



UNIVERSITETET I AGDER

*Building nursing statements based on ICNP*

by

**Bjørn Tore Bjugan**

**Supervisor**

Jan Pettersen Nytnun

Master Thesis in  
**Information and Communication Technology**

FINAL VERSION

The University of Agder

Grimstad, June 3, 2013

## **Abstract**

Correct and professional use of the nursing terminology is important to assure quality information in the nursing practice. The International Classification for Nursing Practice(ICNP) is a unified language system which includes thousands of terms which can be used by nurses when writing electronic health records(EHR). Nurses usually have a specific domain their work relates to and therefore only a limited amount of terms in the ICNP is used. Creating domain specific catalogues with subsets of the ICNP terms and also including common nursing statements in these domains, can help make the teaching and use, more efficient. To make the process of making catalogues easier, a prototype application should be developed with the help of the OWL expressed ontology of ICNP. By extending the ICNP ontology and using the power of Semantic Web for storing catalog data, it will be ready for future research and use.

# Preface

This is a Master Thesis project by Bjørn Tore Bjugan, a student of ICT at University of Agder in Grimstad, Norway. The thesis title is "Building nursing statements based on ICNP".

I would like to thank my supervisor, Associate Professor Jan Pettersen Nytun, for the thesis project and his wise advice and patience through the project. I also send my regards to Associate Professor Mariann Fossum for helping me get my questions answered and Master Student Lene Laukvik for taking interest and the time to attend our meetings.

Grimstad

June 2013

Bjørn Tore Bjugan

# Contents

<b>Contents</b>	<b>2</b>
<b>1 Introduction</b>	<b>4</b>
1.1 Background . . . . .	4
1.2 Problem statement . . . . .	5
1.3 Problem solution . . . . .	6
1.4 Report outline . . . . .	6
<b>2 World Wide Web Consortiums Semantic Web</b>	<b>7</b>
2.1 Resource Description Framework . . . . .	7
2.2 Resource Description Framework Schema . . . . .	10
2.3 Web Ontology Language . . . . .	11
<b>3 International Classification for Nursing Practice</b>	<b>12</b>
3.1 Introduction . . . . .	12
3.2 Components . . . . .	13
3.3 Composing post-coordinated nursing statements . . . . .	14
3.3.1 Nursing diagnosis and outcome . . . . .	14
3.3.2 Nursing intervention . . . . .	15
3.4 Formal model in Web Ontology Language(OWL) . . . . .	17
<b>4 Solution: ICNP Statement Builder</b>	<b>21</b>
4.1 Specification . . . . .	21

## CONTENTS

---

4.1.1	Data Model . . . . .	23
4.1.2	User interface . . . . .	23
4.2	Key technologies . . . . .	24
4.3	Implementation . . . . .	25
4.3.1	Pacakage: icnpstatementbuilder . . . . .	26
4.3.2	Package: gui_components . . . . .	26
4.3.3	Package: model . . . . .	26
4.3.4	Package: icnp . . . . .	27
4.3.5	Autocompletion and building statements . . . . .	27
<b>5</b>	<b>Discussion and further work</b>	<b>30</b>
5.1	Discussion . . . . .	30
5.2	Contributions . . . . .	31
5.3	Further Work . . . . .	31
	<b>Bibliography</b>	<b>32</b>
<b>A</b>	<b>Original UML Class Diagram</b>	<b>34</b>

# Chapter 1

## Introduction

### 1.1 Background

Nursing practice today demands collecting information in a way that ensures quality of understand-ability for the future use of the information. International Classification for Nursing Practice(ICNP<sup>®</sup>)[7] is a unified language that makes this possible, so that the information, not only increases the quality of the practice locally, but also makes the information ready to be a part of a global information infrastructure. An ongoing work with ICNP among nursing professionals, is making catalogues or subsets of the language terms to fit more specific fields of nursing and local needs. This is done so that education and use of the terms will be more efficient.

By developing the right tools for building ICNP catalogues, the process will be made easier and quality of the work will increase. To help in the development of software tools, the ICNP is described as an ontology with RDF/OWL[8]. Extending this ontology to accept nursing statements with multiple ICNP terms can make it possible to make software with a higher understanding of the correlation between the terms and statements, so that more intelligent and efficient software

can be developed to help the nurses in their practice.

There is already a ongoing project[15] at the University of Agder, between Faculty of Health and Sports Sciences, Faculty of Engineering and Science and The Norwegian Nurses Association to take a closer look at how ICNP can be used to create tools for educational purposes. This project was the inspiration for this master thesis where we try to find out how to use the formal definition of ICNP in OWL, to make software for building nursing statements.

