management utilizing linked data. *The Semantic Web: Research and Applications*, 6089:421–425.

Schreiber, G., Wielinga, B., de Hoog, R., Akkermans, H., and Van de Velde, W. (1994). Commonkads: A comprehensive methodology for kbs development. *IEEE expert*, 9(6):28–37.

Schwaber, K. (1995). SCRUM Development Process. In *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications 1995 (OOPSLA)*, pages 117–134. Citeseer.

Sen, R. and Hansen, M. (2003). Predicting a web user's next access based on log data. *Journal of Computational Graphics and Statistics*, 12(1):143–155.

Shachaf, N. (2010). Beyond vandalism: Wikipedia trolls. *Journal of Information Science; Jun2010, Vol. 36 Issue 3, p357-370, 14p, 2 Charts.*

Simperl, E. and Luczak-Rösch, M. (2014). Collaborative ontology engineering: a survey. *The Knowledge Engineering Review*, 29(01):101–131.

Singer, P., Helic, D., Hotho, A., and Strohmaier, M. (2015). Hyptrails: A bayesian approach for comparing hypotheses about human trails on the web. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1003–1013, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Singer, P., Helic, D., Taraghi, B., and Strohmaier, M. (2014). Detecting memory and structure in human navigation patterns using markov chain models of varying order. *PloS one*, 9(7):e102070.

Singer, P., Niebler, T., Strohmaier, M., and Hotho, A. (2013). Computing semantic relatedness from human navigational paths: A case study on wikipedia. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 9(4):41–70.

Solomon, J. and Wash, R. (2014). Critical mass of what? exploring community growth in wikiprojects.

Srikant, R. and Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th*

International Conference on Extending Database Technology: Advances in Database Technology, EDBT '96, pages 3–17, London, UK, UK. Springer-Verlag.

Strohmaier, M., Walk, S., Pöschko, J., Lamprecht, D., Tudorache, T., Nyulas, C., Musen, M. A., and Noy, N. F. (2013). How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects. *Web Semantics: Science, Services and Agents on the World Wide Web*, 20(0).

Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: Principles and methods. volume 25, pages 161–197.

Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., and Gangemi, A. (2012). *Ontology engineering in a networked world.* Springer Science & Business Media.

Suh, B., Convertino, G., Chi, E. H., and Pirolli, P. (2009). The singularity is not near: slowing growth of wikipedia. In *WikiSym '09: Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, pages 1–10, New York, NY, USA. ACM.

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., and Wenke, D. (2002). *OntoEdit: Collaborative ontology development for the semantic web.* Springer.

Tempich, C., Simperl, E., Luczak, M., Studer, R., and Pinto, S. H. (2007). Argumentation-based ontology engineering. *IEEE Intelligent Systems*, 22(6):52–59.

Tudorache, T., Falconer, S. M., Nyulas, C. I., Noy, N. F., and Musen, M. A. (2010). Will Semantic Web technologies work for the development of ICD-11? In *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, ISWC (In-Use), Shanghai, China. Springer.

Tudorache, T., Noy, N. F., Tu, S., and Musen, M. A. (2008). *Supporting collaborative ontology development in Protégé.* Springer.

Tudorache, T., Nyulas, C., Noy, N. F., and Musen, M. A. (2013). WebProtégé: A Distributed Ontology Editor and Knowledge Acquisition Tool for the Web. *Semantic Web Journal*, 4(1/2013):89–99.

Van Laere, S., Buyl, R., and Nyssen, M. (2014). A Method for Detecting Behavior-Based User Profiles in Collaborative Ontology Engineering. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences*, pages 657–673. Springer.

Vrandečić, D. (2012). Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 1063–1064. ACM.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Walk, S., Pöschko, J., Strohmaier, M., Andrews, K., Tudorache, T., Noy, N. F., Nyulas, C., and Musen, M. A. (2013). Pragmatix: An interactive tool for visualizing the creation process behind collaboratively engineered ontologies. *International journal on Semantic Web and information systems*, 9(1):45.

Walk, S., Singer, P., Noboa, L. E., Tudorache, T., Musen, M. A., and Strohmaier, M. (2015a). Understanding how users edit ontologies: Comparing hypotheses about four real-world projects. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, pages 551–568.

Walk, S., Singer, P., and Strohmaier, M. (2014a). Sequential action patterns in collaborative ontology-engineering projects: A case-study in the biomedical domain. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1349–1358.

Walk, S., Singer, P., Strohmaier, M., Helic, D., Noy, N. F., and Musen, M. A. (2015b). How to apply markov chains for modeling sequential edit patterns in collaborative ontology-engineering projects. *International Journal of Human-Computer Studies*, 84:51 – 66.

Walk, S., Singer, P., Strohmaier, M., Tudorache, T., Musen, M. A., and Noy, N. F. (2014b). Discovering beaten paths in collaborative ontology-engineering projects using markov chains. *Journal of Biomedical Informatics*, 51:254–271.

Walk, S. and Strohmaier, M. (2014). Characterizing and predicting activity in semantic mediawiki communities. In *Proceedings of the Third International Workshop on Semantic Web Collaborative Spaces co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, October 19, 2014*.

Wang, H., Tudorache, T., Dou, D., Noy, N. F., and Musen, M. A. (2013). Analysis of user editing patterns in ontology development projects. In *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pages 470–487. Springer.

Wang, J. T.-L., Chirn, G.-W., Marr, T. G., Shapiro, B., Shasha, D., and Zhang, K. (1994). Combinatorial pattern discovery for scientific data: Some preliminary results. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, SIGMOD '94, pages 115–125, New York, NY, USA. ACM.

Wilkinson, D. M. and Huberman, B. A. (2007). Cooperation and quality in wikipedia. In *Proceedings of the 2007 international symposium on Wikis*, WikiSym '07, pages 157–164, New York, NY, USA. ACM.

Zacharias, V. and Braun, S. (2007). SOBOLEO - Social Bookmarking and Lightweight Ontology Engineering. In *Workshop on Social and Collaborative Construction of Structured Knowledge (CKC), 16th International World Wide Web Conference 2007 (WWW 2007)*.

Zaki, M. J. (2001). Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60.

Zukerman, I., Albrecht, D. W., and Nicholson, A. E. (1999). Predicting users' requests on the www. Proceedings of the Seventh International Conference on User Modeling, pages 275–284, Secaucus, NJ, USA. Springer-Verlag New York, Inc.

Zurcher, F. W. and Randell, B. (1968). Iterative Multi-Level Modeling - A Methodology for Computer System Design. In *in Proceedings of the IFIP Congress 1968*, pages 138–142. Computer Society Press.