

# MASTERARBEIT

## MapFace

### A Graphical Editor for MetaMap Transfer (MMT<sub>x</sub>)

Ausgeführt am Institut für  
Softwaretechnik und Interaktive Systeme  
der Technischen Universität Wien

unter der Anleitung von  
ao.Univ.Prof. Mag. Dr. **Silvia Miksch**  
und Mag. Dr. **Katharina Kaiser**

durch **Theresia Gschwandtner**

  
  
Matr.Nr.: 9949197

## Kurzfassung

Medizinische Leitlinien zu formalisieren und zu modellieren, sodass sie automatisiert verarbeitet und ausgewertet werden können, ist eine komplexe, aber sehr wichtige Aufgabe, da die automatische Ausführbarkeit von Leitlinien die Anwendung des best-möglichen medizinischen Wissens bei der Patientenversorgung ermöglicht, ohne diese aufzuhalten. Bevor aber der medizinische Text in eine solche Form gebracht werden kann, ist es wichtig, ihn so aufzubereiten, dass sein Inhalt, d.h. die enthaltenen medizinischen Konzepte, eindeutig identifiziert und beschrieben werden, um eine korrekte Interpretation zu gewährleisten.

Semantische Annotationssysteme erkennen die medizinischen Konzepte in Leitlinien und ordnen ihnen entsprechende Konzepte aus medizinischen Terminologien, wie z. B. dem UMLS® Metathesaurus®[36] zu, die wiederum notwendige zusätzliche Informationen enthalten. Aufgrund der Mehrdeutigkeit von freisprachlichen Texten kann die korrekte automatische Identifikation und Zuordnung von medizinischen Konzepten mit Hilfe dieser Systeme vermutlich nie vollkommen gewährleistet werden. Da aber gerade auf einem so sensiblen Gebiet wie der Medizin die absolute Fehlerfreiheit von Ergebnissen maßgeblich ist für ihre weitere Verwendbarkeit, ist es unbedingt notwendig, dass diese von ExpertInnen kontrolliert und gegebenenfalls korrigiert werden.

Daher habe ich einen Editor für das MetaMap Transfer (MMTx) Programm [2, 3] entwickelt, der medizinischen ExpertInnen diese Arbeit ermöglicht, ohne spezielle Kenntnisse auf dem Gebiet der Informationsverarbeitung vorauszusetzen. **MapFace** wurde entworfen, um sowohl die Bearbeitung der Ergebnisse von MetaMap auf einfache Weise zu gewährleisten, als auch die Verfügbarkeit und Visualisierung aller verknüpften Informationen auf einen "Maus-Klick" anzubieten.

## Abstract

It is a complex but very important task to formalize a clinical practice guideline and model it into a representation that can be executed and utilized automatically, since the automatic execution of guidelines at the point-of-care ensures the best available scientific evidence without interrupting clinical workflow. Before a medical text can be translated into such a model, it is necessary to pre-process the text, so that its contents, i. e. the existing medical concepts, can be identified and described unambiguously in order to ensure a correct interpretation.

Semantic Annotation Systems extract medical concepts from the text of guidelines and map them to concepts from medical terminologies, such as the UMLS®Metathesaurus®[36], which contain important additional information. Due to the ambiguity of free text, the correct and automatic identification of medical concepts and the corresponding mapping generated with the help of these systems will probably never be completely correct. Since medical care is an extremely sensitive discipline, the complete reliability of results is crucial for their usability for further processing, which makes it absolutely necessary for experts to control these results and to modify them, if necessary.

This fact led me to develop an editor for the MetaMap Transfer (MMTx) program [2, 3] that enables experts in medical science to solve this task without requiring special skills in information processing. **MapFace** was designed to realize an easy way to edit the MetaMap results as well as to provide access to all assigned information by a single "mouse-click" in combination with a clearly arranged visualization of the acquired information.

## **Acknowledgements**

This work is supported by "Fonds zur Förderung der wissenschaftlichen Forschung FWF" (Austrian Science Fund), grant L290-N04.

# Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
1.1	Motivation . . . . .	11
1.2	Used Terminology . . . . .	14
<b>2</b>	<b>Related Work</b>	<b>16</b>
2.1	Clinical Practice Guidelines (CPGs) . . . . .	16
2.1.1	Development of Guidelines . . . . .	16
2.1.2	Protocols . . . . .	17
2.1.3	Computerized Guidelines . . . . .	17
2.2	Information Extraction . . . . .	18
2.2.1	Information Retrieval and Text Understanding . . . . .	18
2.2.2	Components of an Information Extraction System . . . . .	18
2.3	UMLS®- Unified Medical Language System® . . . . .	20
2.3.1	The Metathesaurus® . . . . .	20
2.3.2	The Semantic Network . . . . .	21
2.3.3	The SPECIALIST Lexicon and the SPECIALIST NLP Tools . . . . .	21
2.4	The MetaMap Program . . . . .	21
2.5	Other Biomedical Semantic Annotation Systems . . . . .	22
<b>3</b>	<b>Analysis and Realization</b>	<b>24</b>
3.1	Problem Analysis . . . . .	24
3.2	Aim of MapFace . . . . .	26
3.2.1	Processing the text of a guideline with MMTx . . . . .	29
3.2.2	Correcting phrase chunks . . . . .	29
3.2.3	Correcting concept chunks . . . . .	29
3.2.4	Assigning UMLS concepts to concept chunks . . . . .	33
3.2.5	Assigning semantic types to phrase chunks . . . . .	33
3.2.6	Providing Information about medical concepts in the text . . . . .	36
3.2.7	Providing Information about phrases in text . . . . .	36
3.2.8	Highlighting certain semantic types . . . . .	36
3.2.9	Displaying concept relations . . . . .	36
3.2.10	Displaying semantic relations . . . . .	36
3.3	Realization . . . . .	37
3.3.1	Resources used by MapFace . . . . .	37
3.3.2	User Interface . . . . .	37

3.3.3	Modes . . . . .	42
3.3.4	Features . . . . .	44
3.3.5	Example . . . . .	48
3.3.6	Extendability . . . . .	54
<b>4</b>	<b>Evaluation</b>	<b>57</b>
4.1	Test Scenarios . . . . .	57
4.1.1	Results of the MMTx Program . . . . .	57
4.1.2	Correcting phrase chunks . . . . .	58
4.1.3	Correcting concept chunks . . . . .	59
4.1.4	Assigning UMLS concepts to concept chunks . . . . .	60
4.1.5	Assigning semantic types to phrase chunks . . . . .	66
4.2	Usability Testing . . . . .	66
4.2.1	Testing Sessions . . . . .	66
4.2.2	Result . . . . .	66
<b>5</b>	<b>Conclusion</b>	<b>68</b>
5.1	Summary . . . . .	68
5.2	Further Work . . . . .	68
<b>A</b>	<b>Use Cases</b>	<b>70</b>
A.1	Process the text of the guideline with MMTx . . . . .	70
A.2	Correct the results of the MMTx program . . . . .	71
A.3	Correct phrase chunks . . . . .	71
A.4	Correct concept chunks . . . . .	72
A.5	Assign UMLS concepts to concept chunks . . . . .	73
A.6	Remove a UMLS concept from the candidates list . . . . .	74
A.7	Search for additional UMLS concept candidates . . . . .	74
A.8	Assign semantic types to phrase chunks . . . . .	74
A.9	Information about concepts in the text . . . . .	75
A.10	Information about phrases in text . . . . .	76
A.11	Highlight concepts with certain semantic types . . . . .	77
A.12	Highlight phrases with certain semantic types . . . . .	77
A.13	Display concept relations . . . . .	78
A.14	Display semantic relations . . . . .	78
<b>B</b>	<b>FAQ</b>	<b>79</b>
B.1	How can I tell MapFace to structure my XML document? . . . . .	79
B.2	How can I change the colors MapFace uses to highlight the text? . . . . .	79
B.3	How can I make MetaMap process only certain parts of the document? . . . . .	80
B.4	How can I make MetaMap process the whole text of the document at once? . . . . .	80
B.5	How can I find out which medical concepts were detected by the MetaMap program? . . . . .	80
B.6	How can I find out to which concepts / phrases an UMLS concept / semantic type could not unambiguously assigned? . . . . .	80
B.7	Do I need to control only the concepts with more than one candidate? . . . . .	80
B.8	What does it mean, if the background of a concept/phrase is gray? . . . . .	81

B.9	How can I highlight all concepts/phrases of the same semantic type? . . . . .	81
B.10	How can I find out which candidate is assigned to the concept/phrase? . . . . .	81
B.11	How can I highlight certain XML elements of the XML document? . . . . .	81
B.12	How can I choose and assign a UMLS concept to a concept chunk of the guideline text? . . . . .	81
B.13	How can I change the UMLS concept candidate assigned to a concept chunk? . .	82
B.14	What information about concept candidates is available to facilitate my decision?	82
B.15	What, if the correct candidate for a concept does not appear in the candidates list?	82
B.16	How can I find out to which phrase a concept chunk belongs? . . . . .	82
B.17	Why is it not possible to look for additional candidates for phrases? . . . . .	82
B.18	How can I modify the beginning and end of a concept/phrase? . . . . .	83
B.19	What, if a message "No valid text selected to add a concept/phrase" appears, when I want to create a concept chunk/phrase chunk? . . . . .	83
B.20	Will all the information computed with the help of the MMTx program still be available when I save the document and continue work later? . . . . .	83
B.21	Why are the buttons to edit the concepts and phrases disabled? . . . . .	83
B.22	What does it mean, if the semantic types don't show up in the <i>semantic collections</i> <i>view</i> after opening the MapFace editor? . . . . .	84

**Bibliography** **85**

# List of Figures

1.1	A possible sentence of a guideline. . . . .	11
1.2	The MMTx program returns the input sentence tokenized into phrases and a set of best matching UMLS concepts for each medical concept identified within the sentence. . . . .	11
1.3	Wrong or ambiguous MMTx results are to be corrected by means of the MapFace editor. . . . .	12
1.4	The phrase chunk "with mild asthma inhaled steroids" has been split. . . . .	12
1.5	Appropriate UMLS concepts have been selected in case of ambiguity. . . . .	13
1.6	The concept chunks "five" and "years" where merged to a single concept chunk and an appropriate UMLS concept has been assigned. . . . .	13
1.7	A semantic type has been assigned to each phrase chunk containing a medical concept. . . . .	14
3.1	A physician's options to work with MapFace. . . . .	28
3.2	The workflow of creating MMTx result with the MapFace editor. . . . .	30
3.3	The workflow of correcting phrase chunks. . . . .	31
3.4	The workflow of correcting concept chunks. . . . .	32
3.5	The workflow of assigning UMLS concept candidates to concept chunks. . . . .	34
3.6	A knowledge engineer's options to work with MapFace. . . . .	35
3.7	Components of the GUI. . . . .	37
3.8	The toolbar. . . . .	39
3.9	Candidates view. . . . .	40
3.10	The <i>semantic collections view</i> . . . . .	41
3.11	Modes. . . . .	42
3.12	Example of how to process a clinical guideline. . . . .	49
3.13	MapFace after running the MMTx program and highlighting the concept chunks. . . . .	50
3.14	Before and after correcting the phrase chunk "in Scotland.1 Ovarian cancer". . . . .	51
3.15	Correcting phrase chunks (2 different situations). . . . .	51
3.16	Correcting concept chunks. . . . .	52
3.17	Semantic relations of the UMLS concept "Neoplastic Process" to other UMLS concepts affiliated to concept chunks in the same section of the text. . . . .	52
3.18	Assigning a UMLS concept candidate to a concept. Orange nodes indicate optional actions. . . . .	53
3.19	Assigning a semantic type to a phrase. Orange nodes indicate optional actions. . . . .	54



4.1	Before and after splitting the phrase chunk "with no family history the lifetime risk" . . . . .	59
4.2	Before and after merging the phrase chunks "under the age" and "of 30 years" . .	59
4.3	Before and after correcting the concept chunk "family history" . . . . .	60
4.4	The mapping for "woman" is ambiguous. . . . .	61
4.5	After assigning a UMLS concept. . . . .	61
4.6	Before removing a UMLS concept from the candidates list. . . . .	62
4.7	After removing a UMLS concept from the candidates list. . . . .	63
4.8	The candidates list does not contain the correct UMLS concept for the concept chunk "sixth decade" . . . . .	64
4.9	After searching for UMLS concepts matching the text "age" . . . . .	64
4.10	Displaying information about the semantic relations of the UMLS concept "Desease" . . . . .	65
5.1	Examples of additional modes. . . . .	69

# Chapter 1

## Introduction

Clinical practice guidelines (CPGs) and protocols have become an important tool to improve the quality of clinical practice. They provide recommendations on the treatment and care of patients with specific diseases and conditions, they provide access to the best available evidence and they contribute to the development of standards for clinical practice, just to name a few benefits. Computer based applications of clinical guidelines, integrated in clinical data flow, are important as they allow the user to profit from these advantages in an efficient way.

A front end of developing of such an application is the pre-processing of the text, in order to create a mapping of concepts from medical terminologies to medical concepts existing in the text of the guideline. The MetaMap Transfer program [2, 3] is such a semantic annotation system, which automatically maps free text of a guideline to corresponding concepts in the UMLS®Metathesaurus®[36], the largest thesaurus in the biomedical domain. Due to the complexity and ambiguity of free text, it is not always possible to automatically create a complete and correct mapping of a medical document.

This drawback led me to develop the MapFace editor, a Graphical User Interface for the MetaMap Transfer program. Since the absolute reliability of the received information is crucial for further processing steps, it is necessary for experts in medical science to control and correct the annotation of the guideline. Thus MapFace provides a comfortable way for physicians (who in general are not familiar with computer languages or programming) to edit the results of the MMTx program at a syntactical and conceptual level.

Furthermore MapFace provides an easy access to all information provided by MMTx and visualizes this information in a clearly arranged way. By means of this functionality, MapFace supports the correct interpretation of the medical text for knowledge engineers, thus enabling them to correctly convert the semantic information of the guideline.

## 1.1 Motivation

The main object of MapFace is to create a correct mapping of the text of a clinical guideline to medical concepts using the MMTx program. An arbitrary sentence of a guideline could be:

For patients above five years with mild asthma inhaled steroids are the most effective preventer drug.

Figure 1.1: A possible sentence of a guideline.

The MMTx program tokenizes the sentence into phrase chunks and maps the text to medical concepts available in the UMLS Metathesaurus:

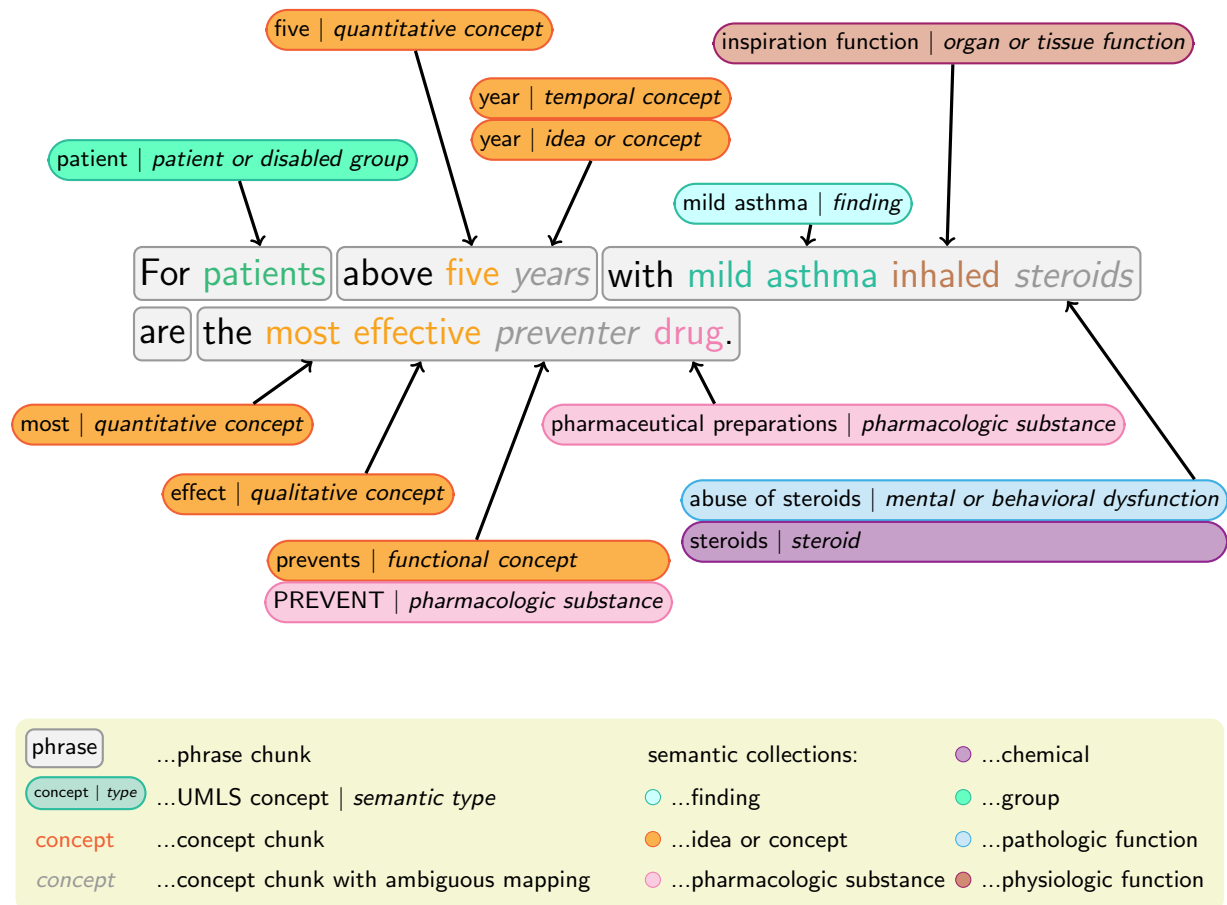


Figure 1.2: The MMTx program returns the input sentence tokenized into phrases and a set of best matching UMLS concepts for each medical concept identified within the sentence.

Corrigenda:

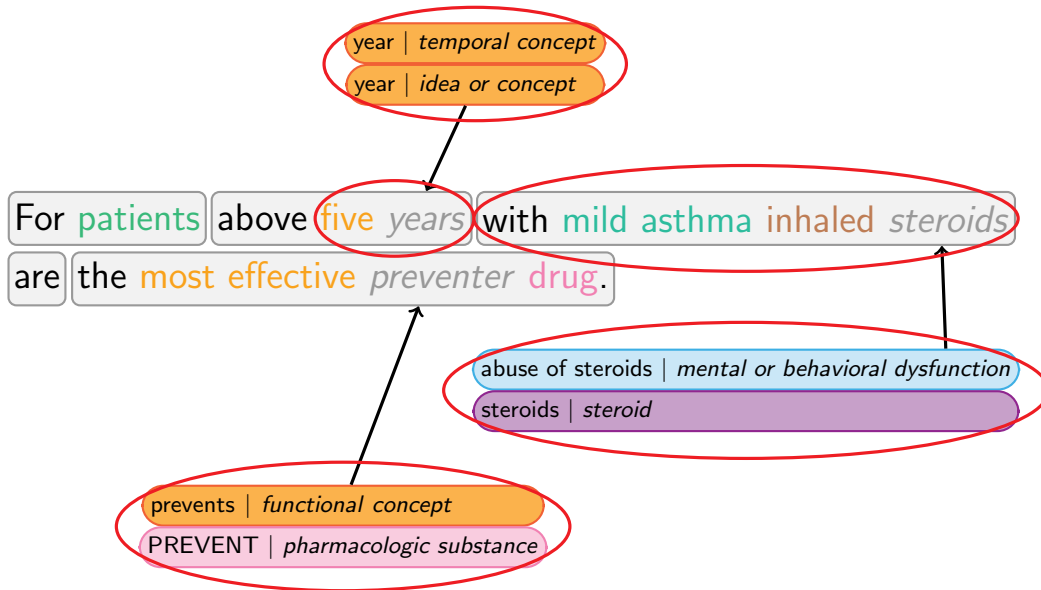


Figure 1.3: Wrong or ambiguous MMTx results are to be corrected by means of the MapFace editor.

