

Modeling Sequential Usage Patterns in Collaborative Ontology-Engineering Projects

Simon Walk^a, Philipp Singer^b, Markus Strohmaier^{c,d}, Denis Helic^b, Natasha 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 biomedical collaborative ontology-engineering projects, such as the creation of the 11th revision of the International Classification of Diseases, new methods and insights are needed 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 on the change-logs of collaborative ontology-engineering projects to extract and analyze sequential patterns. This method also allows to investigate memory and structure in human activity patterns when collaboratively creating an ontology by leveraging Markov Chain models of varying orders. We describe all necessary steps for applying the methodology to collaborative ontology-engineering projects and provide first results for the International Classification of Diseases in its 11th revision. Furthermore, we show that the collected sequential-patterns provide actionable information for community- and project-managers to monitor, coordinate and dynamically adapt to the natural development processes that occur when collaboratively engineering an ontology. We hope that the adaption of the presented methodology will spur a new line of ontology-development tools and evaluation-techniques, which concentrate on the interactive nature of the collaborative ontology-engineering process.

Keywords: Collaborative Ontology-Engineering, Markov Chain, sequential patterns, collaboration, ontology editor, user-interface

1. Introduction

With the increasing popularity of structured data within the last years many large-scale projects were launched with a set goal to collaboratively engineer ontologies. For example, the World Health Organization (WHO) is maintaining the collaborative online-development of the new revision of the International Classification of Diseases (ICD), which represents a very important classification scheme that is used in many countries around the world. Wikidata¹, another collaborative ontology-engineering project initiated by the Wikimedia Foundation², is gathering structured data in multiple languages to link to and between Wikipedia and its different language editions, also including links to and between other projects from the Wikimedia Foundation. To anticipate the new requirements attached to this collaborative approach, researchers have analyzed and developed new ontology-engineering tools, such as Collaborative Protégé and WebProtégé [1, 2], which not only provide a collaborative environment to engineer ontologies but also 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. For collaborative ontology-engineering projects with large numbers of involved users in particular, researchers will have to find new ways of anticipating yet unknown problems while simultaneously reinforcing the benefits attached to the collaborative nature of these projects. Especially when keeping in mind that engineering an ontology by itself already represents a complex task, which 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.

When users collaboratively engineer ontologies several user actions succeed each other. For example, a user sequentially changes properties of a concept in the ontology which results in a sequence of properties (and concepts) the user works on. Better understanding such sequential processes can severely help system designers to e.g., increase the quality of an ontology or contributor satisfaction. To come back to our previous example, if we better understand the process better of how users sequentially edit properties of concepts, we can e.g., recommend users the next property they potentially might want to edit or the other way around, steer users away from their typical behavior in order to also cover niche parts of the ontology. From litera-

¹<http://www.wikidata.org>

²<http://wikimediafoundation.org>

ture we already know, that sequential patterns of human actions usually can be well predicted. E.g., Song et al. [3] showed that human mobility patterns are predictable and the authors also hypothesize that all human activities contain certain regularities that can be detected which might also apply to our ontology usage sequences of interest. Hence, our main goal in this paper is to find methods and techniques for getting detailed insights into these ongoing (sequential) processes of users when collaboratively engineering an ontology. We present an adaptation of a Markov Chain based methodology, originally used to detect memory (i.e., on how many previous clicks the next click of a user depends on) and structure (e.g., the identification of common sequences) in human navigational patterns through websites, to analyze such sequential-patterns in collaborative ontology-engineering projects (e.g., sequential patterns of how users sequentially change properties of concepts). This choice allows us to model human interaction paths in the change-log data provided by our three datasets.

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

- By investigating the structured change-logs of ICD-11 for sequential patterns, we want to identify a formal model (the best performing Markov chain order), which describes the ongoing processes when collaboratively engineering an ontology.
- To gain new insights into potential memory effects present in human usage activities when collaboratively working on an ontology we make use of Markov chains and estimate the best-performing model order by calculating two information criteria. For example, a first order Markov Chain model postulates that the next action taken is only dependent on the current action, while higher order chains say that the next action is dependent on a series of preceding ones – i.e., we also analyze whether the human action sequences are memoryless or whether memory effects are present.
- We are also interested whether we can predict the action that is performed next reasonably well, using a specific Markov Chain order model. By leveraging these models we are also investigating structural patterns in such sequences, such as the identification of edit-strategies (i.e., *bottom-up* or *top-down* development) that users follow while collaboratively editing the ontology.

Results: Our results indicate that the application of Markov Chains on the change-logs of collaborative ontology-engineering projects provides new and actionable insights into the processes that occur while collaboratively creating an ontology for project administrators and ontology-engineering tool developers. We show that sequential patterns of varying lengths can be extracted and used to predict the most likely state that is to occur next in the investigated project.

To the best of our knowledge, the work presented in this paper represents the first and most detailed attempt to apply Markov chains on the structured usage log of a large-scale collaborative ontology-engineering project to date.

Contributions: We provide (i) detailed descriptions of the mapping and extraction process, and (ii) conduct a first sequential-pattern analysis by applying the adapted methodology on ICD-11, representing a large-scale collaborative ontology-engineering project. Additionally, we provide (iii) brief explanations of how the gathered results can be interpreted and utilized in productive environments. Our **high-level contribution** is a novel approach of investigating and understanding ongoing processes when collaboratively engineering an ontology by making use of Markov chains to extract sequential usage patterns.

