# Empirical Analysis of Impacts of Instance-driven Changes in Ontologies

Yalemisew M. Abgaz[1], Claus Pahl[2]

Centre for Next Generation Localization (CNGL),
School of Computing, Dublin City University, Dublin 9, Ireland
{yabgaz[1]|cpahl[2]}@computing.dcu.ie

**Abstract.** Changes in the characterization of instances in digital contents are one of the rationales to change or evolve ontologies which support the domain. These changes can impacts on one or more of interrelated ontologies. Before implementing changes, their impact on the target ontology, other dependent ontologies or dependent systems should be analysed. We investigate three concerns for the determination of impacts of changes in ontologies: representation of changes to ensure minimum impact, impact determination and integrity determination. Key elements of our solution are the operationalization of change operations to minimize impacts, a parameterization approach for the determination of impacts, a categorization scheme for identified impacts, and prioritization technique for change operations based on the severity of impacts.
**Keywords:** Ontology evolution, impact determination, instance-driven change.

## 1 Introduction

Ontology evolution is a continuous process. Whenever there is a change in the domain, its conceptualization or specification, the ontology needs to be changed [11][7][14]. Ontologies, built to give support for specific content within a domain, change as content and embedded ontology instances change and need to be updated synchronously with changes in the domain [1][15].

When new concepts are added or existing ones are deleted or modified in the content, the respective ontology needs to be updated. Implementing the changes requires understanding the changes correctly and representing the changes accurately using ontology change operations. However, it only solves few of the problems associated. These changes can trigger further cascaded changes and affect one or more interrelated ontologies that are dependent on the changing ontology. The effects of the change may propagate back to the domain instances in the content leaving the process in a circle. An ontology engineer who detects a change of an instance in a content document and trying to maintain the ontology accordingly may end up with so many unseen impacts. When the ontology engineer has large interrelated ontologies, the process of determining impacts of change operators will become time consuming and error prone. Thus, in larger and shared ontologies, the determination of change impact is crucial.

In this research, some of the key features we investigate are:

– a case-based real-world requirement analysis.
– an analysis to determine impact of instance-driven changes in ontologies:
  - Operationalization: how to operationalize changes to ensure minimum impact?
  - Parameterization and Categorization: how to determine different impact categories and parameters to determine impact?
  - Integrity: how to determine the integrity (consistency within and among dependent ontologies and validity within instances and ontologies) of the ontology due to the changes?
  - Prioritization: how to choose the best options with minimum impacts in different situations?

While a significant number of approaches [12][10][6][3] focus on addressing consistency and validity of the ontology at the time of change, we focus on analysis and determination of impacts of change operations to minimize their impact in not yet evolved ontologies. We operationalize changes to ensure minimum impact, define parameters to identify and determine impacts, categorize impacts to deal with them at different level of expertise and prioritize impacts to enable us to choose the options with minimum impacts in different situations.

This paper is organized as follows: Section 2 describes the empirical study and Section 3 focuses on the selection of schemes for impact analysis and identification of parameters. Section 4 presents our proposed framework for the impact determination process. The actual categorization of impacts using different criteria is discussed in Section 5. Evaluation of the results is given in Section and related work in Section 6. We give conclusions in Section 7.

## 2 Empirical Study

We conducted an empirical study on the help files of a content management software system with the aim of supporting our proposed theoretical solution with an empirical experiment. The case study is selected because it has a wide coverage domains from the application domain to software systems, which are interdependent on another. Moreover concepts and instances are distributed throughout the content of the help files and create a strong link between the instances in the content and the concepts in the ontologies. This makes it of great interest to investigate instance-driven change impacts because the changes made in the contents of the help files will have a direct impact on the ontology and vice versa.

There are four primary ontologies identified for supporting software systems help files [2]. A high-level description of these ontologies and their dependencies are depicted in Fig. 1. The DocBook ontology gives structure to and defines how elements in the help ontology are organized. The help ontology guides the software ontology in a way that explains how the software ontology makes use of the topics, procedures etc. The software ontology allows us implement the ontology domain which is specific to the components in each application.