As shown above (see Figure 1.3), the MMTx program is not able to create an unambiguous mapping for the text chunks `years`, `steroids`, and `preventer`. In addition, the tokenization of the phrase chunk `with mild asthma inhaled steroids` is not correct and instead of the two concept chunks `five` and `years` we would prefer a single concept chunk with the semantic meaning "age".

We can correct the mapping and tokenization with the help of MapFace. To do so, we delete the wrong phrase chunk and create two new chunks from selected text.

This is what the tokenization looks like afterwards:

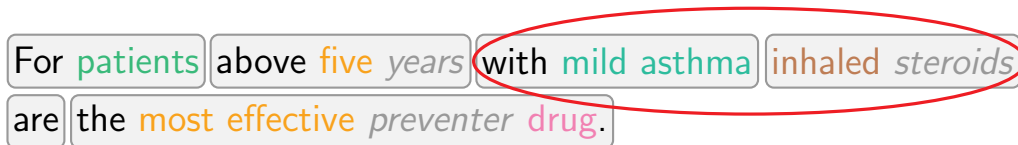


Figure 1.4: The phrase chunk "with mild asthma inhaled steroids" has been split.

Next we need to choose the correct UMLS concept in case of ambiguity and affiliate it to the corresponding concept chunk:

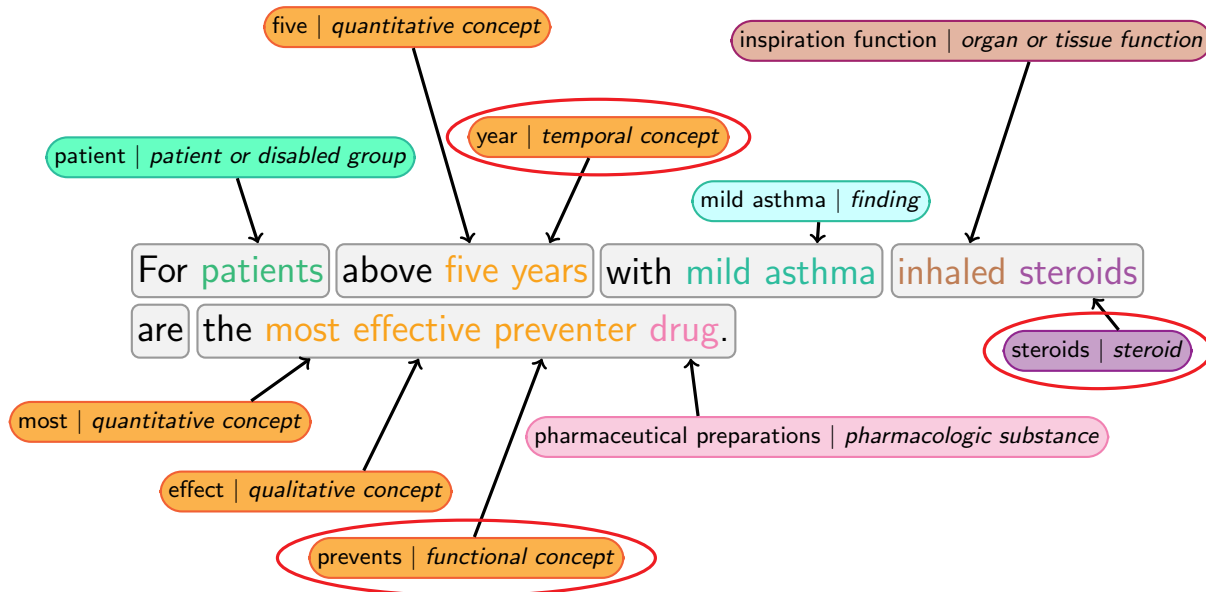


Figure 1.5: Appropriate UMLS concepts have been selected in case of ambiguity.

Still we don't agree with the mapping of the text chunks **five** and **years**. To create a single concept chunk with the semantic meaning "age", we delete the two chunks and create a new one from the text **five years**, whereupon we search for the UMLS concepts matching the text "age" and affiliate the correct match to the created concept chunk:

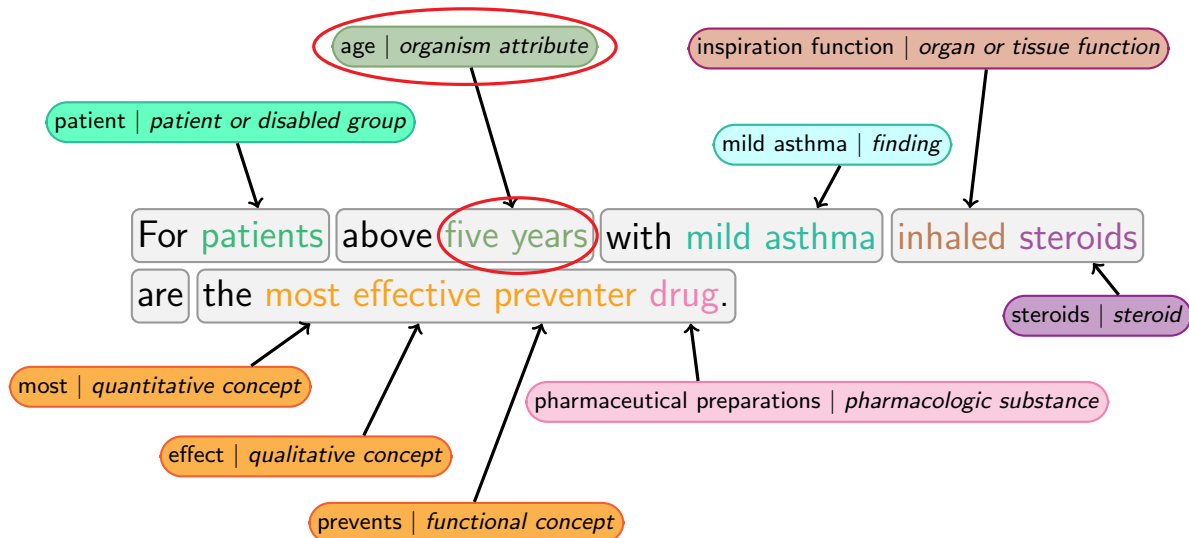


Figure 1.6: The concept chunks "five" and "years" were merged to a single concept chunk and an appropriate UMLS concept has been assigned.

Now, for each concept chunk in the sentence the correct UMLS concept has been found. Last but not least we assign a semantic type to each phrase containing at least one medical concept:

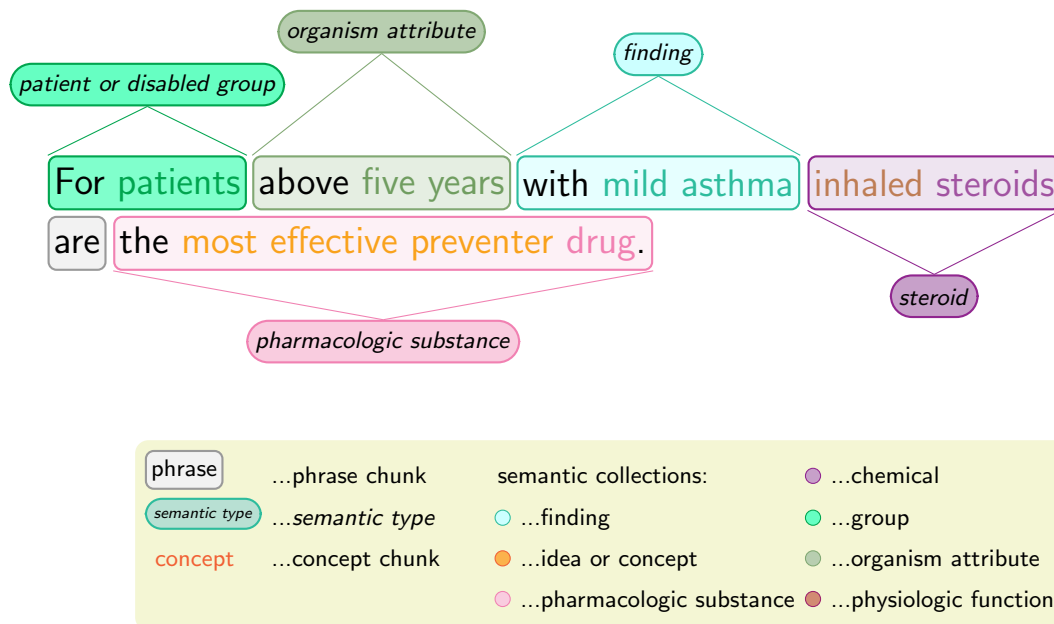


Figure 1.7: A semantic type has been assigned to each phrase chunk containing a medical concept.

## 1.2 Used Terminology

In this thesis I will use the following expressions:

**UMLS concept:** A *UMLS concept* is a medical concept which is part of the UMLS Metathesaurus (see Section 2.3). It is associated with one or more *semantic types*.

**UMLS concept candidate:** The UMLS concept candidates MapFace deals with are the best matching *UMLS concepts* for a given *concept chunk* or *phrase chunk* (see Section 2.4).

**Concept chunk:** A *concept chunk* is a text chunk of a guideline matching one or more *UMLS concepts*.

**Phrase chunk:** A *phrase chunk* is a text chunk of a guideline identified as a phrase by the MMTx program (see Section 2.4). It may contain one *concept chunk*, many *concept chunks*, or no *concept chunk* at all.

**Semantic type:** A *semantic type* describes a medical concept, e.g., the *semantic type* of the medical concept "adrenaline" is "hormone".

**Semantic collection:** A *semantic collection* is a group of semantic types [5], e.g., the *semantic collection* "finding" contains the *semantic types* "finding", "laboratory or test result", and "sign or symptom". MapFace associates each *semantic collection* with a different color.

**Semantic group:** *Semantic groups* arrange *semantic types* [24]. (*Semantic collections*, too, arrange *semantic types*, but they use a different approach.)

**Concept relation:** A *concept relation* is a relation between two *UMLS concepts* obtained from the UMLS database.

**Semantic relation:** A *semantic relation* is a relation between two *semantic types* obtained from the UMLS database.

## Chapter 2

# Related Work

The following subsection specifies basic principles that are essential for the understanding of this thesis. In particular, I will outline the Unified Medical Language System<sup>®</sup>, the MetaMap program, and common biomedical semantic annotation systems to give you an overview of the state of the art.

### 2.1 Clinical Practice Guidelines (CPGs)

”Clinical practice guidelines are systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances.” ([12], p.8)

CPGs are instructions and recommendations on appropriate and effective treatment of patients with specific diseases, with the intention to improve the quality of health care. By providing access to the best available scientific evidence CPGs

- facilitate the decision-making processes in health care,
- set standards to reduce inappropriate variation in health care practice,
- can be used in continuing education and training of physicians,
- serve as reference in quality control,
- encourage further research,
- and promote efficient use of resources [11, 12, 35].

#### 2.1.1 Development of Guidelines

There are several different kinds of guidelines depending on the methodology used to derive the recommendations.

1. **Guidelines based on Expert Opinion:** Early guidelines were developed on the basis of the opinion of experts in special fields. This approach is very inexpensive but due to subjectivity it also involves a great potential of varying quality of the outcome.



2. **Guidelines based on Formal Consensus:** In areas where there is not enough evidence base to derive recommendations, the development of guidelines relies on a consensus process, based on nominal group technique, consensus conferences, or Delphi Technique<sup>1</sup>. The aim of this method is to reach agreement on recommendations.
3. **Evidence-based Guidelines:** Evidence-based guidelines rely on systematic review of clinical evidence. They involve a comprehensive search of the literature, an evaluation of the quality of individual studies, and a system to label the strength of recommendations.

In practice, the majority of guidelines are based on a combination of evidence and consensus [35].

### 2.1.2 Protocols

Clinical protocols are similar to CPGs but defined in greater detail, containing precise instructions as to what must be done.

”Protocols are local tools that set out specifically what should happen, when and by whom in the care process. They can be seen as the local definition of a particular care process derived from a more discretionary guideline. They are in essence tools that assist in quality improvement and reducing inequalities. ... Protocols reflect local circumstances, and variation will due to the differing types of local provision.” [10]

### 2.1.3 Computerized Guidelines

The benefits of using CPGs are numerous and widely recognized, but the availability of patient-specific advice at the point-of-care turned out to be crucial for effectively using guidelines in clinical care [11]. Therefore, several groups and organizations work at the development of computerized guidelines and decision support systems that incorporate these guidelines. Guidelines in a computer-interpretable form offer great benefits, such as

- providing recommendations for individual patients,
- not interrupting clinical workflow,
- improving the clarity of the text,
- helping to reveal inconsistencies and errors in guidelines,
- and providing automatical reminders for treatment or tests [11, 35].

---

<sup>1</sup> The **Delphi Technique** is a method of achieving consensus on controversial topics, developed at the *Rand Corporation* at the beginning of the cold war.

## 2.2 Information Extraction

Being a subfield of Natural Language Processing (NLP), Information Extraction (IE) is a technique to process unstructured natural language text, analyze it, recognize specific types of information, and present the detected information in structured form. Using IE techniques, significant facts about a predefined topic are selected from text documents and usually entered automatically into a database for further processing. Therefore, these relevant facts have to be explicitly present in the text. The analyzation of natural language is necessary when applying IE techniques to unstructured text [1, 14, 20, 31].

”For example, an information extraction system designed for a terrorism domain might extract the names of perpetrators, victims, physical targets, weapons, dates, and locations of terrorist events. Or an information extraction system designed for a business domain might extract the names of companies, products, facilities, and financial figures associated with business activities.” ([31], p. 2)

### 2.2.1 Information Retrieval and Text Understanding

”Information Extraction differs from traditional techniques in that it does not recover from a collection a subset of documents which are hopefully relevant to a query, based on key-word searching (perhaps augmented by a thesaurus). Instead, the goal is to extract from the documents (which may be in a variety of languages) salient facts about prespecified types of events, entities or relationships.” [14]

Information Retrieval (IR) is concerned with identifying passages of text that may contain relevant information or sets of relevant documents. The user still has to analyze the text or documents himself.

Text Understanding is located on the other extreme of text processing. Such systems are expected to extract and represent not only information that is explicitly present in the text, but also implicit information that can be derived from the present explicit information. This is a very complex task that would require a comprehensive representation of a broad spectrum of knowledge, which is still not satisfactory realized yet [1, 14, 20, 31].

### 2.2.2 Components of an Information Extraction System

Usually an IE system contains modules for input tokenization, lexical and morphological processing, parsing, and domain specific analysis of the text [1, 20].

1. **Tokenization:** The tokenization of the text is responsible for splitting it into sentences or tokens, which is not a trivial task for languages like Chinese or Japanese, for which the boundaries of a word are not evident from its orthography. Therefore, the addition of a Word Segmentation module is necessary [1, 20].
2. **Lexical and Morphological Analysis:** Lexical processing aims to determine lexical features of the tokens. The most important step processed by this module is the recognition of proper names. In addition, parts-of-speech tags have to be assigned to each word, which can be accomplished with the help of a lexicon, or by automatic parts-of-speech taggers.

”A parts-of-speech tagger annotates each word of a sentence with its parts-of-speech tag, such as noun, verb, adjective, and so on. It can avoid incorrect analysis ...” ([20], p. 5)

The morphological analysis of the text deals with the structure of word forms. IE systems for languages with simple morphology, like English, do not need a morphological analysis component at all, but for a language like German this component is essential. [1, 20]

3. **Parsing:** The parsing component of the IE system is responsible for analyzing and identifying the syntactic structure of the text. For the English language it is possible to define grammars for simple constructs of the text, such as noun groups or verb groups. The following example shows how a text would be analyzed according to such a grammar:

”[Bridgestone Sports Co.]<sub>NG</sub> [said]<sub>VG</sub> [Friday]<sub>NG</sub> [it]<sub>NG</sub> [has set up]<sub>VG</sub> [a joint venture]<sub>VG</sub> [in]<sub>P</sub> [Taiwan]<sub>NG</sub> [with]<sub>P</sub> [a local concern]<sub>NG</sub> [and]<sub>P</sub> [a Japanese trading house]<sub>NG</sub> [to produce]<sub>VG</sub> [golf clubs]<sub>NG</sub> [to be shipped]<sub>VG</sub> [to]<sub>P</sub> [Japan]<sub>NG</sub>.”  
([1], p.167)

The majority of IE systems just process a shallow, fragment syntactic analysis, because they are only interested in specific types of syntactic information in the text. Other IE systems do not need syntactic analysis at all [1, 20].

4. **Coreference in IE systems:** This module is responsible to handle some problems caused by coreference, such as
  - *Name-alias coreference*, i.e., two or more terms occur that stand for the same name, e.g., ”Resi” and ”Theresia Gschwandtner”,
  - *Pronoun-antecedent coreference*, i.e., pronouns (”he”, ”she”, ”they”, etc.) should be associated with their antecedents, and
  - *Definite description coreference*, for example, a description like ”the machine” could refer to ”the Whirlpool dishwasher” in a text, which has to be correctly associated. Therefore, domain-specific ontological information should be included in IE systems, because a general resolution to this task, however, is unrealistic [1, 20].
5. **Domain specific analysis:** This module is the core of the IE system, for actually acquiring the targeted domain relevant information and filling the templates, usually constructed as attribute-value structures.

”Templates consist of a collection of slots (attributes), each of which may be filled by one or more values. These values can consist of the original text, one or more of a finite set of predefined alternatives, or pointers to other template objects. Typically, slot fills are subjected to normalization rules that standardize the representation of fills representing dates, times, job titles, etc.” [1]

For example, the text

Last Monday, the jeweler's shop Diamonds & Cie was robbed. Among the suspects is Theresia Gschwandtner, a 27 year old student at the Vienna University of Technology.

might be represented in the following template structure in an IE system, constructed for a crime domain [1, 20]:

INCIDENT-0001:  
TYPE: ROBBERY  
DATE: 25-Feb-08  
SUSPECT: <PERSON-0001>  
VICTIM: <INSTITUTION-0001>

PERSON-0001:  
NAME: "Theresia Gschwandtner"  
PROFESSION: STUDENT  
WORKING PLACE: "Vienna University of Technology"  
AGE: 27

INSTITUTION-0001:  
TYPE: JEWELER'S SHOP  
NAME: "Diamonds & Cie"

6. **Merging Partial Results:** Usually the information to fill a single template is spread among different sentences, which makes it necessary to first combine partial results from different templates and then create the final templates. To accomplish that, some IE systems include a merging module to determine which information of which templates can be merged [1, 20].