### **1.2 Problem statement**

There is an on-going project at UiA that concerns correct and professional use of nursing terminology in nursing reports. The nursing terminology in question is described in The International Classification for Nursing Practice (ICPN), which is represented as an OWL expressed ontology. An ontology extension to ICPN should be developed; the extension will deal with the nursing process and the way of registering information in nursing reports (a part of the extension will be supplied by the supervisor). Subsets of the ICNP terms can be extracted and help in making composition of terms more efficient. Also whole nursing statements may be put together in catalogues tailored for specific nursing domains. These pre- coordinated complex statements that relates to specific nursing topics are important means to make the terminology more usable at the point of care. The candidate will develop an application (prototype) for making nursing reports. The user interface of the application will be based on input from nursing professionals. The use of catalogues when composing statements should be investigated.

Making new catalogs of ICNP terms require a lot of work and this work increases by the complexity of adding nursing statements that includes several ICNP terms. Making a software that can help make this process easier and require less technical experience, can make progress and increase the use of ICNP further. Because there is little information and experience in making such a software, we

will go for a prototype where we can test our ideas for making catalogs.

### **1.3 Problem solution**

Because of the wide scope of the problem statements description, the solution will focus on building catalogues of terms and statements. A prototype software has partly been made where the user can create new nursing statements and add them to a catalog. By using the ICNP ontology defined in OWL, the software should suggest ICNP terms while the user is typing so that the terms can be chosen and autocompleted. The new catalog data will also be stored in RDF/OWL to keep to the existing technologies already used.

### **1.4 Report outline**

First, in Chapter 2, core technologies in Semantic Web will be explained. This chapter is from my(the authors), last report about Semantic Web in Social Networking(2012.12). In Chapter 3, there will be explained what the International Classification for Nursing Practice is and how it works. Then, in Chapter 4, the solution will be documented and gone through in detail. The last chapter will be a discussion on how the resulting solution was and what further work can be done in the future.



## **Chapter 2**

# **World Wide Web Consortiums Semantic Web**

The World Wide Web Consortium has had a key role in the development of the Semantic Web[1]. They made the specifications and tools for making machine readable and structured data with semantics. Some of these tools are the Resource Description Framework(RDF), the Resource Description Framework Schema(RDFS) and the Web Ontology Language(OWL). They are needed to make the ontologies which we use to store our semantic data. In Figure 2.1 of the Semantic Web stack, you can see where these technologies belong in the stack.

### **2.1 Resource Description Framework**

RDF is used to make a model or graph of the data. It's built up by statements that says for example that "Bjørn Tore likes University of Agder". Bjørn Tore is then the subject, "likes" is the predicate and "University of Agder" is the object in this statement. The subject, predicate and object are entities in RDF and are referred to with their Uniform Resource Identifiers(URI) or unique identifiers. While the

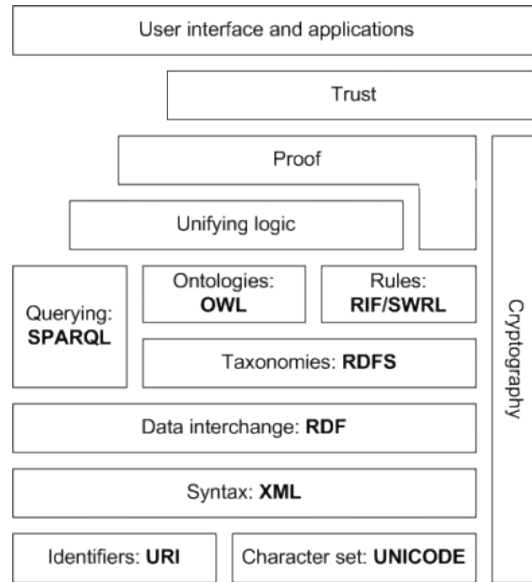


Figure 2.1: The Semantic Web Stack[2]

object can be both a identifier(as a string) and a resource, the predicate and subject can only be a URI. RDF has two parts, the RDF data model specification for defining data models and a serialisation syntax to transport RDF data. The serialisation is often done in RDF/XML, but Notation 3[3] and Turtle[4] is also commonly used syntaxes. The statement in the example earlier consist of two entities and a binary relationship, which is also called a triple.

In this example you see the statement from earlier drawn as a graph.



Figure 2.2: RDF graph

Here you see the statement we made as RDF/XML. We also use the XML namespace mechanism to abbreviate the URIs.

Listing 2.1: RDF/XML example.

---

```
<rdf:RDF
  xmlns="http://www.SemanticWeb.org/#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <Person rdf:about="http://bjugan.no/memebers#Bjorn Tore">
    <likes>
      <rdfs:Resource rdf:about="http://www.uia.no"/>
    </likes>
  </Person>

</rdf:RDF>
```

---

## 2.2 Resource Description Framework Schema

Resource Description Framework Schema is used to define the primitives that describes classes, instances and relationships. It is defined by RDF and is used to describe RDF vocabularies.

Here you see the ontology for our previous example represented in RDF Schema. The "rdfs:" namespace prefix is used as an abbreviation for the URI "http://www.w3.org/2000/01/rdf-schema#".