The paper is structured as follows: In section 2 we review related work. In section 3, we briefly describe and characterize the history of ICD-11, followed by a detailed explanation of the mapping process of the used Markov Chain framework, presented by Singer et al. [4], towards collaborative ontology engineering-projects. We continue with a description, presentation and discussion of the results from our sequential pattern analysis in section 4. In section 6, we summarize our findings and discuss potential implications. We conclude in section ?? and provide ideas for potential future work.

2. Related Work

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

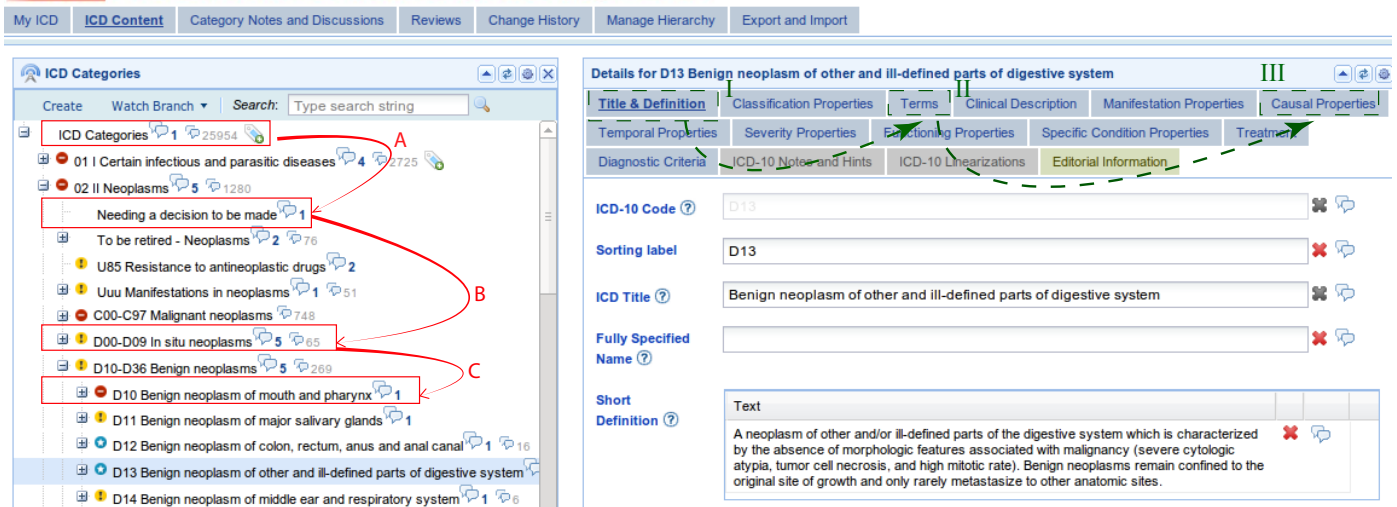
2.1. Collaborative Ontology Engineering

According to Gruber [5], Borst [6], 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.

However, the field of collaborative ontology engineering and its environment pose a new field of research with many new problems, risks and challenges that first have to be identified and can only then be dealt with. 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.

A number of tools [8, 9, 10, 11] in the field of collaborative ontology engineering were developed, specifically aiming at supporting and augmenting different aspects of the collaborative development processes of ontologies. For example, Semantic MediaWikis [12] 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.

As an ontology represents a formalized and abstract version of a specific domain, disagreements between authors on certain



User 1: A → B → C: DOWN → UP → DOWN

User 4: I → II → III: Title & Definition → Terms → Causal Properties

Figure 1: 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 4.2) for one user, while the roman numbers I – III constitute a representative sequential path for the *User-Interface Sections Path* analyses (see section 4.3) 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.

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 (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 [13, 14].

The Semantic Web community has developed a number of tools aimed at supporting the collaborative development of ontologies. For example, Semantic MediaWikis [12] and its derivatives [8, 9, 11] add semantic, ontology modeling and collaborative features to traditional MediaWiki systems.

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 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 [15].

Pöschko et al. [16], and Walk et al. [17] have created *PragmatiX*, a tool to browse an ontology and aspects of its history visually, which provides quantitative insights into the creation process, and applied it to the ICD-11 project. Strohmaier et al. [18] investigated the hidden social dynamics that take place in collaborative ontology-engineering projects from the biomedical domain and provide new metrics to quantify various aspects of the collaborative engineering processes.

Falconer et al. [19] investigated the change-logs of collaborative ontology-engineering projects, showing that users ex-

hibit specific roles, which can be used to group and classify users, when contributing to the ontology. Pesquita and Couto [20] 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 Ontology³. Strohmaier et al. [18] 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. [21] 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.

2.2. 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 [22, 23, 24, 25, 26, 27]. Specific 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 postulated in the *Markovian assumption* which states that the next state 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 [28].