## 2.3 UMLS<sup>®</sup>- Unified Medical Language System<sup>®</sup>

The Unified Medical Language System [21] is developed by the *National Library of Medicine*, USA, within the UMLS R&D project, initiated in 1986. It is a controlled compendium of many vocabularies and classifications of the biomedical domain and also provides a mapping structure between them. The UMLS was created to facilitate the development of computer systems that process biomedical text by offering access to this knowledge. There are three main UMLS knowledge sources: the Metathesaurus<sup>®</sup>[32], the Semantic Network [23], and the SPECIALIST Lexicon [4], [18, 36].

### 2.3.1 The Metathesaurus<sup>®</sup>

The UMLS Metathesaurus [32] is the largest thesaurus in the biomedical domain, containing medical concepts from more than 100 vocabularies. It is built from numerous thesauri (e.g.,

Medical Subject Headings (MeSH) [28], Computer Retrieval of Information on Scientific Projects (CRISP) of the National Institute of Health (NIH) [27]), classifications (e.g., International Classification of Diseases (ICD-9-CM) [7]), clinical coding systems (e.g., Systematized Nomenclature of Medicine (SNOMED CT) [8]), and lists of controlled terms used in various biomedical documents. Consequently it is not built to be a single standard vocabulary, but enables exchange of information between different clinical databases and systems, in accordance with contextual and inter-contextual relationships between these diverse coding systems and vocabularies.

The Metathesaurus is structured by medical concepts or meanings; all alternative names and views of the same concept from the different vocabularies are linked within a hierarchical context. Furthermore, useful relationships between these concepts are represented [18, 36].

### 2.3.2 The Semantic Network

The Semantic Network [23] specifies the categorisation of the concepts in the Metathesaurus to basic semantic types, such as *antibiotic* or *pathologic function*, just to name two of the 135 semantic types of the Network. All concepts in the Metathesaurus are assigned to at least one semantic type. It also defines the set of useful relationships between these types and concepts (the current release of the Semantic Network contains 54 relationships).

In the network the semantic types represent nodes and the Semantic Relations the links between them. There are major groupings of semantic types, such as *organisms*, *anatomical structures*, *biologic function*, *chemicals*, *events*, *physical objects*, and *concepts or ideas* [18, 33].

### 2.3.3 The SPECIALIST Lexicon and the SPECIALIST NLP Tools

The SPECIALIST Lexicon is an English lexicon that includes many terms of the biomedical domain. It contains syntactic, morphological, and orthographic information about each word or term in the lexicon.

The SPECIALIST Natural Language Processing (NLP) Tools have been developed by the *Lexical Systems Group* of the *Lister Hill National Center for Biomedical Communications*, to facilitate NLP by providing lexical variation and text analysis for application developers using the UMLS. Among others there are lexical tools to manage lexical variations, text tools to analyze plain text documents into words, terms, phrases, sentences and sections, and spelling tools to suggest correct spellings for misspelled words [18, 33].

## 2.4 The MetaMap Program

Based on the SPECIALIST NLP tools NLM has developed the MetaMap program (also referred to as the MMTx program, which stands for "Meta Map Transfer") [2]. As a first step towards implementing a computer based application of a guideline, MetaMap detects noun phrases in biomedical text and assigns them to corresponding concepts in the UMLS Metathesaurus. In doing so it goes through five steps:

1. Parsing arbitrary text into simple noun phrases by using the SPECIALIST minimal commitment parser.

2. Generating variants for each phrase, i.e., all its spelling variants, acronyms, abbreviations, synonyms, inflectional and derivational variants and meaningful combinations of these, using the SPECIALIST lexicon and a supplementary database of synonyms.
3. Retrieving a candidate set of all Metathesaurus concepts containing at least one of the variants.
4. Evaluating each candidate against the input text by computing the mapping strength of the candidate using a linguistically principled evaluation function and then arraying the candidates according to their mapping strength.
5. Combining candidates from disjoint parts of the noun phrase, recomputing the mapping strength for the combined candidates and forming a set of best Metathesaurus mappings for the original phrase.

The output of the MMTx program provides important information for knowledge engineers to better understand the medical text and its underlying concepts, which is a prerequisite for the further steps of transforming the text into a computer executable model [2, 3].

## 2.5 Other Biomedical Semantic Annotation Systems

There exist multiple systems which generate a mapping from free biomedical text to UMLS concepts. Typically the unit of analysis is a phrase. In literature, four types of matches between tokens of text documents and UMLS concepts have been described [29]:

- None: no match could be found.
- Simple: an exact match between the text-token and a UMLS concept exists.
- Partial: one or more tokens match a UMLS concept only partially.
- Complex: the original phrase is divided into sets of terms, whereupon a mapping to UMLS concepts is computed for each of them.

In the following subsection I will give a short summary of existing systems and the kind of matches they support.

1. The **CONANN** system [30] generates a list of candidate phrases (phrases associated with a UMLS concept and having words in common with the input phrase) and then decreases the list with the help of a coverage filter (i.e., common words) and a coherence filter. If there is more than one candidate phrase left, a final concept mapping is performed by computing the UMLS concept to which most of the remaining candidate phrases belong. CONANN uses simple and partial mapping.
2. The **Concept Locator** [26] uses the IBM Intelligent Text Miner's Features Extraction Tool to identify input phrases. The algorithm looks for exact matches of all the words or subsets of words in the phrase to UMLS concepts having the same set of words. If no exact match is found, both the words in the UMLS concept and the text-token are stemmed and again exact matches are computed for the stemmed words. Consequently the Concept Locator supports simple and partial mapping.

3. The **Dynamic Taxonomy** system [37] uses NLP and moving window methods for phrase-identification. It normalizes the input phrase using UMLS tools and then looks for exact matches for the phrase.
4. The **KnowledgeMap** system [9] uses a natural language parser to identify sentences and noun phrases. The KnowledgeMap Concept Identifier (KMCI) identifies concepts in noun phrases by looking for a set of UMLS concepts matching the phrase. If no exact match can be found, variants of terms in the phrase are generated and then UMLS concepts matching these variants are looked for. KnowledgeMap supports simple and partial mapping.
5. The **IndexFinder** system [39] does not tokenize the text into phrases. It looks for all possible UMLS concepts by using all combinations of words in the text, regardless of their location. It supports simple and partial mapping.
6. The **PhraseX** program [34] identifies noun phrases with NLP methods and generates a simple mapping for these phrases.
7. The **SAPHIRE** system [16] takes a phrase or a sentence as input and identifies individual terms within this text. For each term a list of UMLS concepts containing this word is generated, whereupon those lists are merged and UMLS concepts are excluded if they don't match at least one-half of the number of terms within the input text. A scored set of UMLS concepts is retrieved. The system supports simple and partial mapping.
8. The **SENSE** (Search with New Semantics) system [38] extracts so called semantic factors from a user query (equivalent to a phrase). Semantic factors are concepts which cannot be decomposed anymore. The Semantic Analyzer component generates identical semantic factors for all input phrases with the same meaning. The system generates a list of matching MeSH terms (Medical Subject Heading terms from the National Library of Medicine), but it does not compute any scoring of these terms. It supports simple mapping.

A comprehensive comparison of these systems can be found in [29] and [30]. The MMTx program is considered a state-of-the-art system for semantic annotation [9]. It supports simple, partial, and complex mapping. Additionally, it scores candidates by combining four different measures [29]:

- *Centrality*, a metric to indicate if the UMLS concept includes the head term of the source phrase;
- *Variation* defines the distance between the term variant of the source phrase and the term in the UMLS concept;
- *Coverage* measures the overlap between the terms of the source phrase and the terms contained in the UMLS concept;
- *Coherence* identifies term sequence overlaps between the source phrase and the UMLS concept.

## Chapter 3

# Analysis and Realization

### 3.1 Problem Analysis

It is not a trivial task to "translate" a clinical guideline - plain text written by medical experts - into a computer executable language. To ensure the correct contextual functioning of such an implementation, it is necessary for physicians and knowledge engineers to work together to guarantee both a correct interpretation of the medical text and a correct implementation of the corresponding software component.

As a first step towards the realization of a computerized guideline, the MMTx program (see Section 2.4) creates a mapping of the text by tokenizing the text into noun phrases and assigning them to corresponding concepts in the UMLS Metathesaurus.

Due to complexity and ambiguity of free text, it is **not always possible** to achieve a **correct tokenization** of the text as well as an **unambiguous mapping** of UMLS concepts to text chunks. Thus, it is still necessary for physicians to control these results and - should the occasion arise - to modify them.

"Users will need a moderate amount of programming knowledge to use MMTx effectively." [25]

Consequently there is a strong demand for a simple editor to facilitate that task for experts in medical science, who generally are not very familiar with programming.

An example of the original output of the MMTx program is given below.

```
Processing 00000000.tx.1: For patients above five years with mild asthma
inhaled steroids are the most effective preventer drug.
```

```
Phrase: "For patients"
Meta Candidates (1)
  1000 Patients [Patient or Disabled Group]
Meta Mapping (1000)
  1000 Patients [Patient or Disabled Group]
```

```
Phrase: "above five years"
Meta Candidates (4)
```



861 Years (year) [Idea or Concept,Temporal Concept]  
827 year (Precision - year) [Idea or Concept]  
694 Five [Quantitative Concept]  
661 Fives [Daily or Recreational Activity]  
Meta Mapping (888)  
694 Five [Quantitative Concept]  
861 Years (year) [Idea or Concept,Temporal Concept]

Phrase: "with mild asthma inhaled steroids"

Meta Candidates (8)  
812 Steroids [Steroid]  
812 Steroids (Abuse of steroids) [Mental or Behavioral Dysfunction]  
694 Mild asthma [Finding]  
645 Inhaled (Inspiration function) [Organ or Tissue Function]  
645 Asthma [Disease or Syndrome]  
645 Mild (Minimal) [Qualitative Concept]  
645 Mild (Mild Severity of Illness Code) [Intellectual Product]  
645 Mild (Mild Allergy Severity) [Qualitative Concept]

Meta Mapping (861)  
694 Mild asthma [Finding]  
645 Inhaled (Inspiration function) [Organ or Tissue Function]  
812 Steroids (Abuse of steroids) [Mental or Behavioral Dysfunction]

Meta Mapping (861)  
694 Mild asthma [Finding]  
645 Inhaled (Inspiration function) [Organ or Tissue Function]  
812 Steroids [Steroid]

Phrase: "are"

Meta Candidates (0): <none>  
Meta Mappings: <none>

Phrase: "the most effective preventer drug."

Meta Candidates (13)  
812 Drug (Pharmaceutical Preparations) [Pharmacologic Substance]  
756 Pharmaceutical [Intellectual Product]  
719 Pharmaceutics (Pharmacy) [Biomedical Occupation or Discipline]  
719 Medicament [Pharmacologic Substance]  
645 Most [Quantitative Concept]  
645 Effective (Effect) [Qualitative Concept]  
574 PREVENT [Pharmacologic Substance]  
574 Effectiveness [Qualitative Concept]  
574 Prevents [Functional Concept]  
545 Prevention (Prophylactic treatment) [Therapeutic or Preventive Procedure]  
545 Prevention [Therapeutic or Preventive Procedure]  
545 Preventable [Intellectual Product]

530 Preventive (Preventive intent) [Qualitative Concept]  
 Meta Mapping (815)  
 645 Most [Quantitative Concept]  
 645 Effective (Effect) [Qualitative Concept]  
 574 Prevents [Functional Concept]  
 812 Drug (Pharmaceutical Preparations) [Pharmacologic Substance]  
 Meta Mapping (815)  
 645 Most [Quantitative Concept]  
 645 Effective (Effect) [Qualitative Concept]  
 574 PREVENT [Pharmacologic Substance]  
 812 Drug (Pharmaceutical Preparations) [Pharmacologic Substance]

It is evident that it is not easy to assess and occasionally correct the information contained in this text list. A GUI would greatly facilitate this task displaying the textual guideline together with all the assigned conceptual information in a straightforward way, thus enabling knowledge engineers to understand and interpret the medical text correctly, thus ensuring the quality of information extracted from the textual guideline.

## 3.2 Aim of MapFace

Two significant aspects in order to take greatest possible advantage from the information gained from semantic annotation of a clinical guideline are:

- **guaranteeing the quality** of the provided information, and
- providing an easy **access to all available information** in combination with **visualizing the comprehensive information**.

With respect to these two aspects, the MapFace editor was designed to serve a twofold purpose, i.e., to provide possibilities to control and correct the information generated by means of the MMTx program and to visualize it in a straightforward way. Thus, on the one hand Mapface enables physicians to guarantee the reliability of the resulting information, and on the other hand it supports the correct interpretation of the text. The quality of further processing steps, too, is thereby potentially enhanced.

In order to provide this functionality the implementation of the MapFace editor must meet the following requirements:

1. Possibilities to correct the tokenization of phrase chunks
2. Possibilities to correct the tokenization of concept chunks
3. Possibilities to control, complete and correct the affiliation of UMLS concepts to concept chunks
4. Possibilities to assign semantic types to phrase chunks

5. Access to all information available for a medical concept in the text of the guideline, i.e., its assigned UMLS concept, its semantic type, all existing relations to other concepts in the text, etc.
6. Access to all information available for a phrase of the guideline, i.e., its associated semantic type, relations to semantic types of other phrases in the guideline, etc.
7. Support of the visualization of information by color-coding all semantic types in order to highlight different constructs of the text or to visualize various relations

To serve the different needs of two user groups, i.e., physicians and knowledge engineers, MapFace has to display the different aspects of information, depending on the user and the purpose of his/her work.

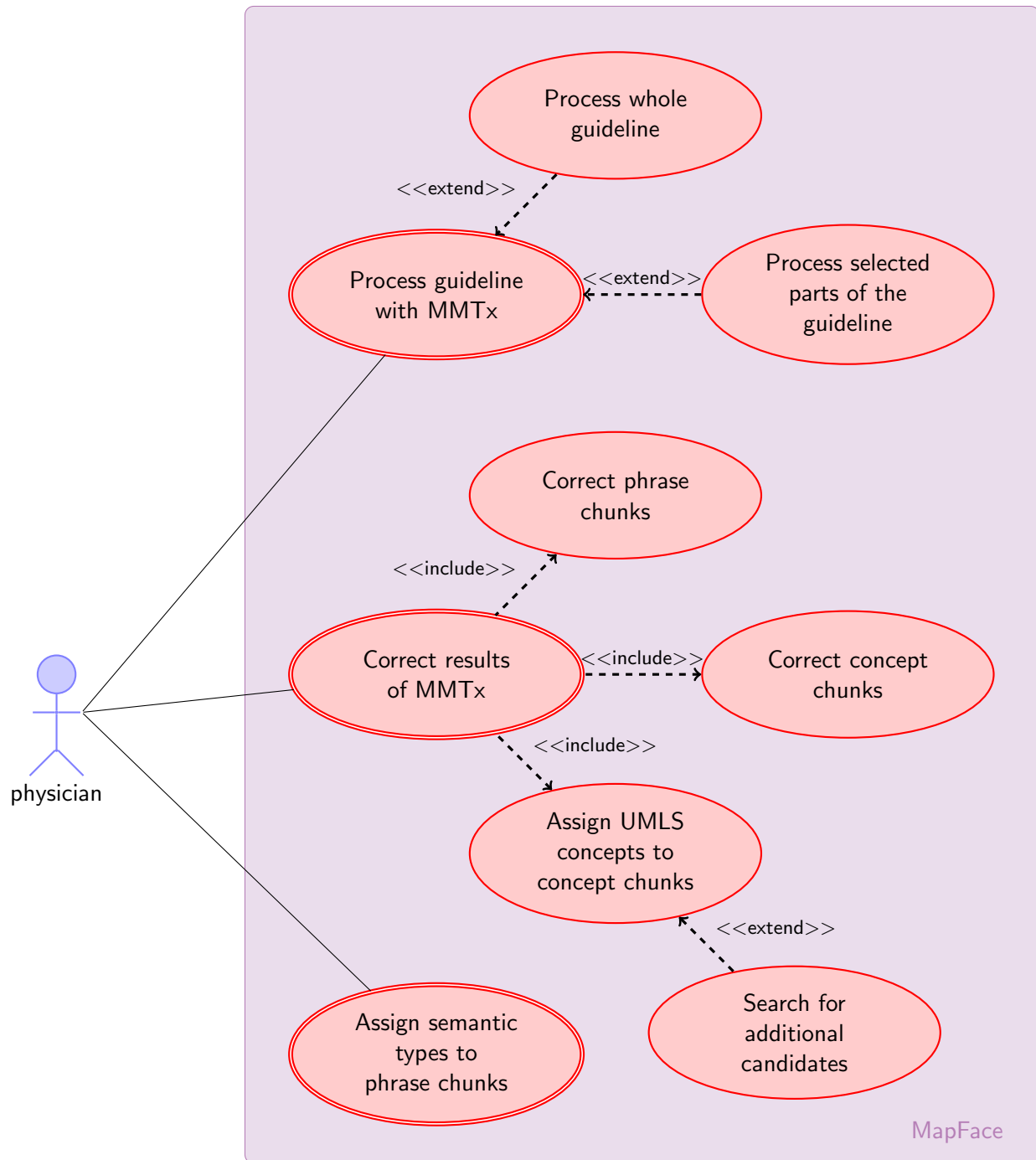


Figure 3.1: A physician's options to work with MapFace.

A comprehensive description of use cases can be found in appendix A; here, however, a short overview seems sufficient.

### 3.2.1 Processing the text of a guideline with MMTx

**User:** Physician

**Description:** The user can process a whole guideline as well as selected parts of the text by means of the MMTx program (see Figures 3.1 and 3.2). The guideline text is split into sections, sentences, and phrases. Furthermore, a mapping of UMLS concepts for each medical concept within a phrase is computed. The MapFace editor re-links these results to the corresponding tokens of the guideline text in order to provide all associated information when selecting such a text chunk in the editor. If a concept chunk is selected, a list of best matching UMLS concepts is displayed together with additional information. If a phrase chunk is selected, MapFace displays a list of all UMLS concept candidates assigned to concepts within the phrase.

### 3.2.2 Correcting phrase chunks

**User:** Physician

**Description:** If a phrase chunk appears to be wrongly tokenized, MapFace provides possibilities to correct the boundaries of this token. Depending on the situation the user can choose between diverse possibilities for correcting a phrase chunk (see Figure 3.3):

1. Simply deleting the phrase chunk
2. Splitting the phrase chunk into two chunks, by deleting it and creating two new chunks from its contents
3. Merging two adjacent phrase chunks into a single chunk
4. A combination of these methods

### 3.2.3 Correcting concept chunks

**User:** Physician

**Description:** If concept chunks appear to be wrongly tokenized, MapFace enables the user to correct them by deleting one or more chunks and manually creating new concept chunks from their contents (see Figure 3.4).