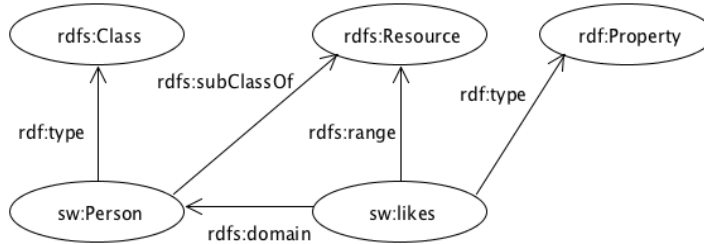


Figure 2.3: Our example ontology represented in RDFS

Here you see the object from our statement, sw:Person of type Class and as a sub class of the Resource class. The property sw:likes is in the domain of the sw:Person class and is of type property. The range is also defined as a Resource.

RDFS also defines other primitives; rdfs:label, rdfs:comment, rdfs:subPropertyOf, rdfs:seeAlso and rdfs:isDefinedBy. The last two is interesting because you can link to other ontologies and related properties or classes.

## 2.3 Web Ontology Language

Because RDFS can't describe all advanced axioms, as for example symmetric or transitive properties, W3C came up with OWL. The semantics in OWL are more expressive than RDFS since it is based on Description Logics[5]. OWL extends on RDFS's classes and properties, and adds new axioms to define advanced characteristics and constraints. One important property is `owl:sameAs`, which defines that two URI references refer to the same thing. OWL has three sub-languages with different degrees of expressivity; OWL-Lite, OWL-DL, OWL-Full. December 11, 2012 a new recommendation[6] from W3C came for OWL 2 specifications which adds new features.

## Chapter 3

# International Classification for Nursing Practice

### 3.1 Introduction

ICNP[7] is a project by the International Council of Nurses[9] which describes ICNP like this[10].

*“As an international standard, ICNP® facilitates collection and analysis of nursing data across populations, health care settings, languages and geographic regions. Data generated by using ICNP® can support decision-making, thus enhancing the safety and quality of care for patients and families.”*

ICNP defines over 3000 terms used in the nursing practice, both specific nursing terms and commonly used words which can for example be used to make electronic health records(EHR). It is meant to be a part of the global information infrastructure that helps health care practice and policy to improve the quality of patient care. With a new ICNP release in 2013 there will also come pre-defined

nursing statements, but since these have not been released yet, we will have to make our own decisions in matters where we could have learned from the new releases. New releases are made available every two years in correlation with the ICN Conference. It has currently been translated from English into 14 languages. ICNP is comprehensive and includes thousands of terms and definitions. By developing catalogues as subsets of ICNP, with nursing diagnosis, intervention and outcome statements for a specific health care and patient groups. It will be easier for nurses working in specialty areas to make use of ICNP. Examples of already available catalogues is Community Nursing and Pediatric Pain Management. This chapter will explain concepts in the ICNP that are in context with development of the respective software.

## 3.2 Components

The ICNP entities(terms and statements) includes properties that are used to identify and associate them with concepts in ICNP. Some of these properties are:

- Knowledge name that are a one word pseudo-English identifier.
- Concept code, which is a unique check-sum defined by ICN.
- Preferred term in the respective language.
- Concept description that describes the term in more detail if needed.
- Axis that specify if the entity is a part of the pre-coordinated statements
  - IC - Intervention Concept
  - DC - Diagnosis Concept

or if it belongs in the post-coordinated 7-axis model for terms

- F - Focus

- J - Judgement
- A - Action
- M - Means
- L - Location
- C - Client
- T - Timing

The axis property is interesting because IC and DC defines entities as a pre-coordinated statements, as opposed to post-coordinated statements that are put together by single terms that belong to the ICNP 7-axis model. ICNP covers statements for nursing diagnosis, that may also be nursing outcomes, and nursing interventions. The context of these nursing statements is not represented by ICNP. So it is up to the user to define the context.

### **3.3 Composing post-coordinated nursing statements**

ICNP has guidelines for composing diagnosis(and outcome) statements and intervention statements. It is based on ISO standard 18104:2003, Integration of a Reference Terminology Model for Nursing (Saba et al.2003). This is applicable for the catalog building process and when making software tools that can help nurses use and learn the terms.

#### **3.3.1 Nursing diagnosis and outcome**

Nursing diagnosis is a statement the nurse makes of the status of the patient. After making a nursing intervention based on the nursing diagnosis, the nursing outcome is the result of the nursing interventions, or the new updated nursing diagnosis. A nursing diagnosis and outcome statement must include a term



from the focus axis and judgement axis. Other than that, the statement can include terms from any of the axis, even those already included. Figure 3.1, from the ISO standard for Reference Terminology Model for Nursing Diagnoses, shows how the focus and judgement terms is central in a nursing diagnosis and outcome statements.