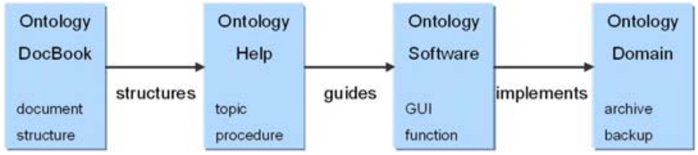


**Fig. 1.** Ontology Hierarchy

We identified scenarios that represent possible changes that may occur when there is new software release, technology change and/or software structures adjustment. The scenarios further represent frequent changes that occur on the instances of the help files. The scenarios are extracted from the real world changes in the software industry - specifically changes between an old and a new version of the software help content. Our focus is on content changes that trigger a change in an ontology and the scenarios are selected based on impact of changes in instances. Initially, 15 scenarios were identified that cover all the four ontologies. Based on their frequency, their cascaded impacts, the operations involved and the number of ontologies they cover, we selected scenarios that are most representative of the evolution process. Two scenarios are discussed below.

**Scenario 1:** The new version of the software resulted in a change on a component which contains other two sub components. The component and one of its subcomponents are removed but another subcomponent is upgraded to a full component. Here we do need to link all the previous instances associated with the removed concepts to the upgraded concept. The desired output is an updated ontology which reflects the change requested. Change operations are:

– Move up ("sub component 2")
  - Add instance of ("instance of component", "sub component 2")...
  - Add instance of ("instance of sub component 1", "sub component 2")...
  - Delete concept ("component")
– Delete concept ("sub component 1")

The primary ontology affected is the Application Ontology.

**Scenario 2:** The software engineers introduced a new software component. The new component has new associated help files. The desired output is a software application ontology that has a description of the new component and its properties. Change operations are:

– Add Concept ("new component")
– Add sub concept ("new component", software Application)
– Add instance ("help file")
– Add instance of ("help file", "new component")

The primary ontology affected is the Application Ontology.

The scenarios are used to evolve the ontologies and to analyze and determine impacts of the changes.

## 3 Schemes for Impact Analysis

In situations where interrelated ontologies are used, the change of one element in one ontology may have an impact on other elements within the same ontology or elements among the interrelated ontologies. The dependency between ontologies, especially when they are used in a specialized domain is often high. Thus, the impact determination process focuses on identifying impacts of change operations on one or more interrelated ontologies.

### 3.1 Types of Ontology Change Impact

The term impact refers to a consequential change of the state of an ontology or elements in the ontology due to the application of a change operation on one or more of the elements in the ontology [10] [11][9]. The impact can be structural or semantic. Structural impact is an impact that occurs on the structural relationship between the elements of the ontology. Semantic impact is an impact that occurs on the interpretation of the ontology and its elements.

Structural impacts are possible consequences on the structure of the ontology due to a structural change.

- Broken Structure:
  - Orphan concept: the change operator may introduce an orphan concept in the ontology
  - Orphan Properties: the change operator may introduce an orphan property in the ontology (properties with out domain, properties with out parents)
  - Orphan instance: the change operator may introduce an orphan instance in the ontology
- Cyclic structure: the change operator may introduce a cyclic structure in the ontology

Semantic impacts are possible inconsistencies and invalidities that arise for the interpretation of the ontology due to structural changes [10].

- Generalization/ specialization: elements (concepts, instances, domains/ranges of properties) move up or down in the hierarchy
- More/less description: a data type property or instance level object property is added to or deleted from a concept
- More/less restrictive: a change to the restriction further restricts or extends its semantics
- More/less extended: a change to its axioms further extend or restricts its semantics

To determine the impact of change operations first, we identified the following parameters that determine the nature of impact of a change operation on the elements of an ontology, see Table 1.

An example shall explain the approach. All instances of Assign Role in one version of a help file have been changed to either Assign AdministrativeRole or

**Table 1.** Sample Parameters for Impact Determination

| *General Parameters* | *Concept Parameters* | *Property Parameters* | *Axiom Parameters* | *Restriction Parameters* |
|---|---|---|---|---|
| Operation | Target concept | Target property | Target axiom | Target restriction |
| Ontology element type | Has sub/super class | Has sub/super property | Has domain | Has domain |
| Ontology target element | Has domain/range | is data/object property | Has range | Has range |

Assign UserRole in the newer software version. This change can be represented by a composite operation SplitConcept(Role, AdministrativeRole, UserRole). To determine the impact of the change on the ontology, we need to know what the target entity is (the concept Role), whether it has subclasses and/or super class, whether it has a data property or object property, and if it is a domain or a range of a property. These parameters about the target concept provide us with information about what potential impacts are associated with a change.