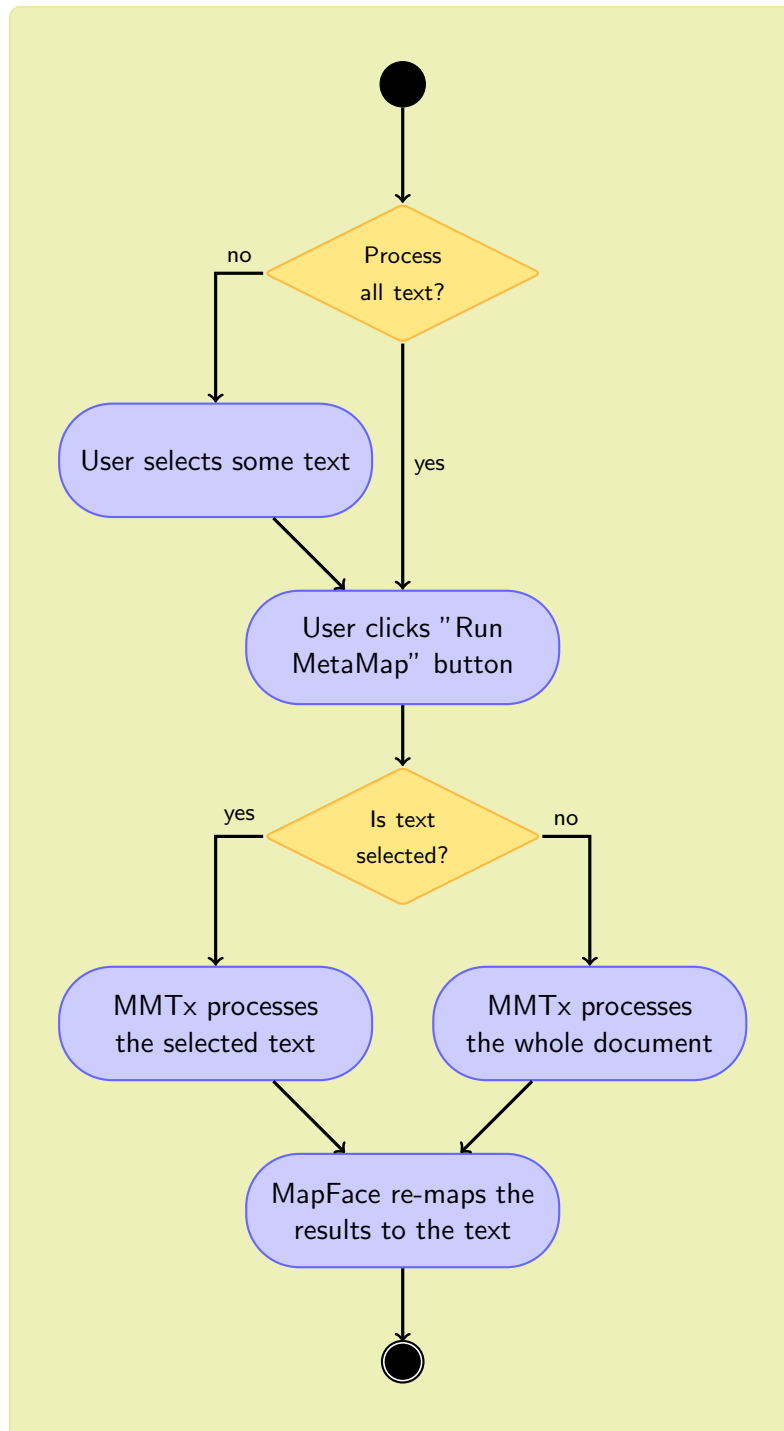


Figure 3.2: The workflow of creating MMTx result with the MapFace editor.

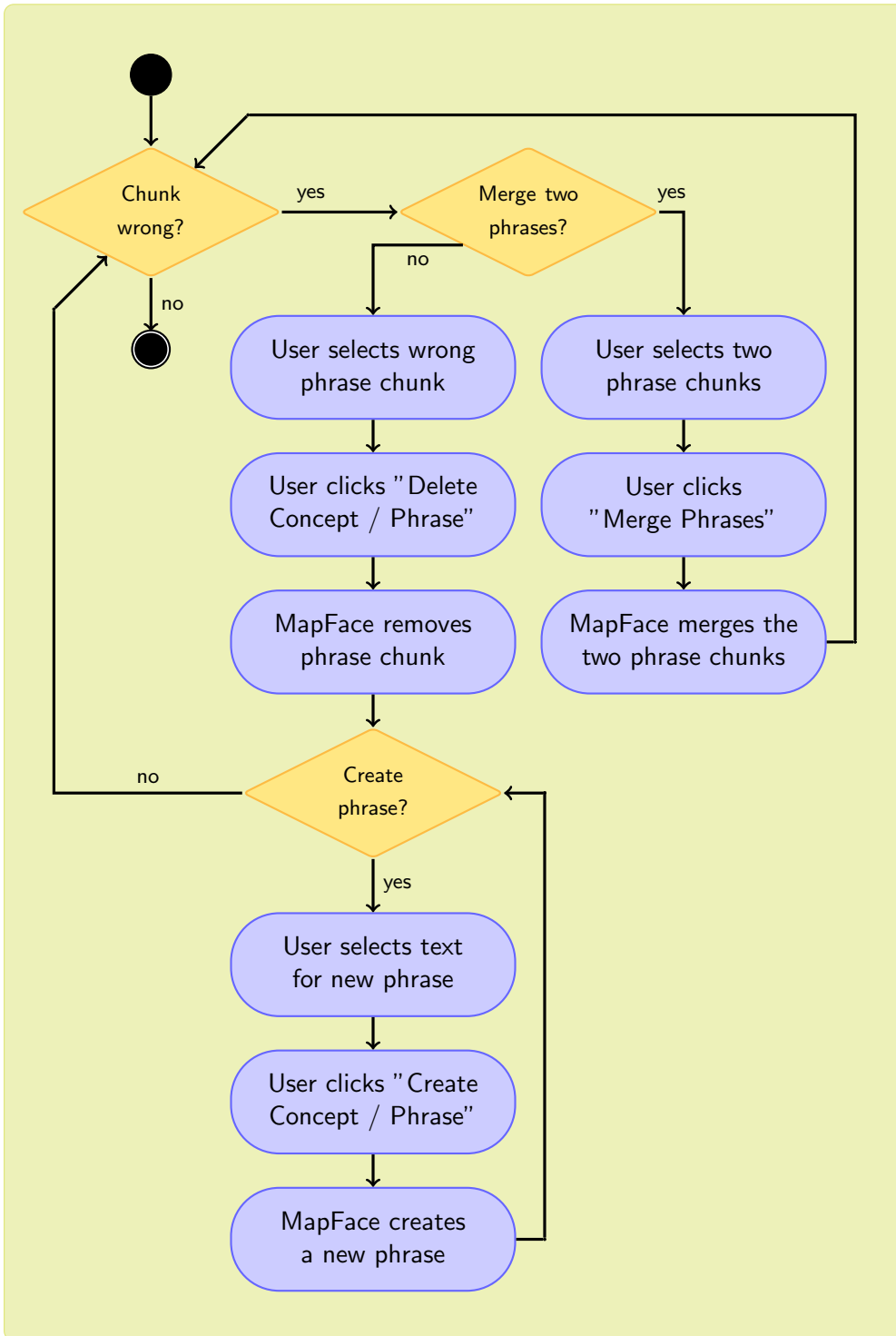


Figure 3.3: The workflow of correcting phrase chunks.

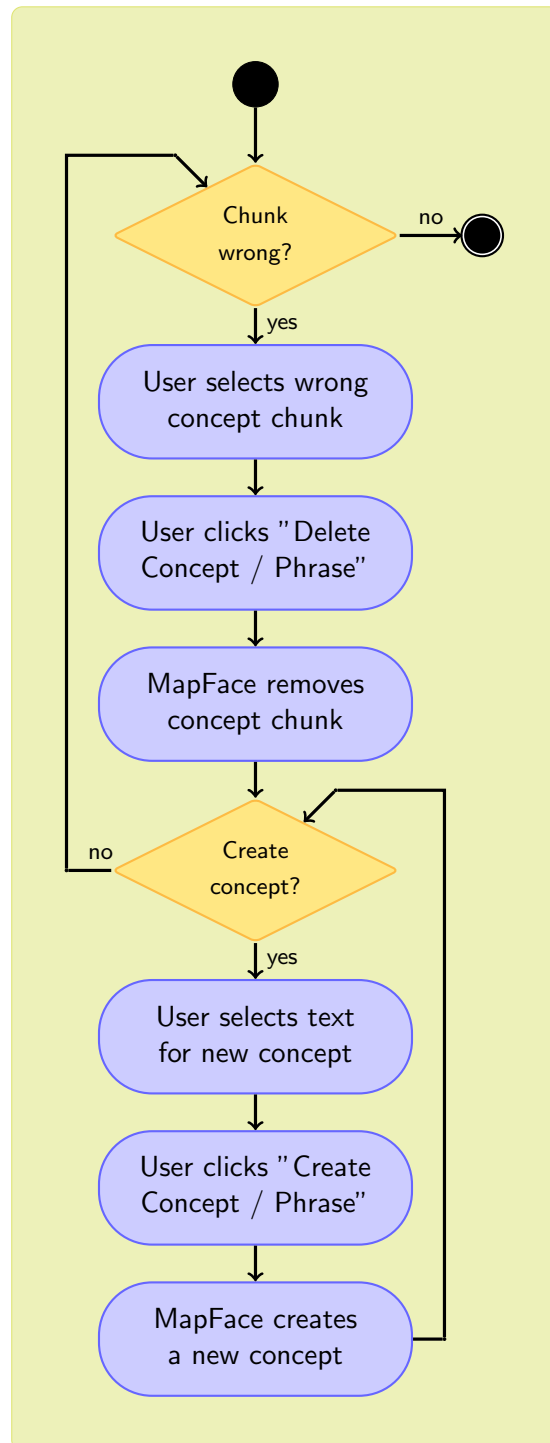


Figure 3.4: The workflow of correcting concept chunks.



### 3.2.4 Assigning UMLS concepts to concept chunks

**User:** Physician

**Description:** When the user selects a concept chunk in the editor, MapFace displays a list of best matching UMLS concepts provided by the MMTx program. In unambiguous cases the UMLS concept is assigned automatically. If there is more than one UMLS concept candidate for a selected concept chunk, the user can choose the appropriate candidate from the list and manually assign it to the text chunk of the concept.

If the correct UMLS concept is not available in the list, MapFace provides the possibility to look for additional UMLS concepts matching an alternative text entered by the user. The UMLS concepts found are added to the candidates list of the concept chunk.

The user can remove inappropriate candidates from the UMLS concept candidates list associated with a concept chunk (see Figure 3.5).

### 3.2.5 Assigning semantic types to phrase chunks

**User:** Physician

**Description:** In order to correctly define the semantic meaning of a phrase, the user can select the semantic type associated with one of the concepts within this phrase and assign it to the phrase chunk.

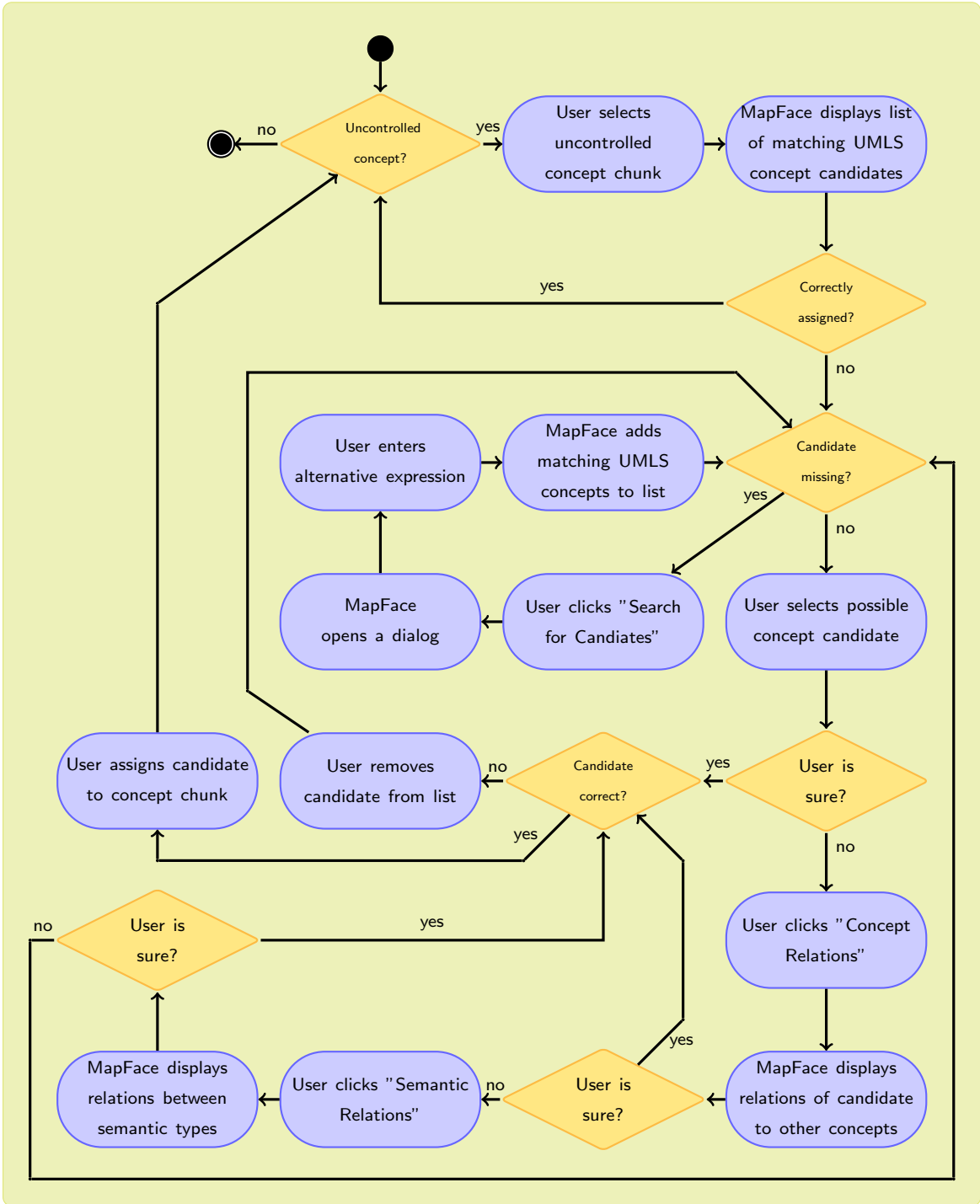


Figure 3.5: The workflow of assigning UMLS concept candidates to concept chunks.

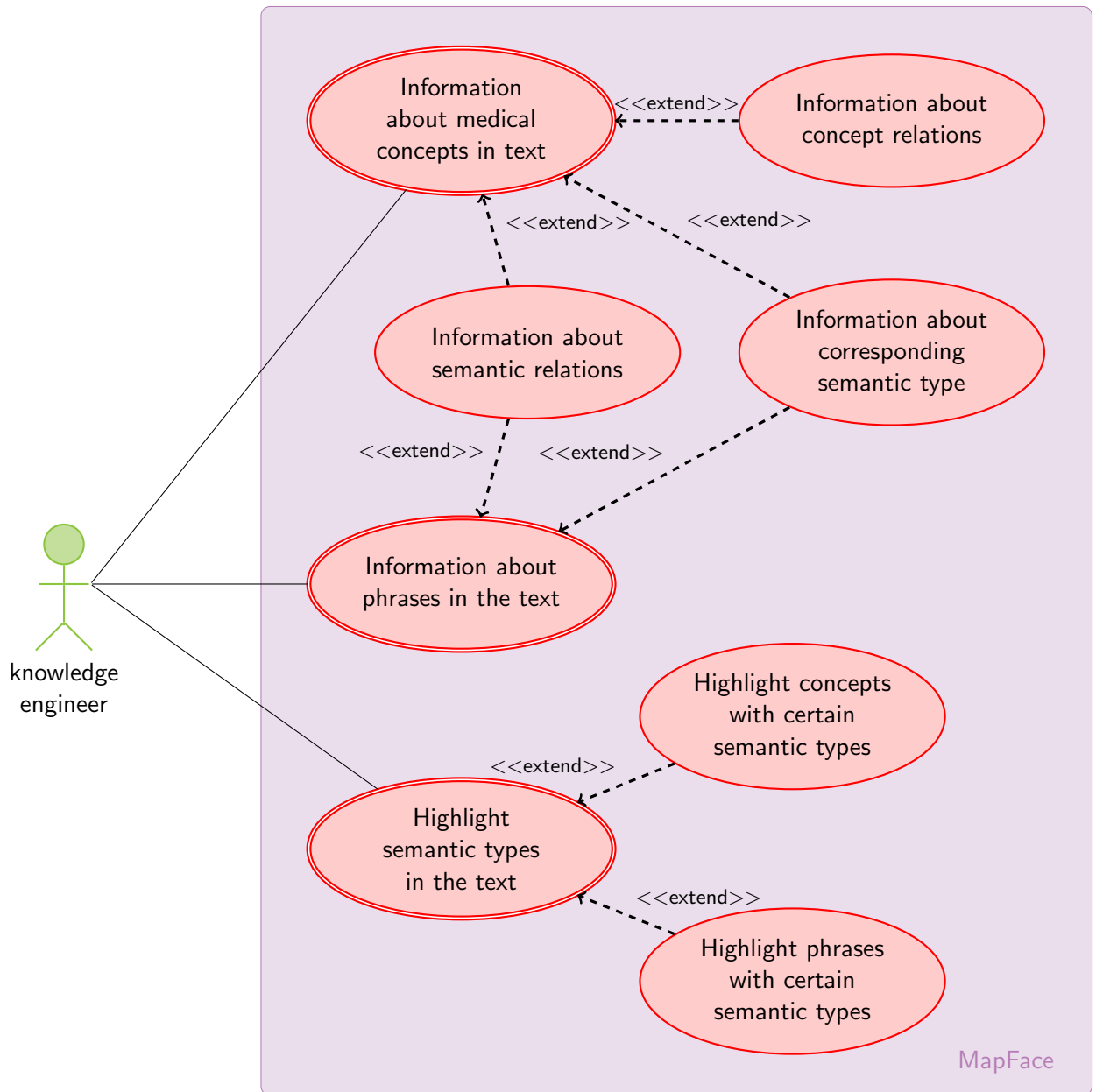


Figure 3.6: A knowledge engineer's options to work with MapFace.

### 3.2.6 Providing Information about medical concepts in the text

**User:** Knowledge engineer

**Description:** When a concept chunk is selected in the editor, MapFace displays the associated UMLS concept together with its semantic type, semantic collection, and semantic group. Furthermore it is possible to display the relations between the selected concept and other concepts as well as between the semantic type of the associated UMLS concept and the semantic types of other concepts occurring in the same section of the text, in combination with highlighting them.

### 3.2.7 Providing Information about phrases in text

**User:** Knowledge engineer

**Description:** When a phrase chunk is selected in the editor, MapFace displays the affiliated UMLS concept together with its semantic type, semantic collection, and semantic group. Furthermore it is possible to display the relations between the semantic type of the associated UMLS concept and the semantic types of other phrases occurring in the same section of the text, in combination with highlighting them.

### 3.2.8 Highlighting certain semantic types

**User:** Knowledge engineer

**Description:** For better visualization all semantic collections are color-coded. The user can select certain semantic types of interest from a list and highlight all associated chunks of the guideline text in the editor.