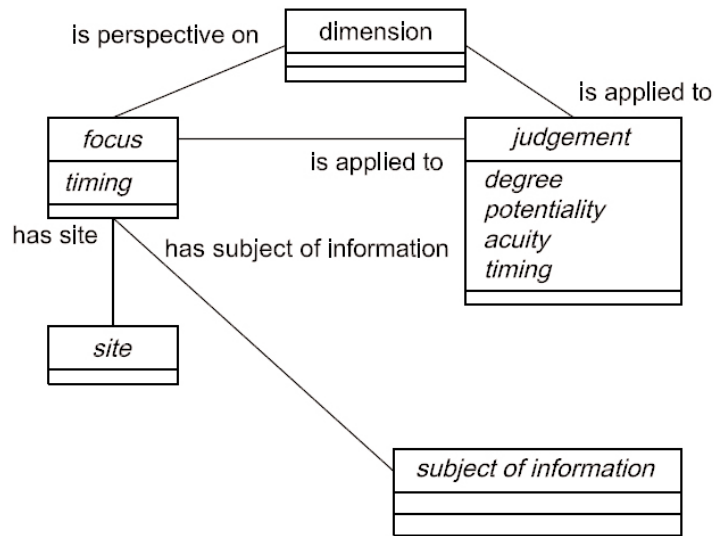


Figure 3.1: ISO Reference Terminology Model for Nursing Diagnoses[10]

### 3.3.2 Nursing intervention

Nursing intervention is the action taken based on the nursing diagnosis. A nursing intervention statement must include a term from the action axis and a target term which is from any of the axis, except the judgement axis. Other than that it can include terms from any of the axis. Figure 3.2, from the ISO standard for Reference Terminology Model for Nursing Diagnoses, shows the term from the action axis that acts on a target term.

### CHAPTER 3. INTERNATIONAL CLASSIFICATION FOR NURSING PRACTICE

---

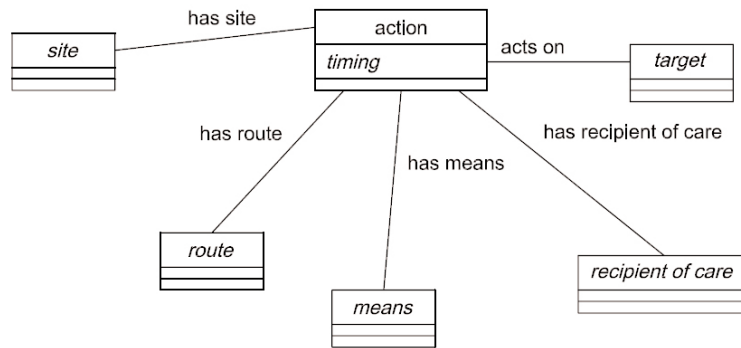


Figure 3.2: ISO Reference Terminology Model for Nursing Actions[10]

## CHAPTER 3. INTERNATIONAL CLASSIFICATION FOR NURSING PRACTICE

---

Listing 3.1: The ICNP Wound class in OWL/RDF XML format. Copied from the ICNP2011.owl file.

---

```
<owl:Class rdf:about="http://www.icn.ch/icnp#Wound">
  <PreferredTerm
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Wound</PreferredTerm>
  <ConceptCode rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >10021178</ConceptCode>
  <Version rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >1.0</Version>
  <ConceptDescription
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Impaired Structural Body Part: Lesion of the tissue usually
    associated with physical or mechanical damage; sloughing and
    tunnelling of tissue; serous, sanguineous or purulent drainage;
    skin erythema; oedema; blistered, macerated and abnormal skin,
    elevated skin temperature, wound odour, soreness and pain.
  </ConceptDescription>
  <rdfs:subClassOf>
    <owl:Class
      rdf:about="http://www.icn.ch/icnp#NegativeStructuralBodyPart"/>
    </rdfs:subClassOf>
    <Axis rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >F</Axis>
  </owl:Class>
```

---

### 3.4 Formal model in Web Ontology Language(OWL)

The RDF(Resource Description Framework)/OWL[8] representation of ICNP comprises classes with associated properties and restrictions which are arranged in a hierarchy. In Listing 3.1, the RDF/OWL XML format of the ICNP Wound class is shown. You can see the same components as mentioned in Section 3.2 and also the `rdf:subClassOf` property that makes the ICNP class `Wound` a sub class of the ICNP class `NegativeStructuralBodyPart`.

## CHAPTER 3. INTERNATIONAL CLASSIFICATION FOR NURSING PRACTICE

---

By opening the ICNP OWL file in a program called Protégé[11], it is possible to browse the ICNP OWL hierarchy. Figure 3.3 shows the same data that the was in Listing 3.1.



Figure 3.3: Screenshot from Protege showing the annotations windows of the ICNP class Wound.