### 3.2 Change Operations

The following is a list of possible changes that may occur on the structure and the semantics of ontology. Changes can be atomic, composite or domain specific. Higher levels of changes (composite and domain-specific) are created by combining atomic changes in a certain order [5]. Renaming can be done by a series of addition and deletion operations; thus, is not discussed here. Addition and deletion are the basic change operations. They are applied on concepts, properties, restrictions, axioms and instances as target elements. Thus, we have change operations like Add concept, Delete concept, Add property, and Delete axiom.

## 4 Framework for change impact and Integrity Analysis

The empirical study further clarifies that impact determination is a step-by-step process, see Fig. 2. These steps fit into the semantics of change phases of the general ontology evolution process [11].

### 4.1 Change Request Capturing and Representation

In this first step, the objective is to represent detected changes using suitable change operators that ensure the efficient implementation of the required change. The accurate and successful execution of the requested change depends on how the change is represented [13] [8] and relies on two factors. First, the selection of the appropriate operator and, second, the order of execution of the operations focusing on efficient ordering of atomic change operations into composite and higher-level granularity to minimize impact [9] [5].

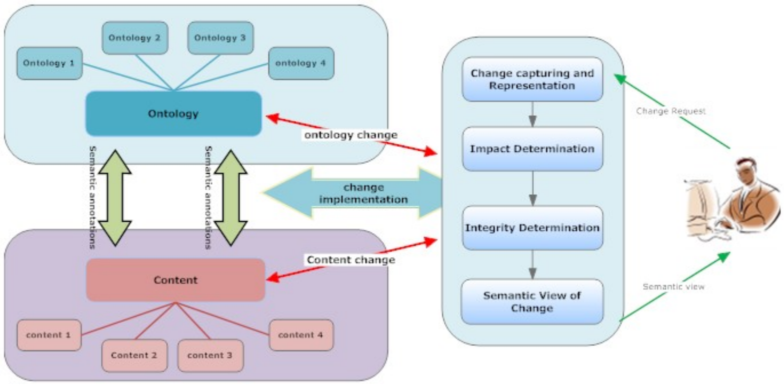


**Fig. 2.** Instance-driven Change and Impact Determination Process

### 4.2 Impact Determination

This step mainly focuses on determining the impacts of the captured change operations on the elements of the ontology. Impact determination process focuses on analyzing the nature of the operations and the target ontology elements using different parameters. Based on these parameters, this phase enables categorization of change operations into different category of impacts. This phase further identifies part of the ontology that is affected by the change. This process is crucial to deal with part of the ontology that is impacted by the change operation rather than dealing the whole ontology.

### 4.3 Integrity Determination

**Consistency.** Once the scope of the impact of the change operations is identified, the next step is to analyze how the consistency of the ontology is affected by these changes. Consistency is analyzed based on consistency rules that are defined for the ontology. Since these rules are defined prior to the implementation of the change operations, it is possible to determine which consistency rules will be violated if a change is made on the ontology [4]. Defining consistency rules is important for consistency analysis. We focus here only on the following widely used rules for an ontology [13]:

1. *Identity invariant:* no two elements should have the same id (URI)
2. *Rootedness invariant:* there should be a single root in the ontology
3. *Concept hierarchy invariant:* no element should have a cyclic graph
4. *Closure invariant:* every concept should have at least one parent concept except the root concept
5. *Cardinality invariant:* the cardinality of a constraint should be a non-negative integer greater than or equal to the minimum cardinality and less than or equal to the maximum cardinality

6. *User-defined constraints:* these constraints are user-defined and needs to be stated in the way they can be implemented like the other invariants. Defining and using user-defined constraints is beyond the focus of this research.