³<http://www.geneontology.org>

Previously, researchers also investigated whether human navigation really is memoryless in a series of studies (e.g., [29, 25]). However, it was mostly shown that the benefit of higher orders is not enough in order to compensate the extreme high number of parameters needed and hence, the memoryless model seems to be a quite plausible abstraction (see e.g., [30, 31, 26, 27]). Recently, a study picked up on these investigations and again suggested that the Markovian assumption might be wrong [32]. However, this study did not reveal any statistical significant improvements of higher order models. This problem was solved by Singer et al. [4], who developed a framework for determining the appropriate Markov Chain order. Their studies on several navigational datasets also revealed that the memoryless model indeed seems to be a plausible abstraction as it is very difficult for higher order models to show statistically significant improvements due to the high number of parameters needed combined with shortcomings in available data. However, their work showed 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.

2.3. 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, [33] have shown that for Wikipedia and del.icio.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) exhibits little if any participation and contribution. 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 [34]) 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 [35]. 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 [35]. Wilkinson and Huberman [36] 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 [37] but also on antagonism and sabotage of said systems [38]. It has also been discovered only recently that Wikipedia editors are slowly but steadily declining

Suh et al. [39]. Therefore [40] have analyzed what impact reverts have on new editors of Wikipedia. Moreover, many publications also deal with automatic information and knowledge extraction from Wikipedia [41, 42] due to the uprising of the semantic web and open linked data.

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.

3. Materials & Methods

The dataset used in this paper to illustrate the process of identifying memory and structure in sequential usage patterns in collaborative ontology-engineering projects exhibits the following characteristics: (i) at least two users have contributed to the projects, and (ii) a structured log of changes (ChAO, see section 3.2) is available, which can be mapped onto the underlying ontology. These characteristics can be seen as the minimum requirements to allow for an application of Markov Chains onto collaborative ontology-engineering projects. In section 3.1 we will provide a brief history of ICD-11. To get a better understanding for the ChAO and how the data for the sequential pattern analysis was extracted we outline its functionality in section 3.2. To aid readers in understanding the analyses conducted in this paper and its possible implications we provide an introduction and overview of Markov Chains and the involved model selection methodology in section 3.3 for determining the appropriate order of a Markov Chain which can help us to better understand memory effects. Section 3.4 describes the extraction and generation process of the *artificial session breaks*.

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

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

3.1. The 11th Revision of the International Classification of Diseases (ICD-11)

ICD-11⁴, 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

⁴<http://www.who.int/classifications/icd/ICDRevision/>

to inform policy makers. As a result, ICD is an essential resource for health care all over the world. ICD traces its origins to the 19th century and has since been revised at regular intervals. The current in-use version, ICD-10, the 10th revision of the ICD, contains more than 20,000 terms. 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.

3.2. The Change-Logs

The ontology used for the demonstration of the Markov Chain-based sequential usage pattern analysis is created using 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 underlying ontology.

Protégé and all of its derivatives use the **Change and Annotation Ontology (ChAO)** [43] to represent changes. Change types are ontology classes in ChAO and changes in the domain ontology are instances of these classes (Figure 2). Similarly, notes that users attach to classes or threaded user discussions are also stored in ChAO. In fact, ChAO records two types of changes, so-called “Atomic” and “Composite” changes.

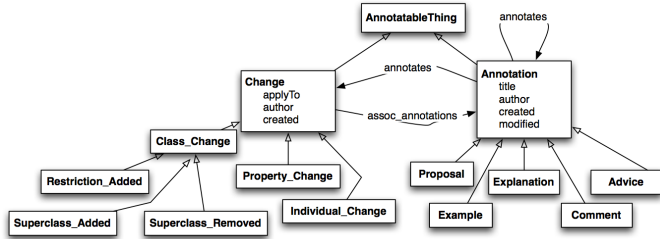


Figure 2: Excerpt of the Change and Annotation Ontology (ChAO) used by Protégé [43]. Boxes represent classes and lines represent relationships.

“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 changes stored in the ChAO, which are always actual changes to the ontology (as opposed to proposed changes).

As *iCAT* users collaborate in developing their ontology, many use the discussion features of the tools to add comments and annotations to the classes in the ontology. These annotations are essential for collaboration as they can be attached

to concepts, for example as *Explanations*, to justify certain changes or as *Comments* to give feedback about a concept and to carry out discussions. These comments and annotations are also represented as instances in ChAO.

3.3. Markov Chain Model Selection

For our analysis we will rely on the methodology initially presented in Singer et al. [4].

For an easier understanding of the collected results, we will provide a description of the Markov Chain Model Selection Methodology, which will also help readers to get a better understanding for the potential implications of our results.

3.3.1. Markov Chain definition

Markov Chain models are well-known tools, among others, for modeling navigation on the web. 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 – e.g., X_1, X_2, \dots, X_n . One of the most well-known hypotheses about Markov Chains is the so-called *Markovian assumption* that postulates that the next state of a sequence only depends on the current state and not on a sequence of preceding ones. Such a first-order Markov Chain holds if:

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

We assume *time-homogeneity* for all our Markov Chains and for simplification we will refer to data as a sequence $D = (x_1, x_2, \dots, x_n)$ with states from a finite set S . Furthermore, as we are also interested in higher order Markov Chains, we can state that in a k -th order Markov Chain the next state depends on k previous ones. One advantage of such representation is 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 concepts 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 :

$$\sum_j p_{ij} = 1 \quad (2)$$

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.