## CHAPTER 3. INTERNATIONAL CLASSIFICATION FOR NURSING PRACTICE

---

Looking further, Figure 3.4 of Protégé shows the complete browsable hierarchy and how the ICNP class Wound is a sub class of ICNP class NegativeStructuralBodyPart.

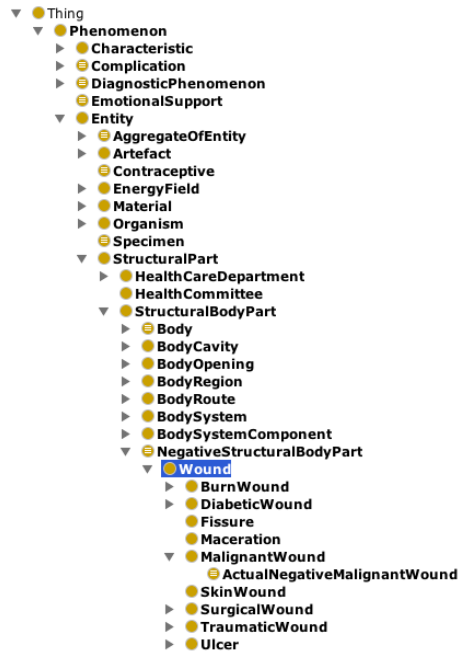


Figure 3.4:

## CHAPTER 3. INTERNATIONAL CLASSIFICATION FOR NURSING PRACTICE

---

Protégé also supports a graph view of the OWL data as seen in Figure 3.5.



Figure 3.5: Screenshot from Protege showing the diagram of the ICNP class Wound

Protégé was used to create the RDF/OWL representation of the ICNP and is a powerful tool for anyone that want to create and explore existing ontologies created in RDF/OWL.

# **Chapter 4**

## **Solution: ICNP Statement Builder**

The prototype software for the solution is called ICNP Statement Builder. Even though it is not a finished product, it shows how different technologies can be used to make a tool for building catalogues with nursing statements and terms. This chapter will explain the details behind how the prototype software is implemented and how the author planned to go forward to complete the development of the software.

### **4.1 Specification**

The prototype software should be a tool for building catalogues of statements and terms. This means both adding data and browsing it. Functionality for importing and exporting catalogues will be important for the usability of the software so that the catalogues can be worked on and used by different users.

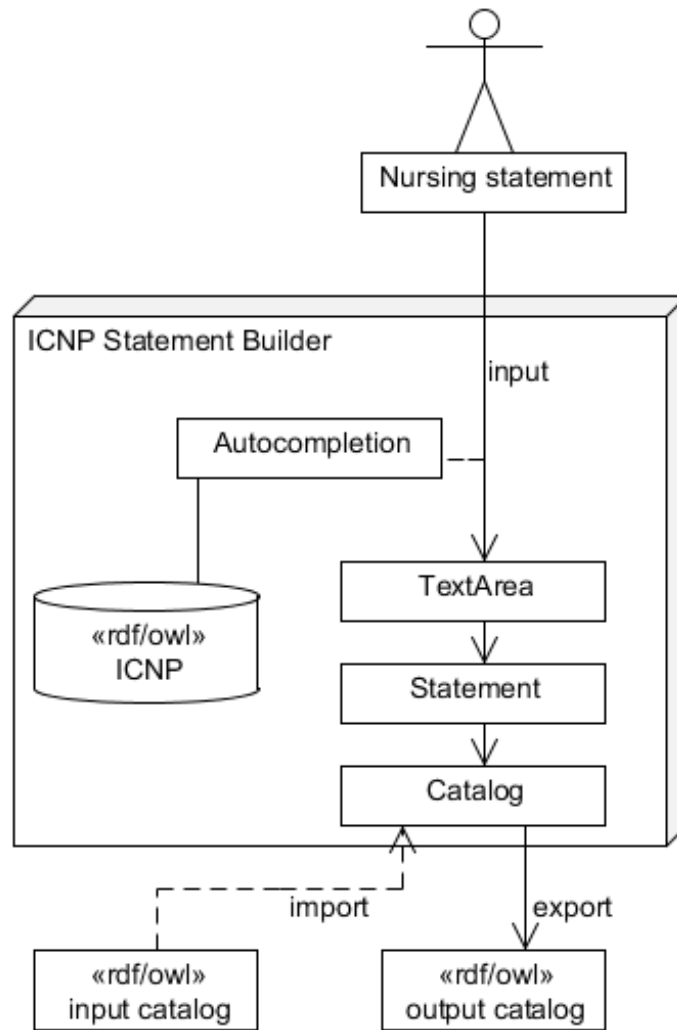


Figure 4.1: Components

Figure 4.1 shows the components in the software. There is a nurse that types in nursing statements in to a textarea. While typing the nurse will get suggestion for terms that matches from the complete ICNP definition. The text in the textarea is then stored as a statement in a catalog, which in turn can be exported. There should also be an import option to work with existing catalogues.