**Validity.** Instances in content centric systems are linked to the ontology using semantic annotation [15]. Thus determining the impact of change operations with regard to the instances is crucial. The determination of the validity of instances and instance properties is also based on the validity rules. Theses rules determine how instances/ instance properties should exist in the ontology structurally and how they should be interpreted:

1. *Invalid instance:* given a consistent ontology, if there is an instance that does not correspond to any of the concepts, then that instance is invalid.
2. *Invalid interpretation:* given a consistent ontology, if there is an instance whose interpretation contradicts any interpretation denoted by the consistent ontology, that instance has an invalid interpretation.

## 5 Categorization of Impact

The categorization of impact is important to systematically handle impacts and prioritise crucial impacts to save much time.

### 5.1 Impact based on Severity

Severity is the degree of impact of a change operation on an ontology. The impact can be on the structure or semantic, consistency or validity of the existing ontology. Using the selected scenarios to determine the severity of impact of change operations, we analyze how different change operations impact the ontologies. Analyzing each parameter used in the scenarios against the impact observed gives us a better understanding of which operations under which condition have a severer impact than others. The need for defined categories and identification of properties is needed as input for automatic categorization at a later stage.

1. *Less or no Impact:* changes with no effect on the consistency or the validity of the ontology, e.g. addition of a concept at the bottom of a hierarchy.
2. *Medium impact:* changes with medium impact that can be solved using predefined operations, e.g. addition of concepts in the middle of a hierarchy.
3. *High impact:* changes that create structural inconsistency and require little or no human involvement, usually restricted to a single ontology, e.g. deletion of concepts with subclasses and annotation links.
4. *Crucial impact:* changes that significantly affect the consistency of the ontology, affecting dependent ontologies and instances and their interpretations. They require expert involvement, e.g. deletion of concepts or addition of axioms which create invalid or inconsistent interpretation of elements.

### 5.2 Impacts based on Type of Operation and Target Elements

Some change operations have little impact on the ontology, others have greater impact. Thus, identifying the operations that have less impact and greater impact is important to approach the problem systematically. Based on the parameters identified in Table 1, Table 2 summarizes impacts based on their severity. The table indicates the severity of the type of atomic change operations and the type of ontology elements in their likelihood of occurrence. The severity of impacts of composite change operations can be determined using the atomic change operations involved.

**Table 2.** severity of change operations and type of elements

| Type of operation | Type of element |
|---|---|
| Deletion | Concept, Property [object property, then data property] |
| | Axioms, Restrictions |
| | Instances, then instance properties |
| Addition | Axioms, Restrictions |
| | Properties, concepts |
| | Instances, instance properties |

### 5.3 Impacts based on Constraints Violated

The constraints that are violated by the change operation have different levels of impact severity on the ontology. This idea is backed by the empirical study and is described below. Categories

1. the strength of the consistency and validity rules being violated, e.g. invariant constraints, soft constraints and user defined constraints.
2. the level of human involvement required, e.g. can the system carry out the operations autonomously or is a human intervention to choose between actions is needed)

In terms of the constraint rules violated the following list shows the severity in a descending order:

- Invalid interpretation, Closure invariant
- Concept hierarchy invariant, Invalid instances, Cardinality invariant
- Identity invariant, Rootedness invariant, Soft constraints
- User defined invariants, maybe severe based on the requirements of the user

## 6 Discussion and Related Work

Our approach has been evaluated based on its practical applicability and its operational applicability in the real world. From the experimental study, we found out that the proposed solution is efficient in reducing the number of change operations, and consequently the number of cascaded impacts, significantly. The solution is further evaluated and we found that it enables us classify impacts into the appropriate categories. Furthermore, it ensures consistency and validity of the resulting ontology.

To put our findings into context, we give a brief summary of current practice in the area of ontology evolution, specifically in handling instance-driven change. An interesting research [10] looks at determining the validity of instances in evolving ontologies. The authors evaluated the validity of data instances against changing ontologies and came up with a formal model. They presented formal notion of structural and semantic validity of data instances. Compared to our work, their work focuses on determination of validity of data instances after a change takes place, but do not address the problem of determining impact. In [14], the effects of domain changes on the performance and validity of the knowledge-based systems are discussed. They analyzed the problems using non-evolved ontologies and present a solution for enabling consistent description of knowledge sources. However, their work emphasizes problems related with metadata evolution and annotation and does not focus on impact determination. [4] discusses consistent evolution of OWL ontologies with the aim of guaranteeing consistency whenever the ontology evolves. Their focus is on structural, logical and user-defined consistency, but does not formally focus on analysis, parametrization and categorization of change impact.