3.3.2. 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 3, where n_{ij} corresponds to the total number of transitions between states x_i and x_j :

$$p_{ij} = \frac{n_{ij}}{\sum_j n_{ij}} \quad (3)$$

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.

3.3.3. Model Estimation

As our goal is to determine the most appropriate Markov Chain order we need to establish some methods for choosing the right ones.

As described in more detail in Singer et al. [4], we calculate the Akaike information criterion (AIC) and Bayesian information criterion (BIC) to determine the goodness of fit for our extracted Markov Chain models.

Likelihood Ratio Test. To be able to calculate AIC and BIC, we have to calculate the likelihood ratio tests, which are used for comparing two competing hypotheses, the null-model and the alternative-model. The test calculates the log likelihood ratio, which 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. We follow the notation by Tong [44] who defines the log likelihood ratio as ${}_k\eta_m$:

$${}_k\eta_m = -2(\mathcal{L}(\mathcal{P}(\mathcal{D}|\theta_k)) - \mathcal{L}(\mathcal{P}(\mathcal{D}|\theta_m))) \quad (4)$$

Where $\mathcal{L}(\mathcal{P}(\mathcal{D}|\theta_k))$ represents the MLE for the null-model, while $\mathcal{L}(\mathcal{P}(\mathcal{D}|\theta_m))$ represents the MLE for the alternative model.

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 – [45] and has been first used for Markov Chains by Tong [44]. We define the AIC based on the work by Tong [44]:

$$AIC(k) = {}_k\eta_m - 2(|S|^m - |S|^k)(|S| - 1) \quad (5)$$

Basically AIC subtracts the degrees of freedom from the likelihood ratio test. In our analysis, the degrees of freedom $(2(|S|^m - |S|^k)(|S| - 1))$ represent two times the difference between the states for the null-model and the alternative model. The basic idea is to choose m reasonable high – i.e., the maximum order we want to test against – and test lower order models until the optimal Markov Chain order is found. The best 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 [46]:

$$BIC(k) = {}_k\eta_m - (|S|^m - |S|^k)(|S| - 1)\ln(n) \quad (6)$$

We proceed similar as for AIC and choose m reasonable high. The specific penalty function is the degree of freedoms multiplied with the natural logarithm of the number of observations n [46], where an observation is always represented as a state in the change-logs.

3.3.4. Evaluation

Exceeding our MLE methods for determining the appropriate Markov Chain order, we also use cross validation prediction for this task. The main idea behind this approach is to calculate the parameters on a training set and validate the model on an independent test set. One needs to note that we also apply Laplace smoothing in this case in order to also be able to predict states that are only present in the test set and not in the training set. For reducing variance we perform a stratified 7-fold cross validation. In this case we refer to the term stratified as we try to keep the number of visited states in each fold equal.

The validations is based on the task of predicting the next step in a path of the test set. This 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 one with the highest probability in the transition matrix. For clearly calculating the prediction accuracy we measure the average rank of the actual state of the test path in the sorted probabilities from the transition matrix P . Hence, we look up the rank of the next state x_{n+1} in the sorted list of transition probabilities of the current state x_n (or series of preceding states for higher order models). 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 way⁵:

$$\overline{r}(D_f) = \frac{\sum_i \sum_j n_{ij} r_{ij}}{\sum_i \sum_j n_{ij}}, \quad (7)$$

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 average rank to such ties. This method also includes a natural Occam's razor (penalty) for higher order models. Finally, we average over all folds and suggest the model with the lowest average rank.

This method also allows us to get detailed insights into the predictive power of our models, datasets and investigations.

3.4. Session Extraction

As browsing and identifying concepts to work on in an ontology, especially with ontologies the size of ICD-11, are still time-consuming tasks, we decided to add *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

⁵alternatively, one could also measures like perplexity

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 4.3).

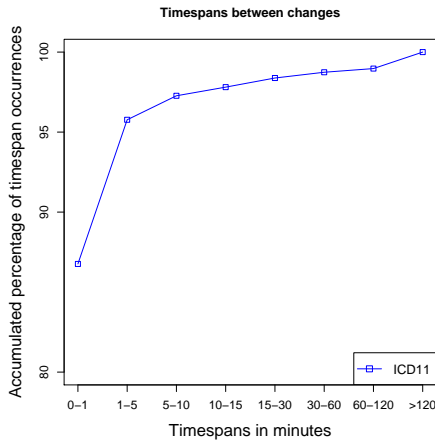


Figure 3: This plot depicts the percentage of all changes that have been performed within a specific timespan for ICD-11. The x -axis lists the timespans in minutes and the y -axis lists the accumulated percentage of all timespans between two consecutively conducted changes for every user. To avoid the 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 *BREAKs* have been introduced if the two changes in question are apart longer than 5 minutes.

Figure 3 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.

3.5. Limitations

It is important to understand that it was not possible to recreate the structure of the ontology for every single change across our observation period. This is partly due to a lack of data in the change-logs but also due to computational feasibility, as we would have to create and store n revisions of each ontology where n equals the number of changes performed on each ontology. Therefore, we decided to conduct our analysis, using all ontologies *as is* at the latest point in time, which is also what would be used in real-world usage scenarios.