First we will specify data models for the data we are about to work with and then define specifications for the prototypes user interface.

### 4.1.1 Data Model

A catalog will, as specified by ICNP[16], consist of statements and terms. Statements can consist of one or several terms where the term can be a specified ICNP term or not. Figure 4.2 shows the data model for the catalog structure in the software.

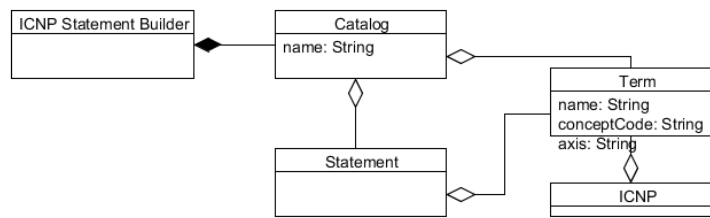


Figure 4.2: UML Class diagram

### 4.1.2 User interface

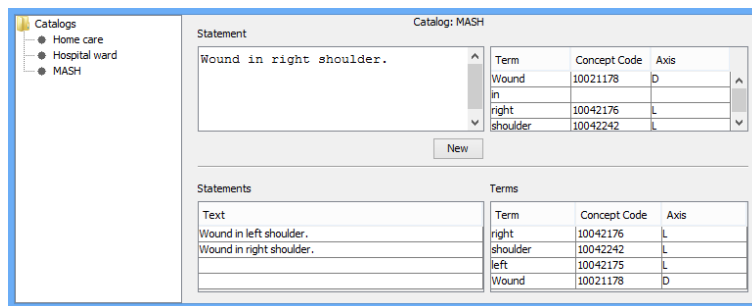


Figure 4.3: GUI proposal

This software will be handling textual data, as in catalogues with terms and statements. So there will have to be a way to browse between these catalogues. A tree view with a hierarchy of data will be fitting for this, especially in a prototype software where we do not have experience with user interfaces for this kind of software. This mean we can add new depths to the hierarchy without too much change in the user interface and the implementation. When choosing a catalog from the data tree, the terms and statements in the respective catalog should be listed. A text area for adding new terms and statements is also needed. This text area should have a autocompletion functionality, so it will suggest terms from ICNP while the user is writing. A button for removing selected statements and terms should also be added.

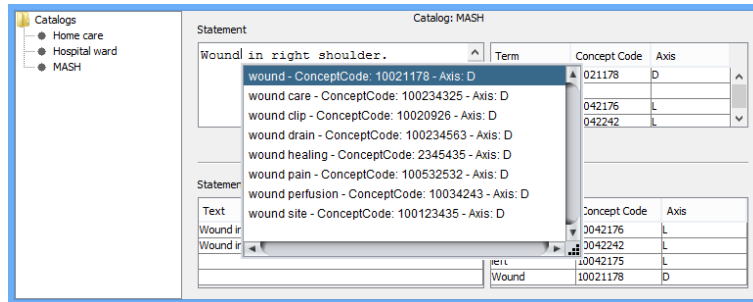


Figure 4.4: GUI proposal with autocompletion

## 4.2 Key technologies

The International Classification for Nursing Practice(ICNP) is formally defined as an ontology in OWL/RDF. So it was logical to find a framework for handling this kind of data. Apache Jena[12] is one such framework available as Java libraries. Because the author has experience with Java, and Apache Jena is developed by Apache, a well known software development foundation, this framework was chosen. For the autocompletion there is used a open source library[13] from Fife

Soft[14]. Both Jena and Fife Softs autocomplete is available in Apache Maven, a software project management tool which can be used to download libraries and add dependencies to projects in for example Netbeans. Netbeans is the chosen IDE used for this project.

### 4.3 Implementation

For the implementation of this software, a Model-View-Controller(MVC) architecture is applied and this is how the java packages is divided to achieve this architecture.

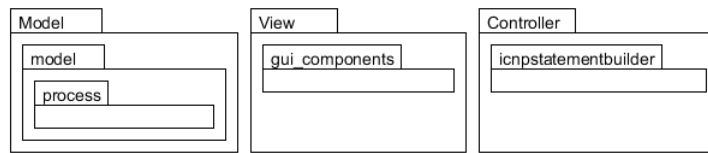


Figure 4.5: Package diagram

In the problem statement, there was described functionality for a nursing process which was planned to implement as a simple journaling system that used the ICNP catalogues in the software, to practice writing electronic health records and learn the terms and statements. This was not the main focus, but there was created classes and code to support this kind of functionality. It will not be documented any further in this report, but it will not be removed from the software source code. The complete class diagram with the journaling system is included in Appendix A. The following subsections will explain key classes and parts of the different packages. Then the functionality for autocomplete and building statements and catalogs will be explained.