### 3.2.9 Displaying concept relations

**User:** Both physician and knowledge engineer

**Description:** MapFace provides a list of relations between the UMLS concept of a selected concept chunk and the UMLS concepts assigned to concept chunks in the same section of the text. For better visualization it is possible to highlight the concept chunks concerned in the editor.

### 3.2.10 Displaying semantic relations

**User:** Both physician and knowledge engineer

**Description:** MapFace provides a list of relations between the semantic type of a selected concept chunk or phrase chunk and the semantic types assigned to concept chunks or phrase chunks in the same section of the text. For the sake of better visualization it is possible to highlight the concerned text chunks in the editor.

## 3.3 Realization

### 3.3.1 Resources used by MapFace

Besides relying on the MMTx program and UMLS Metathesaurus, MapFace profits by the iUMLS program, developed by Katharina Kaiser, at the *Institute of Software Technology and Interactive Systems, Vienna University of Technology*. The iUMLS is an interface which provides a convenient access to the MMTx program and its results, as well as to information obtained from the UMLS database, such as relations between concepts and between semantic types.

In addition, MapFace takes advantage of the IR (Information Retrieval) program, developed by Karl-Michael Edlinger, Alex Hörmandinger, Matteo Savio, and Johannes Strodl, at the *Vienna University of Technology*. IR evaluates the most likely semantic types for a phrase from a given list of possible types, by testing each semantic type for relations to semantic types assigned to other phrases in the same sentence.

### 3.3.2 User Interface

If you open MapFace, a window with a **menubar**, a **toolbar**, and three different panes appears: the **editor pane**, the **candidates pane**, and the **annotation scheme pane**.

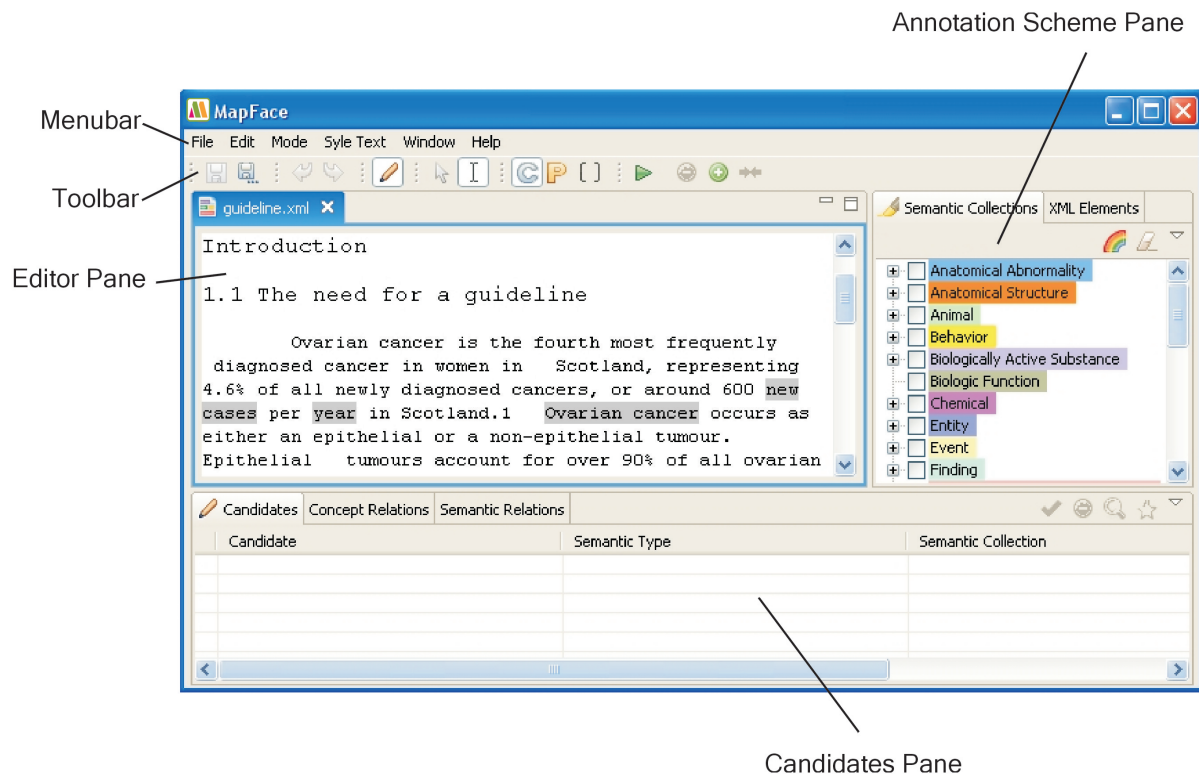


Figure 3.7: Components of the GUI.

### 3.3.2.1 The Menu Bar

All actions provided by the main toolbar of the window and in the local toolbars of the *annotation scheme pane* can be accessed through the menubar as well.

The menubar includes the following menus:

1. "File":
  - "Save" / "Save As.." (see Section 3.3.4.4),
  - "Exit", and
  - "Open File..".
2. "Edit":
  - "Undo" / "Redo" (see Section 3.3.4.4),
  - "Run MetaMap" (see Section 3.3.4.1),
  - "Delete Concept / Phrase" (see Section 3.3.4.1),
  - "Create Concept / Phrase" (see Section 3.3.4.1),
  - "Merge Phrases" (see Section 3.3.4.1),
  - "Arrow Cursor" (see Section 3.3.4.4), and
  - "Text Selection" (see Section 3.3.4.4).
3. "Mode":
  - "Editing Mode" (see Section 3.3.3.1),
  - "Concepts Mode" (see Section 3.3.3.3), and
  - "Phrases Mode" (see Section 3.3.4.1).
4. "Style Text":
  - "Highlight Text" (submenu containing "Highlight All" and "No Highlighting" for semantic types and XML elements (see Section 3.3.4.3)), and
  - "Mark Phrases" (see Section 3.3.4.4).
5. "Window":
  - "Reset Perspective" (see Section 3.3.4.4).
6. "Help":
  - "FAQ" (see Section 3.3.4.4), and
  - "User Manual".

### 3.3.2.2 The Tool Bar

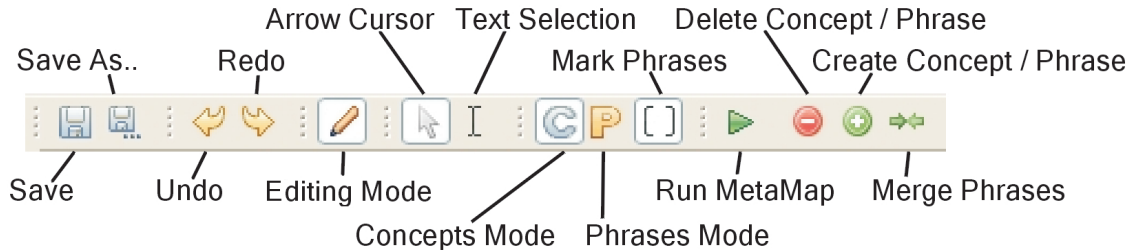


Figure 3.8: The toolbar.

### 3.3.2.3 The Editor Pane

The main window of the Graphical User Interface of MapFace is the **editor pane**. It displays the **text of the clinical guideline**. This is where you can select the text you want to process by means of the MMTx program.

In addition, you can select the concepts or phrases of processed text by double-clicking the text chunk of the concept or phrase in the editor, whereupon a list of best matching UMLS concept candidates will be displayed in the *candidates view* (see Section 3.3.2.4).

### 3.3.2.4 The Candidates Pane

The **candidates pane** is at the bottom of the user interface. It contains three different views, the *candidates view*, the *concept relations view*, and the *semantic relations view*.

#### The Candidates View

The main view of the *candidates pane* is the **candidates view**. It provides information about the **best matching UMLS concept candidates** for a given text chunk and possibilities for editing. Therefore the *candidates view* displays a list of candidates detected by the MMTx program for a selected concept chunk or phrase chunk in the editor, together with their semantic types, semantic collections [5], and semantic groups [24]. The lines of the table are highlighted according to the color coding of the corresponding semantic collections (see Section 3.3.2.5).

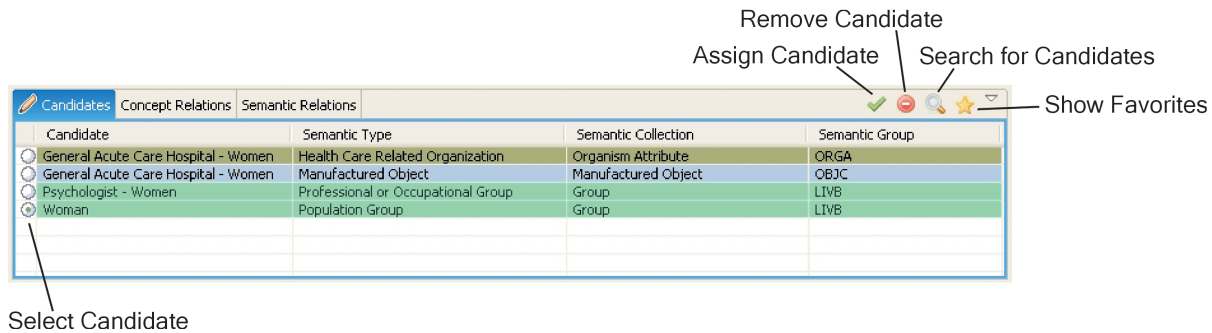


Figure 3.9: Candidates view.

Now you can select the right candidate from the list and assign it to the concept chunk or phrase chunk by clicking the "Assign Candidate" button. If the appropriate candidate does not appear in the list, you can look for additional candidates by clicking the "Search for Candidates" button and then enter an alternative expression for the text of the concept chunk. Furthermore you can remove candidates from the list by clicking the "Remove Candidate" button. If the *phrases mode* is active, a shorter list containing the most likely candidates for the selected phrase chunk is displayed when clicking the "Show Favorites" button.

### The Concept Relations View

The **concept relations view** provides additional information for a UMLS concept candidate selected in the *candidates view*. It displays a list of all relations between the concept candidate selected in the *candidates view* and the UMLS concepts affiliated to concept chunks in the same section of the text. By selecting a relation from the list, the two concept chunks concerned are highlighted in the editor. Relations between concepts are only available when the *concepts mode* (see Section 3.3.3.3) is active.

### The Semantic Relations View

The **semantic relations view** provides a list of all relations between the semantic type of the UMLS concept candidate selected in the *candidates view* and the semantic types occurring in the same section of the text. The two concerned concept chunks or phrase chunks are highlighted in the editor by selecting a relation from the list.

#### 3.3.2.5 The Annotation Scheme Pane

The **annotation scheme pane** is at the right of the user interface. The *annotation scheme pane* of the basic MapFace program contains two different views, the *semantic collections view*, and the *XML elements view*. In both views you can select all elements in the view and accordingly highlight all associated elements in the editor by clicking the "Highlight All" button (see Section 3.3.4.3), as well as you can deselect all elements by clicking the "No Highlighting" button (see Section 3.3.4.3).



## The Semantic Collections View

The **semantic collections view** contains a **list of all semantic types**, grouped by semantic collections [5], each associated with a different color. Here you can select semantic types of interest, which will **highlight** every text chunk in the editor associated with these semantic types.

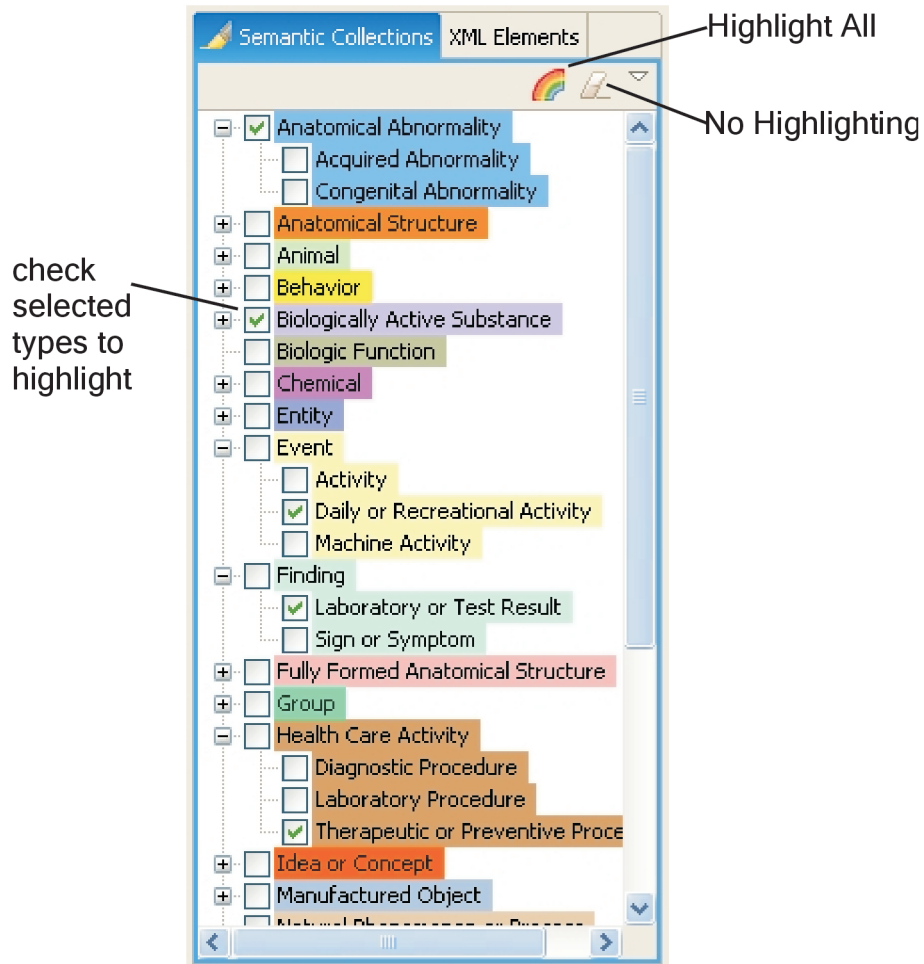


Figure 3.10: The *semantic collections view*.

## The XML Elements View

The **XML elements view** contains all names of **XML tags** occurring in the XML document that is being processed. By selecting an XML element in the view, you **highlight** the corresponding text chunk of the guideline in the editor.

### 3.3.3 Modes

The figure below provides an overview of the different modes of MapFace.

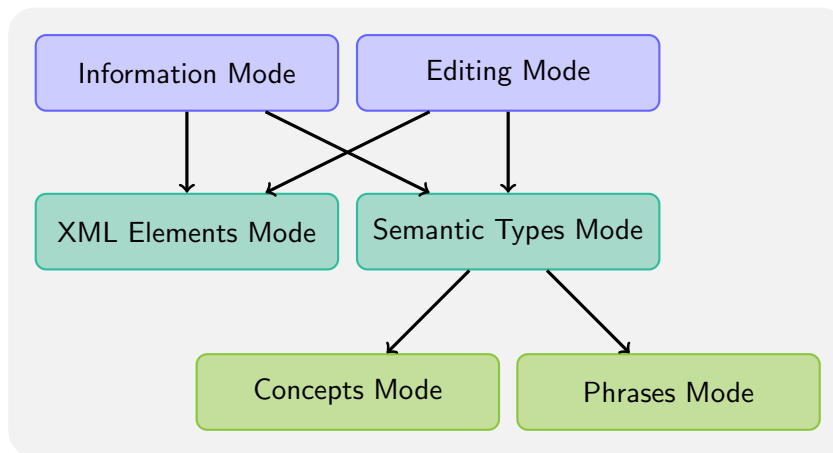


Figure 3.11: Modes.

#### 3.3.3.1 Information Mode vs. Editing Mode

The **information mode** and the **editing mode** exclude each other, so only either the *editing mode* or the *information mode* can be active at one time. Whenever the "Editing Mode" button in the toolbar is checked, the *editing mode* is active; otherwise the *information mode* is active.

#### The Information Mode

The **information mode** is to serve the needs of knowledge engineers working with MapFace. Therefore, an efficient representation of the information is provided by the following features:

- The **clinical guideline text** is displayed in the editor,
- The **assigned information** for a selected concept or phrase in the text is displayed in the *candidates view* (see Section 3.3.2.4),
- Concepts and phrases can be **highlighted** according to their **semantic types** (see Section 3.3.4.3),
- All phrases existing in the document can be delimited by brackets (see Section 3.3.4.4),
- **XML elements** defined by **XML tags** in the current XML file can be **highlighted** (see Section 3.3.4.3), and
- **Relations** between concepts and between semantic types can be **displayed** in the *candidates pane*, in combination with **highlighting** them in the text.

## The Editing Mode

The **editing mode** is to serve the needs of physicians when working with MapFace. In addition to the features of the *information mode*, it offers possibilities to edit the generated phrases and concepts.

1. The following functions are available when working in the **concepts mode** (see Section 3.3.3.3):
  - **Running** the MetaMap program,
  - **Deleting** an existing concept,
  - **Creating** a new concept from selected text,
  - **Choosing** the right concept from a list of equally matched metathesaurus concepts,
  - **Searching** for other metathesaurus concepts,
  - **Assigning** candidates to concepts, and
  - **Removing** concept candidates from the list.
2. The following functions are available when working in the **phrases mode** (see Section 3.3.3.3):
  - **Running** the MetaMap program,
  - **Deleting** an existing phrase,
  - **Creating** a new phrase from selected text,
  - **Merging** two adjacent phrases,
  - **Choosing** the correct match from a list of semantic types for the phrase,
  - **Assigning** semantic types to the phrases of the guideline, and
  - **Reducing** the list of candidates by using the "Show Favorites" action.

### 3.3.3.2 Semantic Types Mode vs. XML Elements Mode

The **semantic types mode** and the **XML elements mode** exclude each other, so only either one of them can be active at one time.

#### The Semantic Types Mode

The **semantic types mode** is active every time the *semantic collections view* is visible in the *annotation scheme pane*. In this mode, you can make a choice between the *concepts mode* and the *phrases mode* (see Section 3.3.3.3).

#### The XML Elements Mode

The **XML elements mode** is active whenever the *XML elements view* (see Section 3.3.2.5) is visible in the *annotation scheme pane*. The *XML elements view* displays a tree structure, representing all XML elements occurring in the open XML document (except the elements inserted by MapFace). In this mode you can select different XML elements and accordingly highlight the text in the editor.

### 3.3.3.3 Concepts Mode vs. Phrases Mode

The **concepts mode** and **phrases mode** are only available if the *semantic types mode* (see Section 3.3.3.2) is active. The two modes exclude each other, so only either the *concepts mode* or the *phrases mode* can be active at one time.

These modes are accessible in the toolbar of the MapFace window, and as subitems in the *Mode* menu.

#### Concepts Mode

When the **concepts mode** is active, all information, editing and highlighting actions refer to medical concepts assigned to the text by MetaMap. You can select a concept by double-clicking it in the editor. Then you may read the assigned information, or process it (see Section 3.3.3.1).

#### Phrases Mode

A phrase is a meaningfully arranged combination of words within a sentence, which may contain medical concepts. When the **phrases mode** is active, all information, editing and highlighting actions refer to phrase chunks detected in the text by MetaMap.

### 3.3.4 Features

#### 3.3.4.1 Editor Actions

You can find the editor actions in the main toolbar of the MapFace window.