However, this approach introduces a potential bias to our *Edit-Strategy Paths* analyses. Hence, to measure the bias we collected every change that was performed on a concept, which was moved to a different location in the ontology at a later point in time. Applying this selection criterion on our change dataset, we collected a total of 116, 204 of 439, 229 changes for ICD-11. This represents about 1/4 of all changes in the change-log.

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 is necessary, as specifically higher order models need an exponential growing number of parameters which might not be reflected by the statistical significant benefit against lower order models. This is also reflected by the initial choice about the set of states to consider. Hence, it is also necessary to have the availability of large change-logs 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 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, 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. Therefore, without enough observations (changes), the identification of higher order Markov Chain models is either very hard and can only be approximated or not possible at all. As can be seen in Table 1 ICD-11 satisfies this requirement with a total amount of close to 440, 000 changes.

4. Results

In the following, we present the model selection results of the conducted sequential pattern analyses for ICD-11. In section 4.1 we investigate if and to what extent memory and structure of sequential patterns of performed change types can be detected.

To see where and how users contribute to the ontology and if they exhibit sequential patterns when doing so, we analyzed edit strategy patterns (i.e., *bottom-up* or *top-down* editing behavior; cf. section 4.2)?

In section 4.3 we investigate if users have to frequently switch between the different sections of the user-interface while contributing to ICD-11 and how often (and in which order) different sections of the user-interface are used to add information for a concept.

4.1. Change-Type Paths

The analysis of change types provides information about the type of a change which a user will most likely conduct next. To reduce the state space we aggregated similar types of changes into more abstract change-classes.

Path Extraction: For analyzing and extracting change-type sequences we iterated over all the changes of each user in our datasets, mapped the different types of changes to our aggregated change-classes, and 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*. For example, if the same user consecutively replaced properties (*EDIT_REPLACE*) on the same concept multiple times we would merge these multiple *EDIT_REPLACE* actions into one single *self-loop* from the state *EDIT_REPLACE* to the state *EDIT_REPLACE*. This approach has been taken to avoid the detection of higher order Markov Chains solely due to repetitive tasks (i.e., consecutively replacing values of properties of the same concept).

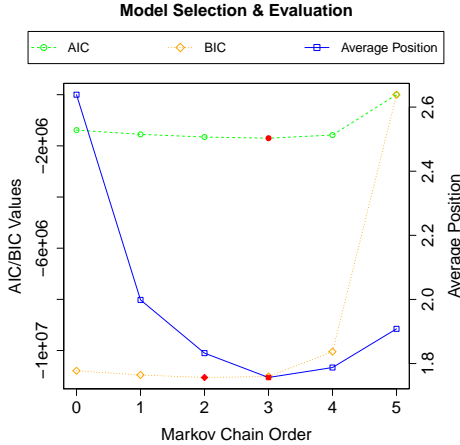


Figure 4: **Change Type Paths Model Selection and Evaluation:** This plot depicts the results of the AIC and BIC model selection criteria as well as the stratified cross-fold evaluation for the *Change Type Paths* analysis. The x-axis represents the different Markov Chain orders. The left y-axis lists 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.

State Description: For this analysis 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. The states *CREATE*, *MOVE* and *REMOVE* include all changes that have a corresponding effect on classes or concepts of the ontology. Every state that has the prefix *EDIT_* aggregates all changes performed on properties of the ontology. For example, when a user performs a change of type *EDIT_ADD* a value is added to a property of a concept, while *EDIT_REPLACE* would be the description of a change that replaces an already existing value of a property with a new (non empty) value. The states *EDIT_IMPORT* and *BOT* contain changes of concepts to properties with references (or imports) to other ontologies as well as changes marked as automatically performed. All changes that are collected under *OTHER* are changes that are not changing the content or the structure of the ontology.

Model Selection: We calculated AIC and BIC for the extracted Markov Chain models of varying order (Figure 4) to identify the appropriate order that reflects to what extent contributors exhibit memory patterns when changing concepts.

Both 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.

Evaluation: To determine which order of a Markov Chain contains the highest predictive power, a stratified cross-fold val-

idation (see Figure 4; section 3.3.4 for a detailed explanation) was conducted.

As depicted in Figure 4, 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.

4.2. Edit-Strategy Paths

The analysis of *Edit Strategy Paths* focuses on the investigation of relative movement along the ontological structure that is to be created. Using the gathered data we can identify if users who are contributing to the ontology likelier follow a *bottom-up* or *top-down* manner. 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 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/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 datasets do not exhibit such behaviors we have manually investigated the structured log of changes of ICD-11 to verify that the mentioned kind of contribution behavior is not present.

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.

Path Extraction: The sequences used for this analysis were gathered by calculating the shortest paths between all the concepts of the ontology and the root node, for ICD-11 being *ICD-Category* which is an equivalent of *owl:Thing*, following *isA* relationships. Analogously to the previous analysis, we merged multiple *self-loops*, represented by consecutive changes performed by the same user on the same concept, into one single transition (i.e., multiple transitions between *SAME* into one transition from *SAME* to *SAME*) to avoid the generation of higher-order *self-loop*-based Markov Chain models. Users who contributed less than 2 changes have been removed for this analysis, as no level transition could be observed.