### **4.3.1 Pacakage: icnpstatementbuilder**

This is the controller part in the MVC architecture. It includes the important class named Controller that takes care of initializing instances of everything needed by the software. This includes graphical interfaces(View in MVC) and data models(Model in MVC). It also has methods for actions we need the software to do, like adding new catalogs and showing them. Future actions will be exporting and importing catalog data.

### **4.3.2 Package: gui\_components**

As the view part, this includes all graphical user interface components in the software. The DataTree class which extends the TreeSelectionListener and is added as a listener to our JTree component is a key component in the graphical user interface and represents the hierarchical data browser. It takes care of what happens when we select the different nodes in the data hierarchy. Another small but important class is the StatementTextArea, it takes the a class that implements DocumentListener as one of its parameters, which is in this case the Catalog class. This way the Catalog class will have control of every change in the text area where the statements are typed. More on how the changes are handled in the next subsection.

### **4.3.3 Package: model**

The model package, which is the Model part in our MVC architecture, consist of classes representing the data models that is specified and some classes needed specifically for the implementation. CatalogImpl is one such class that are used to handle the catalogs branch of the JTree component. Catalogs consist of statements, so there is a Statement class and statements is then built up by words, so then there is a Word class. These classes are core concepts in this paper and will be explained in its own subsection along with how the autocompletion is thought

to be implemented.

### **4.3.4 Package: icnp**

This package includes two classes specifically for ICNP data. TranslationCVS-ToRDF is for converting translations of the ICNP terms from a CSV(Comma-separated values) file format which is the format it is being distributed in, to RDF, which is the format this software is already working with. Keeping to the same schema as the official ICNP OWL format means we can change between language used in this software by choosing either file. It should be noted that this is only the case as long as non-complex select queries are made towards this RDF data. This is because the translations RDF does not have the OWL schema data that the English official ICNP OWL/RDF format have. The ICNP class is for loading RDF/OWL data from a file and making queries with SPARQL.

### **4.3.5 Autocompletion and building statements**

Because the autocompletion feature should look and behave much like similar feature found in text editors for programming, a open source library from Fife Soft was chosen. The library is called Autocomplete and is a part of a bundle with another library from Fife Soft called RSyntaxTextArea. This is a TextArea with extended functionality for text editors with syntax templates and autocompletion. Autocomplete is a independent library that works with the standard TextArea that comes with Oracle's Java. Because the software only need the autocompletion feature there is only need for the Autocomplete library. To implement this, an instance of the AutoCompletion class is created before it calls its install method with the textarea component as the parameter. Now, the textarea that the auto-completion will work on, is defined. Then the autocompletion provider needs to be specified. This is a dictionary for the autocompletion to look in for words

to suggest to the user. By extending the Statement class with DefaultCompletionProvider and implementing the getCompletionsImpl() method and giving the Statement class instance as a parameter to the AutoCompletion class constructor, the autocompletion provider is set. Every time getCompletionsImpl() is called, either by hot key(Ctrl + Space) or automatically when something is typed, a query will be run to find ICNP terms that match and a list returned.

The statements does not necessarily consist only of terms from ICNP, and it therefore need to keep track of which terms from ICNP are in the statement and their positions. This means the text input needs to be analyzed and put in a fitting data structure that later can be exported to RDF. Figure 4.6 show the data structure that

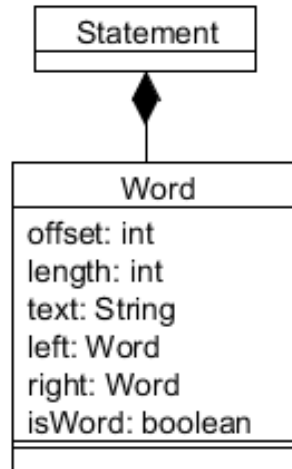


Figure 4.6: UML Class diagram

the statements are stored in in Java. It is a double linked list with Word objects containing words or separators like spaces, commas or periods. Because there can be choices made while creating the statements and concerns of efficiency, a state-full approach to changes in the textarea was chosen. This made it necnecessary to make an algorithm that took care of all cases that can occur while typing in a textarea and therefore a bit big and difficult to implement, but it has a low

Listing 4.1: Example of a statement exported as RDF

---

```

<Statement rdf:about="http://.../#Statement_id">
  <Translation xml:lang="en">
    <rdf:type rdf:resource="http://.../#Seq"/>
    <Term>
      <ConceptCode>10021178</ConceptCode>
      <PreferredTerm>Wound</PreferredTerm>
    </Term>
    <Term>
      <PreferredTerm> </PreferredTerm>
    </Term>
    <Term>
      <PreferredTerm>in</PreferredTerm>
    </Term>
    <Term>
      <PreferredTerm> </PreferredTerm>
    </Term>
    <Term>
      <ConceptCode>100xxx</ConceptCode>
      <PreferredTerm>shoulder</PreferredTerm>
    </Term>
    <Term>
      <PreferredTerm>.</PreferredTerm>
    </Term>
  </Translation>
</Statement>

```

---

complexity and is efficient. The result is that the statement structure is maintained every time a change is done, as a state-full state-machine should. At the point these decisions were made, a format for statements in catalogues was not yet available, so a simple format was made as shown in the RDF/XML example in Listing4.1.