#### Run MetaMap

”Run MetaMap” is **the first step** to process a new XML file. If you process text of your document with MetaMap, the text will be tokenized into sections, sentences, phrases, and concepts, and a great amount of **syntactic and semantic information** will be computed for these elements. MetaMap primarily deals with the semantic information, i.e., the detected UMLS-Metathesaurus concepts. However, all the information will be saved in the “.idoc” file for later processing.

You can either **process the whole document at once**, by simply clicking the ”Run MetaMap” button, or you can select text in the editor first (with the text selection cursor; see Section 3.3.4.4), and then make MetaMap **process only the selected text**. This action will take some time, depending on the amount of text to be processed. If you do not want to wait so long, select smaller parts of the text.

You can undo and redo this action by using ”Undo” / ”Redo” (see Section 3.3.4.4) in the toolbar of the MapFace window. After running MetaMap, you can edit the detected concepts or phrases by selecting the corresponding text chunks in the editor (using the *arrow cursor*; see Section 3.3.4.4).

#### Delete Concept / Phrase

This action is thought to correct the tokenization of concept chunks and phrase chunks detected by MetaMap. To **delete an existing concept chunk or phrase chunk**, you select it in the

editor and simply click the "Delete Concept / Phrase" button. If you delete a phrase this way, all included concepts will be deleted as well.

You can undo and redo this action by using "Undo" / "Redo" (see Section 3.3.4.4) in the toolbar of the MapFace window.

### Create Concept / Phrase

This action is to correct the tokenization, too.

To **create a new concept chunk from a selected text**, the selected text must not contain any concept chunks. If this is the case, delete the concept chunks first (see Section 3.3.4.1). Then select the appropriate text chunk with the *text selection cursor* (see Section 3.3.4.4) and click the "Create Concept / Phrase" button.

To **create a new phrase chunk**, proceed the same way in the *phrases mode*. If you create a new phrase this way, included concept chunks will be created automatically.

You can undo and redo this action by using "Undo" / "Redo" (see Section 3.3.4.4) in the toolbar of the MapFace window.

### Merge Phrases

To **merge two adjacent phrase chunks into one phrase chunk**, you select the two phrases with the *text selection cursor* (see Section 3.3.4.4) and then click the "Merge Phrases" button in the toolbar. You can undo and redo this action by using "Undo" / "Redo" (see Section 3.3.4.4) in the toolbar of the MapFace window.

#### 3.3.4.2 Candidates View Actions

##### Assign Candidate

To **assign a UMLS concept to a concept chunk** or a **semantic type to a phrase chunk**, you need to select the text chunk in the editor. A list of best matching candidates will appear in the *candidates view*, where you can choose the correct candidate by selecting the radio button to its left, and then assign it to the concept chunk or phrase chunk by clicking the "Assign Candidate" button.

You can undo and redo this action by using "Undo" / "Redo" (see Section 3.3.4.4) in the toolbar of the MapFace window.

##### Remove Candidate

If you are sure that a UMLS concept candidate does not match the selected concept chunk, and you want to keep the candidates list clean, you can **remove the candidate from the list** by selecting the radio button to its left and clicking the "Remove Candidate" button.

This function is only available in the *concepts mode*. The list of UMLS concept candidates for a phrase consists of the candidates for the concept chunks included in the phrase.

You can undo and redo this action by using "Undo" / "Redo" (see Section 3.3.4.4) in the toolbar of the MapFace window.

## Search for Candidates

If the appropriate candidate does not appear in the candidates list, you can **search for additional UMLS concepts** for the selected concept chunk by clicking the "Search for Candidates" button and then entering an alternative expression for the concept text. This action is only available in the *concepts mode*.

You can undo and redo this action by using "Undo" / "Redo" (see Section 3.3.4.4) in the toolbar of the MapFace window.

## Show Favorites

"Show Favorites" is available in the toolbar of the *candidates view*, when working in the *editing mode* combined with the *phrases mode*. It is to facilitate choosing the correct semantic type for a phrase, by decreasing the number of semantic types in the list automatically, which is accomplished by taking advantage of the information about the semantic relations of each type. **Semantic types with no relation to other semantic types occurring in the same sentence will be removed from the list.** This may take a few minutes, since the favorite semantic types will be computed for all phrases in the same sentence at once. This action does not modify the original semantic type lists of the phrases, so if you select the phrase again, the original list will be displayed again.

### 3.3.4.3 Highlighting

For both views of the *annotation scheme pane* - the *semantic collections view* and the *XML elements view* - there are options for selecting or deselecting all elements in the view by clicking a button, which effects the **highlighting of the corresponding text chunks in the editor.**

#### Highlight All

If you click the "Highlight All" button in the *semantic collections view*, or the *XML elements view*, **all elements** in the view will be selected and the text chunks in the editor will be **highlighted** accordingly.

In the *semantic collections view*, the "Highlight All" button stays checked, if you click it once, and will be unchecked, if you click it again. A checked "Highlight All" button has the effect that all text chunks in the editor with affiliated semantic type will be highlighted all the time. However, clicking the "Highlight All" button twice will once highlight all the text chunks in the editor associated with a semantic type, but assigning a semantic type to a text chunk afterwards will not affect the highlighting of this chunk.

#### No Highlighting

If you click the "No Highlighting" button in the *semantic collections view*, or the *XML elements view*, all elements in the view will be deselected and the **highlighting** in the editor will be **cleared**.

In the *semantic types mode* (the *semantic collections view* is visible), concept chunks or phrase chunks with no assigned UMLS concept are marked by gray background highlighting,

even if you clear the rest of the highlighting by clicking the button. The gray background serves as reminder that you should manually assign a UMLS concept to this chunk.

#### **3.3.4.4 Additional Features**

##### **Mark Phrases**

This function is available in the main toolbar of the MapFace window and in the *Style Text* menu. It delimitates all phrase chunks in the processed text by surrounding them with brackets, which adds quite a bit of clarity as to which concept chunks belong to which phrase chunk.

##### **Arrow Cursor**

Choosing the **arrow cursor**, you can select concepts or phrases in the editor by double-clicking the corresponding text chunk of the guideline. Depending on the mode (*concepts mode* or *phrases mode*), a list of UMLS candidates for a selected concept chunk, or all UMLS candidates of concepts included in a selected phrase chunk, will be listed in the *candidates view*.

##### **Text Selection Cursor**

Choosing the **text selection cursor** in the toolbar of the MapFace window or in the *Edit* menu of the menubar, you can select text in the editor for various purposes, e.g., for processing it by the means of the MMTx program, for creating a new concept chunk or phrase chunk from selected text, or for selecting two adjacent phrase chunks to merge them.

##### **Undo and Redo**

You can **undo and redo** each executed action by clicking the "Undo" or the "Redo" button in the toolbar or in the *Edit* menu of the menubar.

##### **Save and Save As..**

Saving the processed XML document inserts XML tags for sections ("mf\_section"), sentences ("mf\_sentence"), phrases ("mf\_phrases") and concepts ("mf\_concepts"), as well as for the text between these constructs ("mf\_text"). For concept chunks and phrase chunks with affiliated UMLS concepts, these XML tags contain the attributes "cui" (concept unique identifier) and "sem\_type" (semantic type). Thus, the arrangement of these constructs in the document text is saved, together with some information about assigned UMLS concepts and semantic types. Since the output of the MMTx program provides much more syntactic and semantic information for processed text, a second file is created containing all computed information about these constructs. This file is saved to the same directory and has the same name as the XML document but bears the extension "idoc". When you open the XML document with MapFace again, the information of both files will be merged.

##### **Reset Perspective**

This function resets the MapFace GUI to the initial perspective, i.e., all initially visible views and their arrangement.

### 3.3.5 Example

This is an example of how you, as a physician, would use MapFace to process a clinical guideline.

1. Open an XML document: To **open a file**, select the *File* menu in the menubar and choose "Open File.". Navigate through your disk system and choose an XML document of a clinical guideline to open.
2. Get an impression of the **elements of the XML document**: If you now activate the *XML elements view* in the *annotation scheme pane* at the right of the window, a treestructure of XML elements occurring in your document will be displayed. By selecting an element from the tree, you highlight the corresponding text in the editor.
3. Change to **editing mode**: Checking the "Editing Mode" button in the toolbar will enable you to modify the guideline document.
4. Process selected text by means of the **MMTx** program: Activate the *semantic collections view* to change to *semantic types mode* again. Now you can select some text of the guideline in the editor and click the "Run MetaMap" button in the toolbar. The MMTx program will tokenize the selected text into sections, sentences, phrases, and concepts, and compute syntactic information as well as a list of matching UMLS concepts for each concept in the text.



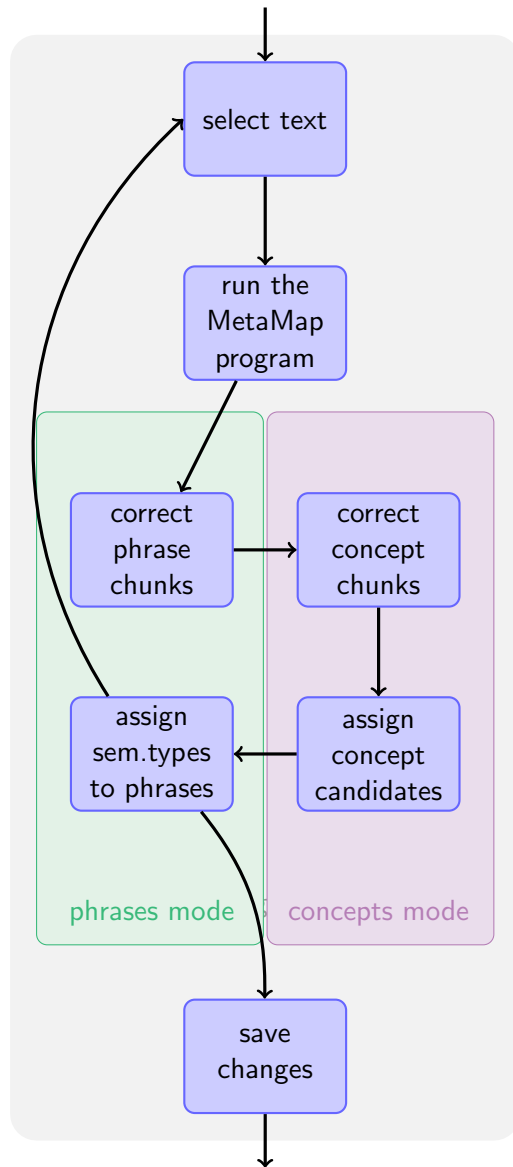


Figure 3.12: Example of how to process a clinical guideline.

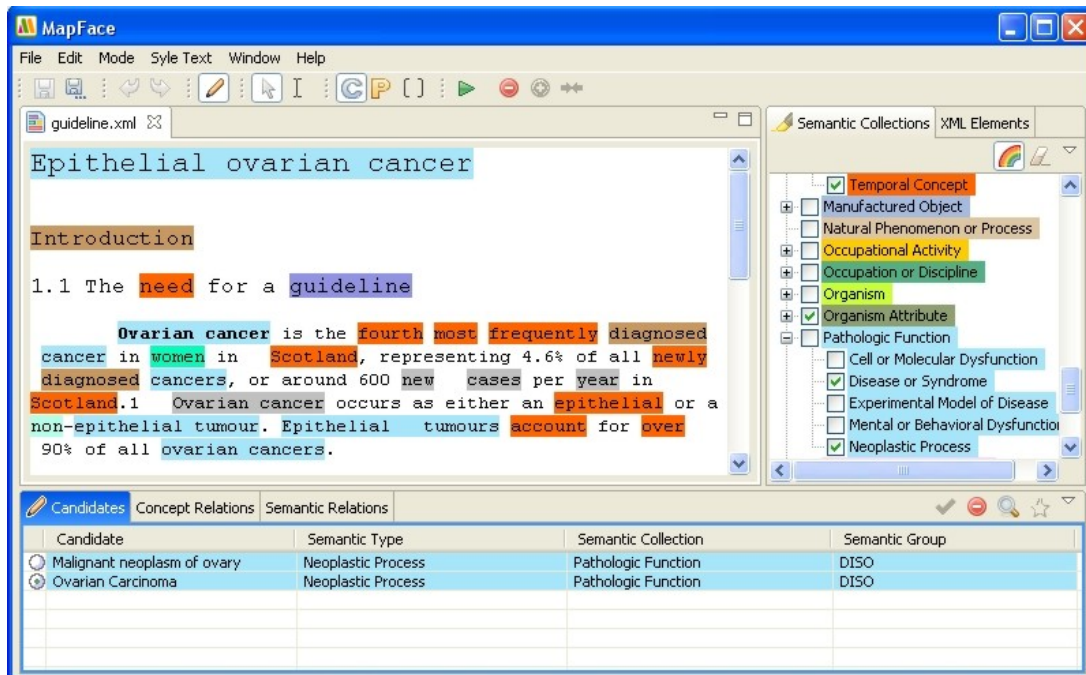


Figure 3.13: MapFace after running the MMTx program and highlighting the concept chunks.

5. **Highlight** the assigned semantic types: To make the result of the MMTx program visible, you can click the "Highlight All" button in the *semantic collections view*. All concepts with only one UMLS concept candidate will be highlighted in the color of the semantic collection the candidate refers to. The background of concepts with more than one UMLS concept candidate will be gray.
6. **Mark Phrases**: clicking the "Mark Phrases" button in the toolbar will **delimitate all phrases** in the processed text **with brackets**, indicating which concepts belong to which phrase.
7. Switch to the **phrases mode**: By checking the "Phrases Mode" button, you can switch to the *phrases mode*. Now the phrases in the processed text will be highlighted.
8. Correct phrase chunks (see Figure 3.15):
  - (a) Remove phrase chunks and then add new chunks:
    - Choose the *arrow cursor* from the toolbar to **select a phrase** by double-clicking its text in the editor.
    - click the "Delete Concept / Phrase" button in the toolbar to **delete the selected phrase**. Repeat these two steps as often as is necessary.
    - Choose the *text selection cursor* to **select the text chunk** in the editor you want to create a new phrase chunk from.

- **Create a new phrase chunk** from the selected text by clicking the "Create Concept/Phrase" button in the toolbar. Again repeat the last two steps as often as is necessary.
- (b) Merge two adjacent phrase chunks into one chunk:
- **Select these two phrases** with the "Text Selection" cursor, and
  - **merge these phrases** into one phrase by clicking the "Merge Phrases" button in the toolbar.

[Ovarian cancer] [is] [the fourth most frequently diagnosed cancer] [in women] [in Scotland], [representing] [4.6%] [of all newly diagnosed cancers], [or] [around 600 new cases] [per year] [in Scotland.1 Ovarian cancer] [occurs] [as] [either] [an epithelial] [or] [a non-epithelial tumour]. [Epithelial tumours] [account] [for over 90%] [of all ovarian cancers].

[Ovarian cancer] [is] [the fourth most frequently diagnosed cancer] [in women] [in Scotland], [representing] [4.6%] [of all newly diagnosed cancers], [or] [around 600 new cases] [per year] [in Scotland].1 [Ovarian cancer] [occurs] [as] [either] [an epithelial] [or] [a non-epithelial tumour]. [Epithelial tumours] [account] [for over 90%] [of all ovarian cancers].

Figure 3.14: Before and after correcting the phrase chunk "in Scotland.1 Ovarian cancer".

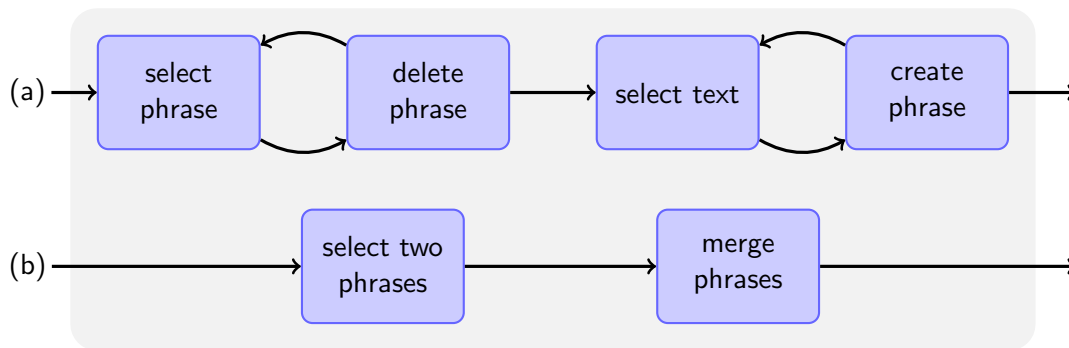


Figure 3.15: Correcting phrase chunks (2 different situations).

9. After doing so for every incorrect phrase chunk, change to the **concepts mode** by clicking the "Concepts Mode" button in the toolbar.
10. Proceed with **correcting concepts chunks** in a similar way as you did with the phrase chunks (see Figure 3.16):
  - **Select a concept chunk** in the editor with the "Arrow Cursor".

- **Delete this concept chunk** by clicking the "Delete Concept / Phrase" button. Repeat these two steps as often as is necessary.
- **Select the text** you want to create a new concept chunk from with the "Text Selection" cursor, and
- **Create a new concept chunk** by clicking the "Create Concept / Phrase" button. Again repeat the last two steps as often as is necessary.

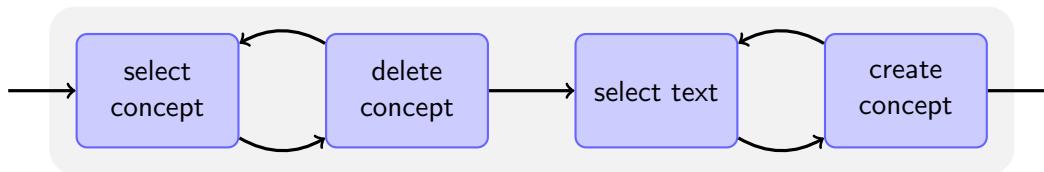


Figure 3.16: Correcting concept chunks.

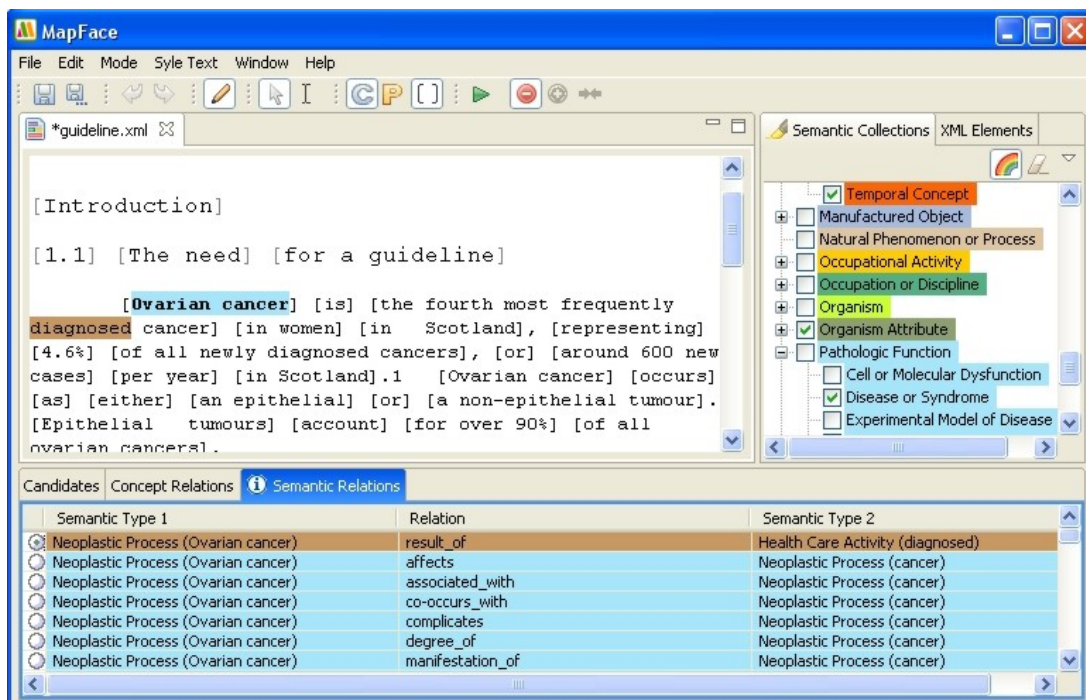


Figure 3.17: Semantic relations of the UMLS concept "Neoplastic Process" to other UMLS concepts affiliated to concept chunks in the same section of the text.