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, UP, DOWN*.

State Descriptions: The states used for this analysis indicate if a user, when contributing to the ontology, moved either closer (state *UP*), further away (state *DOWN*) or kept the same distance (state *SAME*) to the root concept of the ontology.

Model Selection: We calculated AIC and BIC for the extracted Markov Chain models (see Figure 5) 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.

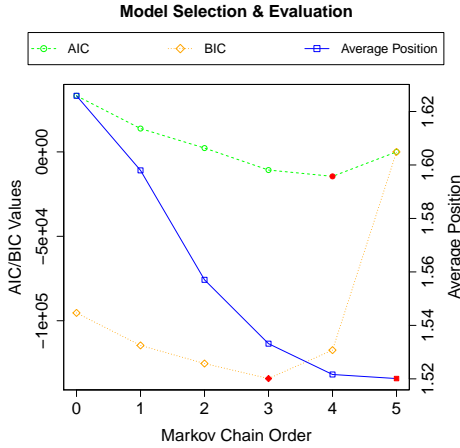


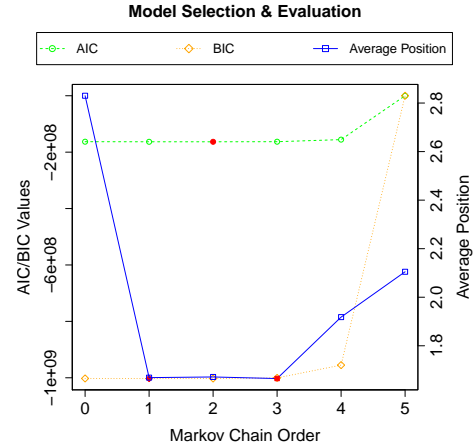
Figure 5: **Edit Strategy Paths Model Selection and Evaluation:** This plot depicts the results of the AIC and BIC model selection criteria as well as the stratified cross-fold evaluation for the *Edit-Strategy Paths* analysis. The x-axis represents the different Markov Chain orders. The left y-axis 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 fifth-order Markov Chain model.

Evaluation: To determine the best-fitting Markov Chain model orders to predict the next relative depth-level a stratified cross-fold validation (see Figure 5) was conducted. The results of our prediction experiment suggest a fifth-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 significantly different, we agree with BIC and the significance test on the usage of a third-order Markov model for predictive tasks, due to the high increase in complexity of the higher-order models.

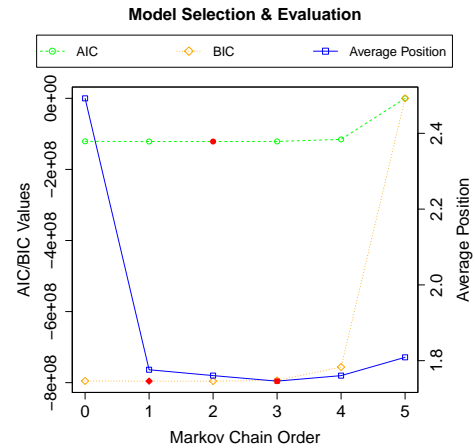
4.3. User-Interface Sections Paths

The goal of this analysis is to investigate if we can map changes that occur in the ontology to actual areas of the user-interface of the used collaborative ontology-engineering tool. 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, the ontology-development tool used to create ICD-11, are already actively used as ICD-11 is still under active development.



(a) User-based approach



(b) Concept-based approach

Figure 6: **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 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 were able to detect a second- and first-order Markov Chain respectively for both approaches, while the prediction task produced the best average position with a Markov Chain of third order in both approaches.

Path Extraction 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. The states

Table 2: **This Table depicts a summary of all gathered results for ICD-11 and the performed analyses** of section 4. The numbers in this table represent the calculated and suggested Markov chain orders from our model selection (AIC and BIC) 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	Prediction Task	Best Balance
Change-Type Paths (cf. section 4.1)	3	2	3	3
Edit-Strategy Paths (cf. section 4.2)	4	3	5	1
User-Interface Sections Paths (User) (cf. section 4.3)	2	1	3	1
User-Interface Sections Paths (Concept) (cf. section 4.3)	2	1	3	1

used for the four different approaches of the *User-Interface Sections Paths* analysis are the different sections available in the user-interfaces of iCAT. Whenever a change did not affect a property (e.g., because the change-action dealt with moving or creating a concept) the *no property* state was used. To get a better feeling for the interface of the ontology-editor used to develop ICD-11 please refer to Figure 1. Analogously to the previous analyses, consecutive changes of the same user on the same concept on the same property have been merged 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*.

State Descriptions: The states for this analysis are represented by the different user-interface sections of iCAT, the ontology-engineering tool used to develop ICD-11. An excerpt of all different user-interface sections of iCAT can be seen in Figure 1.

Model Selection: We calculated AIC and BIC for the extracted Markov Chain models (see Figures 6(a) and 6(b)) to find out 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.

Evaluation: To determine the predictive power of the investigated Markov Chain models of different orders for predicting the section most probably used to edit a property next, a stratified cross-fold validation (see Figure 6) was conducted. For both approaches, the user-based approaches and the concept-based approach, a third-order Markov Chain performed best. However, due to the fact that the determined third-order Markov Chains performed virtually equally as good 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.