# Chapter 5

## Discussion and further work

### 5.1 Discussion

The solution, that was to be a prototype software for building ICNP catalogues of statements based on the ICNP, is not completed. So no actual testing of a software solution is possible and therefore, no experience with testing such a tool can be reported. Even though the prototype software is not completed, all the technologies needed was tested and found suitable for the planned solution. The technologies mentioned is more or less implemented in the software, but features binding them together is not completely implemented.

The framework for working with RDF/OWL, Jena, worked well and was a natural match with Java. Other programming languages could have been used, but for a system developer with previous experience with Java, I feel this was the best choice.

The ICNP project seems like a quality project for the future, but that may also be the reason why they do not make everything available and easy to access on the Web. To get access to the ICNP data you have to request it from someone on the inside of the community already working with it. It may also be a bit closed



because they are still finding out how it should work, which was apparent when the technical implementation guide[17] wasn't released until May 21st 2013. At the same time the newest release of ICNP was published. A lot of changes have been done and this project could have gone further if this new information had been available earlier.

## **5.2 Contributions**

With more work and research, this kind of software solutions can help the ICNP project expand and also make it easier for professionals to learn and use the terminology.

## **5.3 Further Work**

This field has a lot of future work to be done. Completing the software solution and testing it on nursing professionals would be interesting. Also making the software more intelligent by implementing the rules specified in the Guidelines for ICNP Catalogue Development. This will help the user input correct nursing statements and will be a leap in the right direction.

# Bibliography

- [1] [Online]. Available: <http://www.semanticweb.org>
- [2] [Online]. Available: [http://en.wikipedia.org/wiki/Semantic\\_Web\\_Stack](http://en.wikipedia.org/wiki/Semantic_Web_Stack)
- [3] (2011, 3). [Online]. Available: <http://www.w3.org/TeamSubmission/n3/>
- [4] (2011, 3). [Online]. Available: <http://www.w3.org/TeamSubmission/turtle/>
- [5] (2012, 1). [Online]. Available: <http://arxiv.org/abs/1201.4089>
- [6] (2012, 12). [Online]. Available: <http://www.w3.org/TR/owl2-overview/>
- [7] (2013, 05). [Online]. Available: <http://www.icn.ch/pillarsprograms/international-classification-for-nursing-practice-icnpr/>
- [8] (2013, 05). [Online]. Available: [http://www.w3.org/2001/sw/wiki/Main\\_Page](http://www.w3.org/2001/sw/wiki/Main_Page)
- [9] (2013, 05). [Online]. Available: <http://www.icn.ch/>
- [10] (2013, 05). [Online]. Available: <http://www.icn.ch/pillarsprograms/icnpr-catalogues/>
- [11] (2013, 05). [Online]. Available: <http://protege.stanford.edu/>
- [12] (2013, 05). [Online]. Available: <http://jena.apache.org/>
- [13] (2013, 05). [Online]. Available: <http://fifesoft.com/autocomplete/>
- [14] (2013, 05). [Online]. Available: <http://fifesoft.com/>

## BIBLIOGRAPHY

---

- [15] K. M. A. P. Jan Pettersen Nytnun, Mariann Fossum, “The icnp in nursing education,” 03 2013, dRAFT 03.03.2013.
- [16] I. C. of Nurses, *GUIDELINES FOR ICNP CATALOGUE DEVELOPMENT*, International Council of Nurses, 2008.
- [17] —, “Icnp: Technical implementation guide,” 05 2013.

# Appendix A

## Original UML Class Diagram

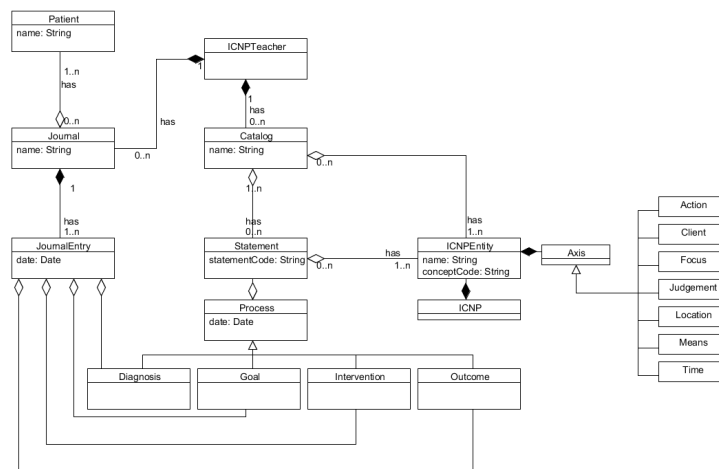


Figure A.1: Original UML Class Diagram