11. After correcting the chunks in the text, you can start to assign UMLS concept candidates to the concept chunks (see Figure 3.18)

- **Select the concept chunk** in the editor, and
- **Select a candidate** in the *candidates view*.
- Now you can take a look at the **relations of the selected concept candidate** to other concepts occurring in the same section of the text by activating the *candidate relations view*,
- As well as activating the *semantic relations view* to take a look at the relations of the semantic type of the selected candidate and the semantic types of other concepts in the same section of the text (see Figure 3.17).
- If you are sure a candidate does not match the selected concept and you want to keep the list clean, **remove the candidate from the list** by selecting it and clicking the "Remove Candidate" button.
- If the correct candidate does not appear in the list, you can **search for additional UMLS concept candidates** by clicking the "Search for Candidates" button and entering an alternative expression for the selected concept.
- Finally **assign the matching candidate** to the selected concept chunk, by clicking the "Assign Candidate" button.

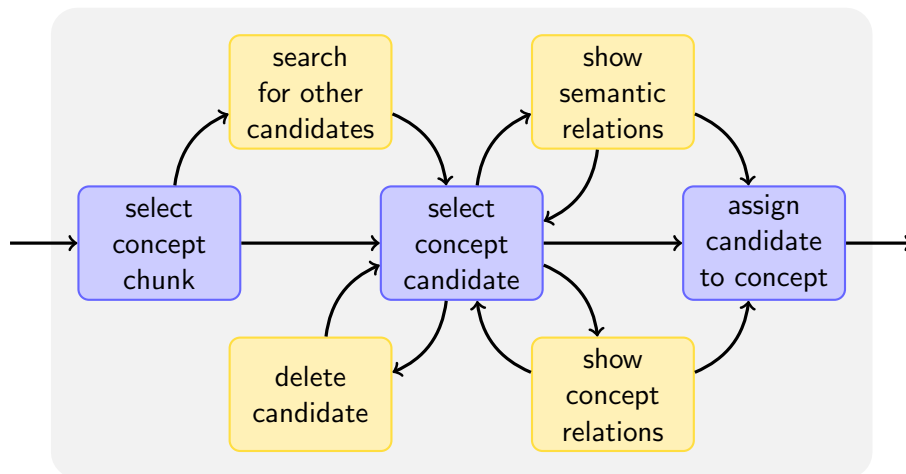


Figure 3.18: Assigning a UMLS concept candidate to a concept. Orange nodes indicate optional actions.

12. Switch to the **phrases mode**.
13. Assign semantic types to phrases (see Figure 3.19):
  - **Select a phrase chunk** in the editor.
  - Now you may **decrease the candidates list** to candidates being of semantic types that are most likely to be correct for the selected phrase by clicking the "Show Favorites" button.

- By selecting a candidate from the list and activating the *semantic relations view* you can take a look at the **relations of the semantic type** of the selected candidate to the semantic types of phrases occurring in the same section of the text.
- Finally **assign the semantic type** you consider to be correct to the phrase chunk by checking its radio button and clicking "Assign Candidate".

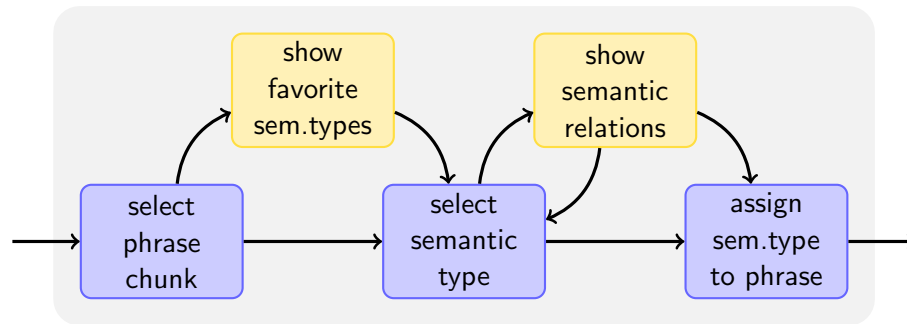


Figure 3.19: Assigning a semantic type to a phrase. Orange nodes indicate optional actions.

### 3.3.6 Extensibility

While developing MapFace I paid special attention to its extensibility. To accomplish this I implemented it as a Java RCP application (using eclipse 3.2 and 5.0 JRE), which has the great advantage that the architecture is based on plugins. This architecture facilitates the extension of MapFace by additional plugins. On the other hand I implemented the editor in a way that enables other plugins to use it. The editor can easily be adapted to other requirements by applying new modes and style schemes to it.

In the following subsection I will list a few details of the architecture of the MapFace application enabling other developers to easily extend it (for example by means of a plugin that would contribute an additional annotation scheme).

#### 3.3.6.1 Defining new Modes for the Editor

A developer can define new modes for the MapFace editor. Such a mode determines the style scheme to use (see Section 3.3.6.2) and the element type to operate on (see Table 3.1). For example the *concepts mode* uses a style scheme called `SemTypesConceptsScheme.java` and operates on the element type `ConceptElement`.

By calling the method

```
setMode (String mode, String elementType, MStyleScheme styleScheme)
```

from the class `MapfaceViewer.java`, the new mode is activated.

- The `String mode` identifies the mode,
- The `String elementType` defines the element type to operate on (e.g., to select on mouse click) (see Table 3.1), and
- The `MStyleScheme styleScheme` is used to style the editor while this mode is active.

ID-String	Element Type
"con"	ConceptElement
"phr"	PhraseElement
"sen"	SentenceElement
"sec"	SectionElement
"xml"	XMLElement

Table 3.1: Id-strings for available element types.

### 3.3.6.2 Creating Style Schemes

A style scheme for the MapFace editor defines `StyleRange` objects (`org.eclipse.swt.custom.StyleRange`) for each element in the editor that is to be highlighted (or somehow styled differently) according to the current conditions. Additionally, `StyleRange` objects for text chunks with bigger font size than the default value (e.g., headlines) are needed. A `StyleRange` object determines the start and end position, the background color, the foreground color, and the font for a given element in the text.

Developers can define new style schemes for the editor by extending the class `MStyleScheme.java` and overwriting its methods.

The class `MStyleScheme.java` extends the class `BasicScheme.java`, which provides several methods for standard activities:

- `getStyle (DocumentElement element, Color background, Color foreground, boolean bold)` returns a `StyleRange` object with the coordinates of the given element, the given background color and foreground color and bold font, if `bold` is `true`.
- `getBasicFontSize ()` returns the font size of normal text in the editor, and
- `getForeground (DocumentElement element)`, which returns black by default and dark-gray for the brackets used to determine phrases in the text.

Developers can use these methods or overwrite them. In addition, they definitely need to overwrite the following methods in the class `MStyleScheme.java`:

- The method

```
getBackground (DocumentElement element)
```

is to return the associated background color (highlighting color) for the given element regardless of its current "highlight state", e.g., if the element is selected in the editor by the user.

- In contrast to this the methods

```
getBackgroundIfChecked (DocumentElement element) and
getForegroundIfChecked (DocumentElement element)
```

are to return the background color and foreground color of the given element according to its "highlight state".

- The most important and also most complex method of this class is the method

`getStyles (DocumentElement element, boolean color, boolean bold)`.

It returns an array list of `StyleRange` arrays (`ArrayList<StyleRange[]>`), containing `StyleRange` objects for the given element and occasionally for elements of its subtree. This is where the developer can create the `StyleRange` objects representing the new style scheme. Overlapping `StyleRange` objects need to be stored in different elements of the array list, otherwise an error will be produced. To optimize execution time it is advisable to keep the array list as short as possible; in other words to put as many parallel (not overlapping) `StyleRange` objects as possible into one array. If `color` is `true` the background of the `StyleRange` object should be colored without respect to its current "highlight state". If `bold` is `true` the `StyleRange` objects should contain bold font.

### 3.3.6.3 Additional Advice for Extensions

To extend the GUI of the MapFace editor, e.g., by an additional view, the developer can either add a new view to the default perspective of MapFace, or create an additional perspective for the application.

The class `MapfacePlugin.java` provides a view-registry. This enables developers to easily access all instances of views contributing to the GUI as well as to register their views with id-strings. Additionally, the class implements an action-registry to provide central access to all registered actions.

The class `MPartListener.java` plays an important role for activating the different modes of the editor. It implements an `IPartListener2` to perform all necessary actions when the state of a GUI-part changes. For example, when the *XML elements view* becomes active, the `MPartListener` disables all actions meant to modify concept chunks or phrase chunks and activates the *XML elements mode* which determines the corresponding style scheme and element type to deal with.

Plugin developers may modify or extend this class.



## Chapter 4

# Evaluation

To evaluate the MapFace editor I chose two different methods: firstly, certain test scenarios in order to demonstrate the usefulness of the features of the editor, and secondly, testing its usability with the help of three non-trained users. Due to the estimated extravagant expenses, a comprehensive usability study was not conducted.

### 4.1 Test Scenarios

In this section I will describe a scenario to demonstrate the necessity to be able to modify and correct the results of the MMTx program with the help of MapFace, and to ensure that MapFace offers possibilities to correct all incorrect results.

#### 4.1.1 Results of the MMTx Program

**Description:** The user selects some text of the guideline and clicks "Run MetaMap".

**Result:** The selected text is processed by means of the MMTx program and the results are re-linked to the text in the editor. The text is tokenized into sections, sentences, phrases, and concepts, and additionally, the final mapping of UMLS concept candidates is computed for each concept chunk. MapFace highlights all detected concept chunks or phrase chunks (*concepts mode*, *phrases mode*) in the editor.

For each kind of result that needs to be modified or corrected, an example is given:

<b>Description</b>	<b>Problem</b>	<b>Example</b>
Tokenization of phrase chunks	phrase chunks need to be split	The phrase "with no family history the lifetime risk" needs to be split into two phrases (see Figure 4.1).
	Phrase chunks need to be merged	The phrase chunks "under the age" and "of 30 years" need to be merged (see Figure 4.2).
Tokenization of concept chunks	Wrong concept chunks	The text "family history" should be one concept chunk (see Figure 4.3).
Mapping of UMLS concepts to concept chunks	Ambiguous mapping	The mapping of a UMLS concept to the concept chunk "woman" is ambiguous (see Figure 4.4).
	Inappropriate mapping	The UMLS concept for the concept chunk "sixth decade" is not correct (see Figure 4.8).

Table 4.1: Testresult of processing guideline text with MMTx.

Additionally, we want to assign a semantic type to each phrase containing at least one UMLS concept. Each step necessary to correct the results of the MMTx program is described in the following subsection.

#### 4.1.2 Correcting phrase chunks

**Case 1:** Splitting a phrase chunk

**Description:** The user deletes the phrase chunk by selecting it in the editor and clicking "Delete Concept / Phrase", whereupon he selects parts of the chunk and clicks "Create Concept / Phrase" to create new phrase chunks.

**Result:** The selected phrase chunk is removed from the guideline text and two new phrase chunks are created (see Figure 4.1).

cancer] [are] [sporadic], [occurring] [in individuals] [with  
no family history] [of the disease]. [Among women] [in  
Scotland] **[with no family history the lifetime risk]** [of  
developing ovarian cancer] [is] [estimated] [to] [be] [1] [in  
59.3] [In 5] [to 10%] [of women] [with the disease], [an

cancer] [are] [sporadic], [occurring] [in individuals] [with  
no family history] [of the disease]. [Among women] [in  
Scotland] **[with no family history]** [the lifetime risk] [of  
developing ovarian cancer] [is] [estimated] [to] [be] [1] [in  
59.3] [In 5] [to 10%] [of women] [with the disease], [an

Figure 4.1: Before and after splitting the phrase chunk "with no family history the lifetime risk".

**Case 2:** Merging two phrase chunks

**Description:** The user selects two phrase chunks in the editor and clicks "Merge Phrases".

**Result:** The two selected phrase chunks are merged into a single phrase chunk (see Figure 4.2).

[The disease] [is] [rare] [in girls] [and] [in women]  
**[under the age]** [of 30 years], [with incidence increasing]  
[with age], [reaching] [its maximum] [in the sixth decade.1]  
[The aetiology] [of the disease] [is] [unknown]. [It] [is]

[The disease] [is] [rare] [in girls] [and] [in women]  
**[under the age of 30 years]**, [with incidence increasing]  
[with age], [reaching] [its maximum] [in the sixth decade.1]  
[The aetiology] [of the disease] [is] [unknown]. [It] [is]

Figure 4.2: Before and after merging the phrase chunks "under the age" and "of 30 years".

### 4.1.3 Correcting concept chunks

**Description:** The user deletes the concept chunk by selecting it in the editor and clicking "Delete Concept / Phrase"; then he/she selects parts of its text and clicks "Create Concept / Phrase" to create new concept chunks.

**Result:** The selected concept chunk is removed from the guideline text and new phrase chunks are created from the selected text (see Figure 4.3).

no family history] [of the disease]. [Among women] [in  
Scotland] [with no family history] [the lifetime risk] [of  
developing ovarian cancer] [is] [estimated] [to] [be] [1] [in  
59.3] [In 5] [to 10%] [of women] [with the disease], [an

no family history] [of the disease]. [Among women] [in  
Scotland] [with no family history] [the lifetime risk] [of  
developing ovarian cancer] [is] [estimated] [to] [be] [1] [in  
59.3] [In 5] [to 10%] [of women] [with the disease], [an

Figure 4.3: Before and after correcting the concept chunk "family history".

#### 4.1.4 Assigning UMLS concepts to concept chunks

**Case 1:** Assigning a UMLS concept in case of ambiguous mapping

**Description:** The selected concept chunk has no assigned UMLS concept (see Figure 4.4). The user selects a UMLS concept from the candidates list. The user clicks "Assign Candidate".

**Result:** The selected UMLS concept is assigned to the selected concept chunk in the editor. When the concept chunk is selected again, the assigned UMLS concept is selected in the candidates list automatically. The background color of the concept chunk changes to the color of its associated semantic type (see Figure 4.5).

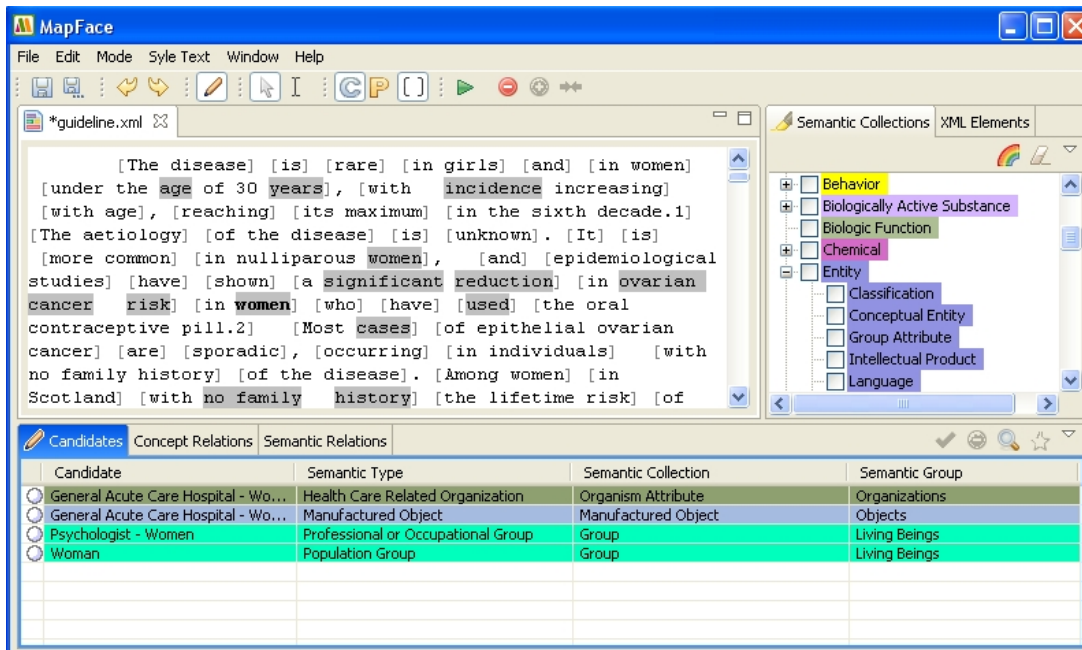


Figure 4.4: The mapping for "woman" is ambiguous.

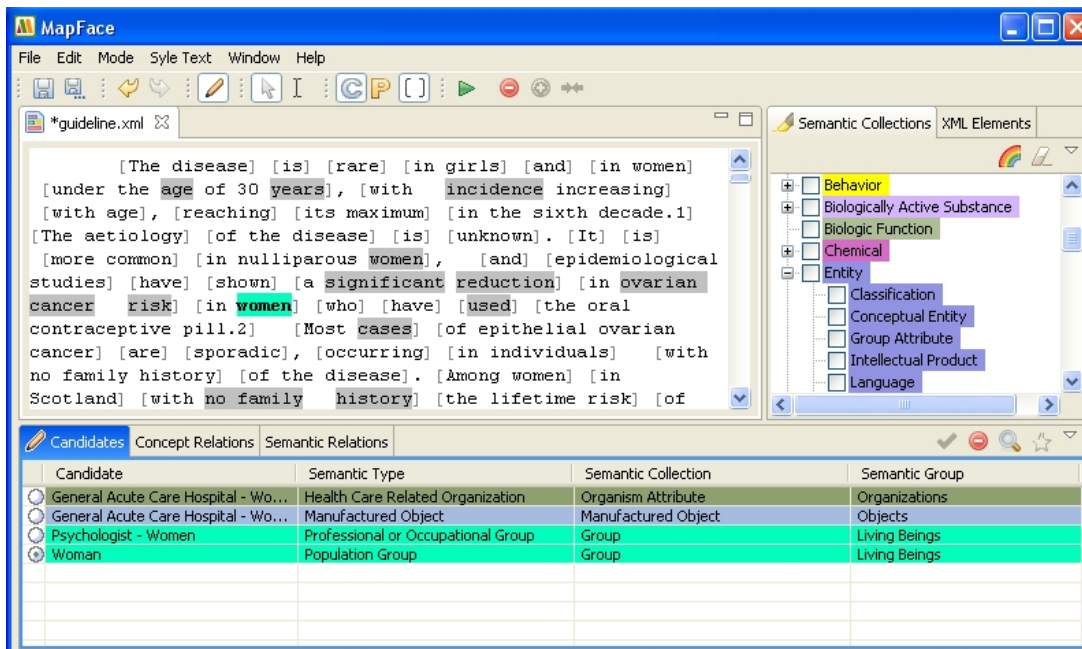


Figure 4.5: After assigning a UMLS concept.