## 7 Conclusion and Future Work

In this paper, we empirically analyzed and determined the impact of instance-driven change in ontologies. Based on results of the empirical study, we operationalize, categorize, parameterize and prioritize changes and analyzed their impacts. Based on the severity of the impact, the changes are further analyzed. The case study has highlighted details of problems associated with instance-driven ontology changes and the difficulty of the problem solutions.

The major contribution of our work is the determination of the impact of change operations that are carried out on content-oriented ontologies. The research further contributes to identifying and categorizing change operations based on their impacts, identifying parameters that play significant role in determination of impacts and categorization of the change operations based on the severity of the impacts. We identified parameters for determining severity of the impacts like the cascaded effect, the time required (number of operations) and the human involvement to resolve complex choices.

Our future work will focus on content-oriented ontology change impact determination in a web-based multilingual environments. Specifically, we will focus on the sliced Web content annotated using the domain ontologies investigated here. Another complexity to be investigated are multilingual content and ontologies infrastructures.

## 8 Acknowledgment

This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/CE/I1142 as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University (DCU)

## References


1. V. Benjamins, J. Contreras, O. Corcho, and A. Gomez-perez. 'six challenges for the semantic web'. *In Cristani, M(ED): KR2002 Workshop on the Semantic Web, Toulouse, France.*, 2002.
2. S. Boyce and C. Pahl. The development of subject domain ontologies for educational technology systems. Journal of Educational Technology and Society (ETS) IEEE **10**(3) (2007) 275–288
3. V. Gruhn, M. Wever, and C. Pahl. Data Model Evolution as Basis of Business Process Management. In 14th International Conference on Object-Oriented and Entity Relationship Modelling O-O ER'95, LNCS 1021. 1995.
4. P. Haase and L. Stojanovic. Consistent evolution of owl ontologies. *The Semantic Web:Research and Application*, 3532:182–197, 2005.
5. M. Javed, Y. Abgaz, and C. Pahl. A pattern-based framework of change operators for ontology evolution. *The 4th International Workshop on Ontology content OnToContent 2009. Springer-Verlag. LNCS Series.*, 2009.
6. Y. Liang, H. Alani, and N. Shadbolt. Ontology change management in protégé. *In Proceedings AKT DTA Colloquium, Milton Keynes, UK.*, 2005.
7. A. Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz. Ontologies for enterprise knowledge management. *Intelligent Systems, IEEE.*, 18(2):22–33, 2003.
8. D. E. Oliver, Y. Shahar, E. H. Shortliffe, and M. A. Musen. Representation of change in controlled medical terminologies. *Artificial Intelligence in Medicine.*, 15(1):53–76, 1999.
9. P. Plessers, O. De Troyer, and S. Casteleyn. Understanding ontology evolution: A change detection approach. *Web Semantics: Science, Services and Agents on the World Wide Web.*, 5(1):39–49, 2007.
10. L. Qin and V. Atluri. Evaluating the validity of data instances against ontology evolution over the semantic web. *Information and Software Technology.*, 51(1):83–97, 2009.
11. L. Stojanovic. *Methods and tools for ontology evolution.* PhD thesis, University of Karlsruhe, 2004.
12. L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. User-driven ontology evolution management. *Lecture Notes in Computer Science.*, 6(4):285–300, 2002.
13. L. Stojanovic, A. Maedche, N. Stojanovic, and R. Studer. Ontology evolution as reconfiguration-design problem solving. *Proceedings of the 2nd international conference on Knowledge capture*, 2003.
14. N. Stojanovic, L. Stojanovic, and S. Handschuh. Evolution in the ontology-based knowledge management systems. *Proceedings of the European Conference on Information Systems - ECIS 2002, Gdask, Poland.*, 2002.
15. V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management:requirements and survey of the state of the art. *Web Semantics: Science, Services and Agents on World Wide Web.*, 4(1):14–28, 2006.