5. Discussion

In section 5 we have shown that the presented and adapted Markov chain model selection framework provides new insights into the ongoing development processes of collaborative ontology-engineering projects. As shown in Table 2, Markov chains of orders three to five yield the best results in our prediction task. 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 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. For the *Edit-Strategy Paths* as well as both approaches of the *User-Interface Sections Paths* analyses a first-order Markov chain constitutes the best mix between model complexity and performance.

To further expand 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 Sections Paths* analysis including information about potential use-cases in productive environments.

5.1. Results of the User-Interface Sections Paths Analysis

Figure 7(a) depicts the user-interface section sequence for properties changed on concepts in ICD-11 and shows that sections of the user-interface frequently receive consecutive changes with minimal transition probabilities to different sections of the user-interface. All sections, which were rarely used, have been removed from Figure 7 as they do not hold important information but their removal drastically increased the readability of the plots. When directly comparing the *self-loops* for the user-based approach (see Figure 7(a)) and the concept-based approach (see Figure 7(b)), we can deduce, based on the high transition probabilities between the same sections and the low transition probabilities between different sections) that users have a higher tendency towards consecutively inserting the same set of properties (displayed in the same section of the user-interface), into multiple concepts rather than completing all properties of a concept before moving on to change the next concept.

As depicted in the histograms of Figures 7(a) and 7(b), the majority of changes were concentrated on a few selected sections, which are *Title & Definition, Classification Properties* and *Terms*.

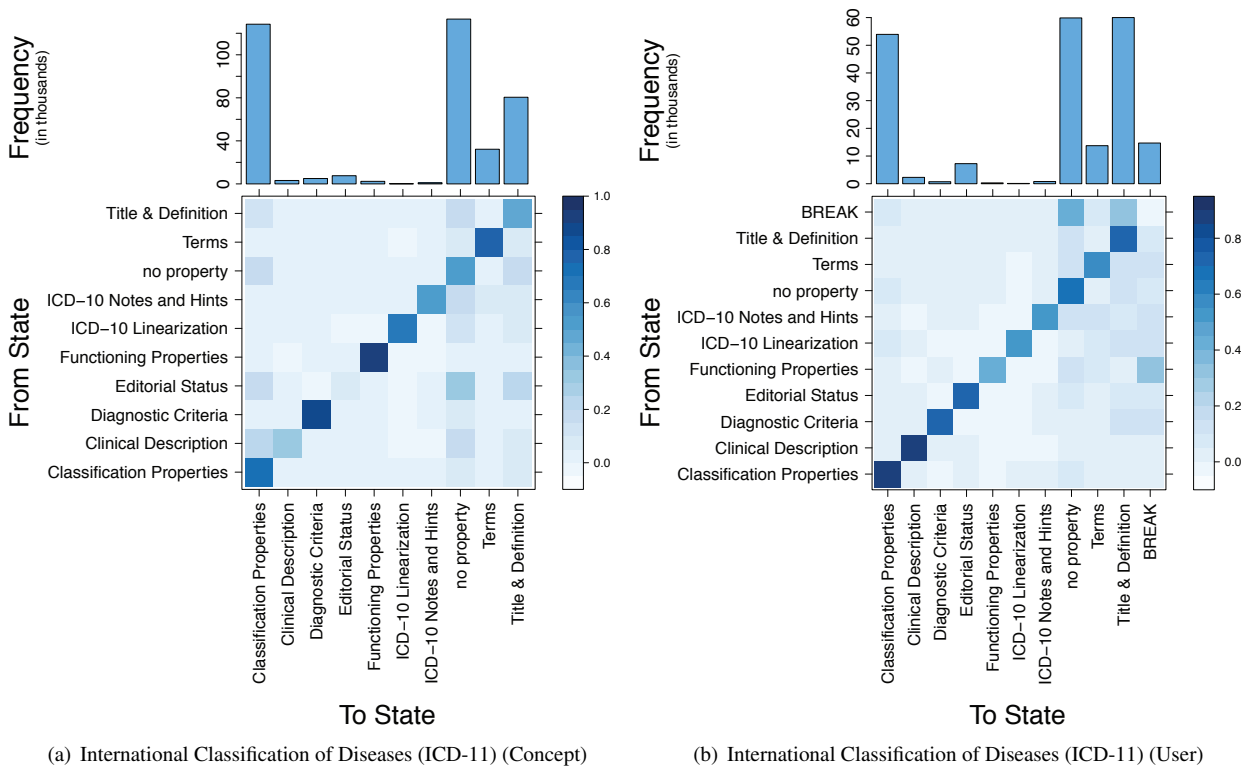


Figure 7: **Results for the User-Interface Sections Paths analysis:** The states for these analyses are represented by the different sections of the user-interfaces of the ontology-engineering tool iCAT. The transition probabilities for the first-order Markov Chains are depicted in the transition maps (**bottom row**) in Figures 7(a) and 7(b). Columns and rows represent the states (for $k = 0$), 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. Absolute probability values are dependent on the number of investigated states, hence relative differences are of greater interest. Across both approaches a trend towards *self-loops* can be observed, which is marginally higher and less diverse for the user-based approach in ICD-11, rather than the concept-based approach. The histograms in Figures 7(a) and 7(b) depict the absolute number of occurrences for each section (at $k = 0$) for ICD-11 in alphabetical order. Note, that the y-axis for both histograms are scaled differently. Sections with very low numbers of observations have been removed from the plots for reasons of readability. All sections, which were rarely used, have been removed from this figure as they do not hold interesting information but their removal drastically increased the readability of the plots. Note, that the differences in frequencies of the different user-interface sections

Contributors of ICD-11 also exhibit a very high tendency to either change *no property* or a property of the *Title & Definition* section when resuming work after a *BREAK*.

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 the section *Title & Definition* exhibits a very high *self-loop* probability, 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, which is creating the ontology in question. 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 This could mean that the sections

6. Summary & Conclusions

The novel application of Markov Chains on change-logs of collaborative ontology-engineering projects represents a first step towards a broader methodology to collect new insights about the processes and interactions of users with the collaboratively developed ontology and the used ontology-engineering tool. The main contributions of this paper are: (i) The novel application of (higher order) Markov Chains on collaborative ontology-engineering projects to extract and analyze sequential patterns. Moreover, we have shown that (ii) analyzing sequential patterns can be used to gather new insights into various aspects for collaborative ontology-engineering projects and (iii) determined the model that provided the best balance between model complexity and predictive power in our model selection task.

Additionally, we have shown that historic change-log information of collaborative ontology-engineering projects can be used to predict the state that is most likely to occur in the system next, using Markov chains. 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 exemplary investigation of the structure of a first-order Markov chain and its implications and use-cases for productive environments.

As change-tracking and even click-tracking data will become available more broadly, we believe that the mapping analysis conducted in this paper and the possible benefits of putting the results into practical use represent an import step towards even better (and simpler) ontology editors, which can 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 right 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.

- [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. URL: <http://www.sciencemag.org/cgi/content/abstract/327/5968/1018>. doi:10.1126/science.1177170. arXiv:<http://www.sciencemag.org/cgi/reprint/327/5968/1018.pdf>.
- [4] P. Singer, D. Helic, B. Taraghi, M. Strohmaier, Memory and structure in human navigation patterns, arXiv preprint arXiv:1402.0790 (2014).
- [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] 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.
- [9] C. Ghidini, B. Kump, S. Lindstaedt, N. Mahub, 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.
- [10] 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.
- [11] T. Schandl, A. Blumauer, Poolparty: SKOS thesaurus management utilizing linked data, *The Semantic Web: Research and Applications* 6089 (2010) 421–425.
- [12] M. Krötzsch, D. Vrandečić, M. Völkel, Semantic MediaWiki, in: Proceedings of the 5th International Semantic Web Conference 2006 (ISWC 2006), Springer, 2006, pp. 935–942.
- [13] 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. URL: <http://dblp.uni-trier.de/db/conf/aaais/aaais2008-7.html#NoyT08>.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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).
- [18] 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>.
- [19] 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. URL: <http://dblp.uni-trier.de/db/conf/kcap/kcap2011.html#FalconerTN11>.
- [20] C. Pesquita, F. M. Couto, Predicting the extension of biomedical ontologies, *PLoS Comput Biol* 8 (2012) e1002630. URL: <http://dx.doi.org/10.1371/journal.pcbi.1002630>. doi:10.1371/journal.pcbi.1002630.
- [21] 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.
- [22] 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.
- [23] 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.

- [24] R. Lempel, S. Moran, The stochastic approach for link-structure analysis (salsa) and the tlc effect, *Comput. Netw.* 33 (2000) 387–401. URL: [http://dx.doi.org/10.1016/S1389-1286\(00\)00034-7](http://dx.doi.org/10.1016/S1389-1286(00)00034-7). doi:10.1016/S1389-1286(00)00034-7.
- [25] 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.
- [26] 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>.
- [27] 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>.
- [28] 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.
- [29] 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>.
- [30] 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.
- [31] 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 networking*, North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2000, pp. 377–386. URL: <http://dl.acm.org/citation.cfm?id=347319.346322>.
- [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] 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.
- [34] 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.
- [35] A. Cabrera, E. F. Cabrera, Knowledge-Sharing Dilemmas, *Organization Studies* 23 (2002) 687–710.
- [36] 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.
- [37] 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.
- [38] N. Shachaf, Beyond vandalism: Wikipedia trolls., *Journal of Information Science*; Jun2010, Vol. 36 Issue 3, p357-370, 14p, 2 Charts (2010).
- [39] 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.
- [40] 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.
- [41] F. Wu, D. S. Weld, Open information extraction using wikipedia, in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10*, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 118–127.
- [42] S. Xu, S. Yang, F. C.-M. Lau, Keyword extraction and headline generation using novel word features., in: M. Fox, D. Poole (Eds.), *AAAI, AAAI Press*, 2010.
- [43] 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.
- [44] 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>.
- [45] P. Gates, H. Tong, On markov chain modeling to some weather data, *Journal of Applied Meteorology and Climatology* 15 (1976) 1145–1151.
- [46] R. W. Katz, On some criteria for estimating the order of a markov chain, *Technometrics* 23 (1981) 243–249.