**Case 2:** Removing non-matching UMLS concepts from the candidates list

**Description:** The user selects a UMLS concept from the candidates list. The selected UMLS concept is not assigned to the concept chunk. The user clicks "Remove Candidate" (see Figure 4.6).

**Result:** The selected UMLS concept is removed from the candidates list of the corresponding concept chunk (see Figure 4.7).

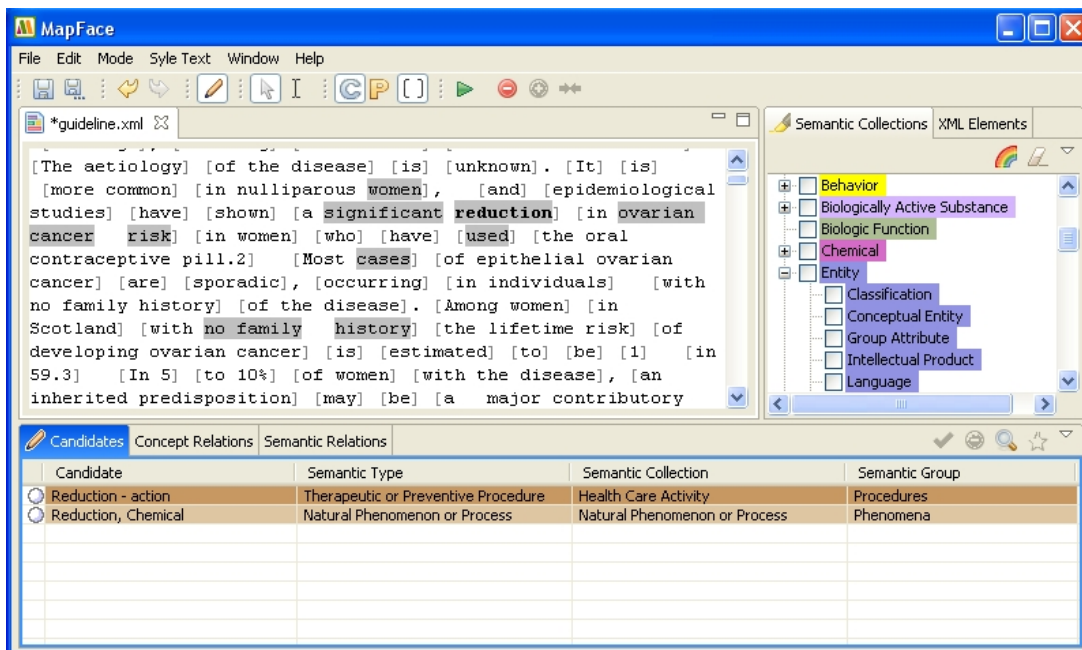


Figure 4.6: Before removing a UMLS concept from the candidates list.

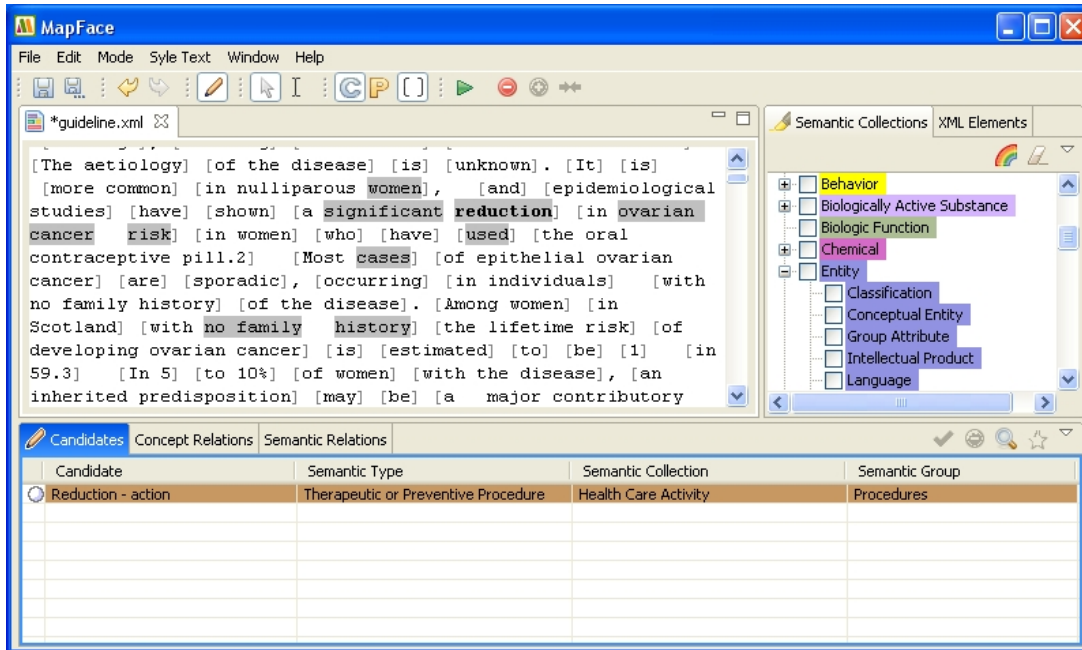


Figure 4.7: After removing a UMLS concept from the candidates list.

**Case 3:** Searching for additional UMLS concept candidates

**Description:** If the candidates list does not contain the correct UMLS concept, the user clicks "Search for Candidates", whereupon he enters an alternative expression for the corresponding concept (see Figure 4.8).

**Result:** An input dialog appears and prompts to enter the alternative expression. The UMLS concepts matching this expression are added to the candidates list. The candidates list does not contain duplicate entries (see Figure 4.9).

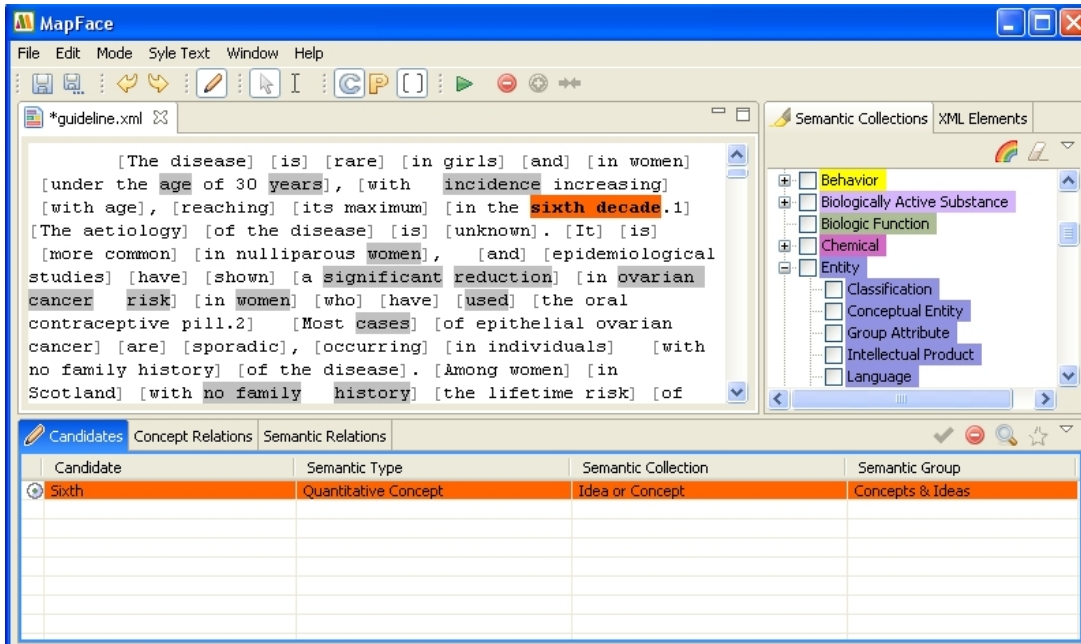


Figure 4.8: The candidates list does not contain the correct UMLS concept for the concept chunk "sixth decade".

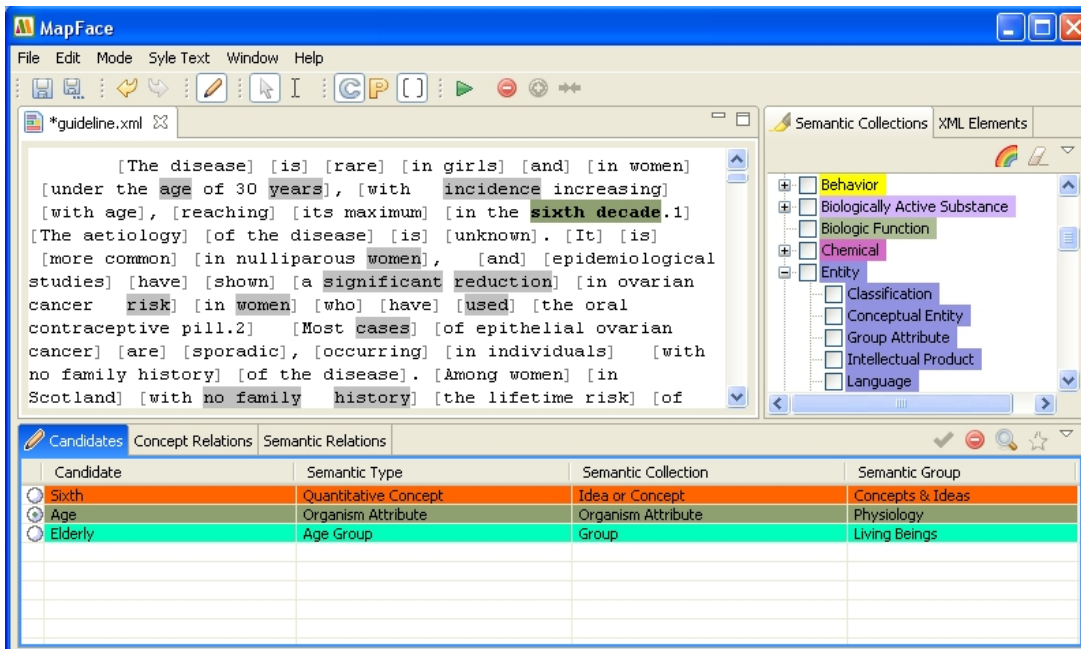


Figure 4.9: After searching for UMLS concepts matching the text "age".



**Case 4:** Displaying information about concept relations

**Description:** To facilitate the decision for a UMLS concept candidate, the user selects a UMLS concept from the candidates list and activates the *concept relations view*.

**Result:** A list of relations between the UMLS concept selected in the *candidates view* and the UMLS concepts assigned to concept chunks in the neighborhood is displayed in the *concept relations view*.

**Case 5:** Displaying information about semantic relations

**Description:** To facilitate the decision for a UMLS concept candidate, the user selects a UMLS concept from the candidates list. The user activates the *semantic relations view*.

**Result:** A list of relations between the semantic type of the UMLS concept selected in the *candidates view* and the semantic types of UMLS concepts assigned to concept chunks in the neighborhood is displayed in the *semantic relations view* (see Figure 4.10).

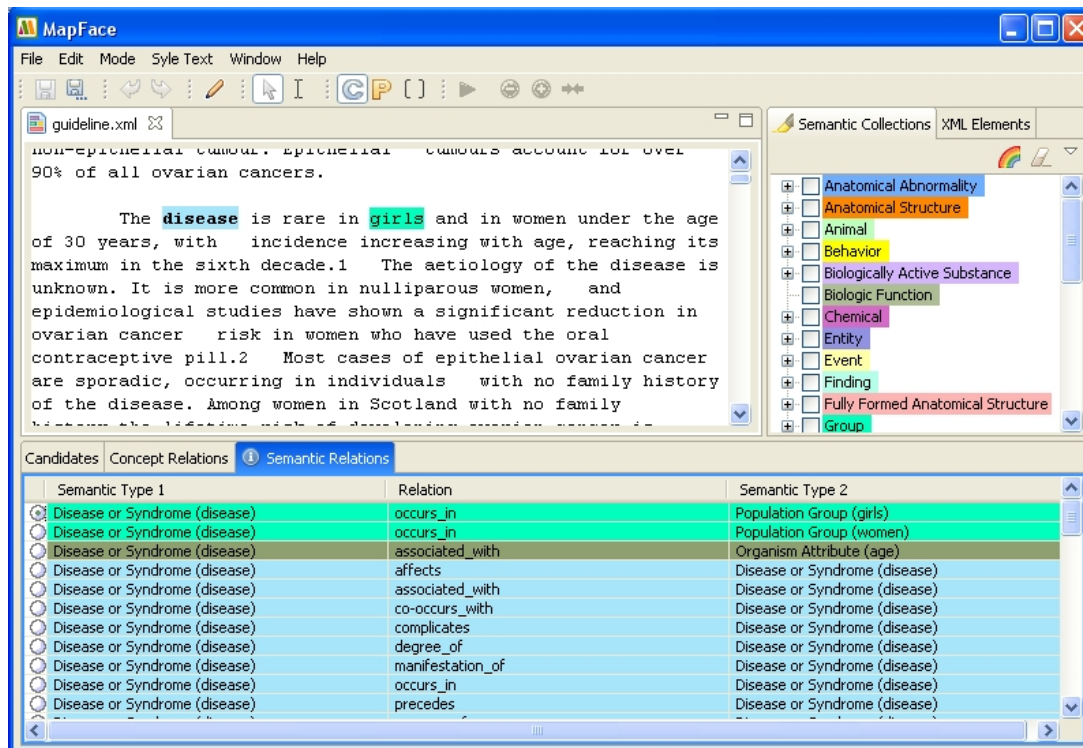


Figure 4.10: Displaying information about the semantic relations of the UMLS concept "Disease".

#### 4.1.5 Assigning semantic types to phrase chunks

**Case 1:** Assigning semantic types to phrase chunks

**Description:** The user selects a phrase chunk in the *phrases mode*, whereupon a list of UMLS concepts together with their semantic types appears in the *candidates view*. The user selects a semantic type from the list and clicks "Assign Candidate" .

**Result:** The selected semantic type is assigned to the phrase chunk. The background color of the phrase chunk changes to the color of its associated semantic type.

## 4.2 Usability Testing

To ensure the usability of MapFace and to receive constructive feedback, a small user study with three participants was performed. At the time of the study, the participants were neither familiar with the MapFace editor and its underlying tools, nor did they have an educational background in medicine or computer science to illustrate the behavior of a non-trained user. The aim of this study was to derive possible lacks of clarity of the MapFace editor and to create a document dealing with its outcome and answering the frequently asked questions.

### 4.2.1 Testing Sessions

Separate testing sessions were performed with each participant. First, the user was introduced to the main aspects of the MMTx program and how to solve a predefined set of tasks with help of the MapFace editor. The list of tasks the user was supposed to solve:

1. Process parts of the guideline text by means of the MMTx program.
2. Correct the tokenization of phrase chunks and concept chunks.
3. Assign UMLS concepts to concept chunks showing ambiguous mapping.
4. Take advantage of the information provided about relations between UMLS concepts and between semantic types.
5. Assign semantic types to phrase chunks.

### 4.2.2 Result

I assisted the participants during the testing sessions and recorded all arising questions as well as my own impressions:

- Due to the complexity of the issue MapFace deals with, the participants needed most advice during the initial phase of the testing session (understanding the tasks).
- All users gave positive feedback regarding the visualization of information by means of color coding and highlighting.

- Since the aim of this user study was to create a document assisting new users to get familiar with MapFace, I created a FAQ document answering relevant questions, which is available in the MapFace editor through the *Help* menu.
- The overall impression was that the participants could manage their tasks very well after an initial phase of getting familiar with the editor.

# Chapter 5

## Conclusion

### 5.1 Summary

In this thesis I introduced the MapFace editor, a useful, even essential tool to edit the results of the MMTx program at a syntactic as well as at a semantic level. Additionally, MapFace is focused on the intelligible visualization of the acquired information.

The MMTx program tokenizes free text of a clinical guideline into sections, sentences, phrases and medical concepts, and maps UMLS concepts to the medical concepts found in the text of the guideline. Due to the complexity of this task, it is still necessary to correct parts of the tokenization as well as the mapping of UMLS concepts. To correct these results with the help of the MapFace editor ensures the quality of the output, which in turn improves the output of subsequent tasks using this information as input.

Besides correcting the information provided by the MMTx program, MapFace serves the purpose of visualizing these results in a transparent and intelligible way. Phrases or medical concepts can be selected in the text of the guideline, whereupon the corresponding information is displayed. Visualizing the semantic information is accomplished by color-coding the semantic types and highlighting the text of the guideline accordingly. In addition, MapFace provides information about relations between the constructs in the text. The visualization of the acquired information is important for knowledge engineers in order to better understand the medical text, thus ensuring the quality of further transformation processes leading to a stable, computer-executable model.

### 5.2 Further Work

In order to extend the MapFace editor to provide all required information for further processing steps, it might be desirable to implement additional plugins that would add certain annotation schemes, e.g., for coreference or negation detection [15].

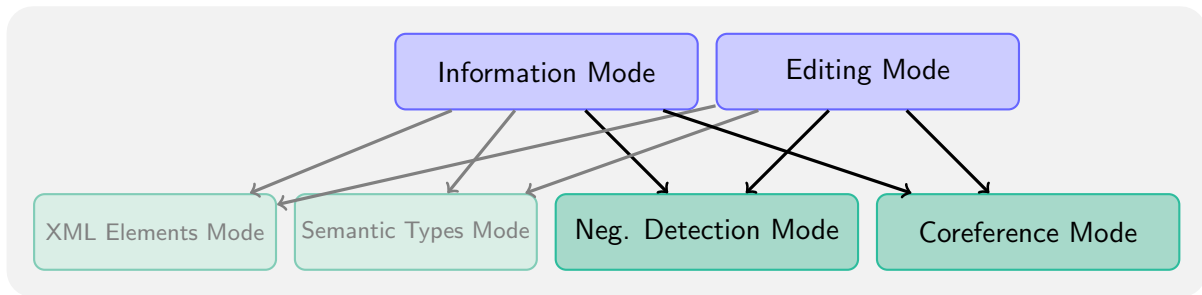


Figure 5.1: Examples of additional modes.

Figure 5.1 illustrates examples of additional modes of the editor, expanding the features of MapFace, in order to provide a comprehensive tool for editing and visualizing clinical guidelines.

# Appendix A

## Use Cases

### A.1 Process the text of the guideline with MMTx

**ID:** UC1

**User:** Physician

**Goal:** The guideline text is split into sections, sentences, phrases, and concepts. Furthermore, semantic and syntactic information about these constructs has been computed with the help of the MMTx program. Each text section of the guideline matching one or more UMLS concepts can be selected, whereupon a list of these UMLS concepts is displayed. In addition, each text section identified as a phrase by the MMTx program can be selected, whereupon a list of the UMLS concept candidates of all included concepts is displayed.

**Preconditions:** *editing mode, semantic types mode*, open XML document.

**Main path:** *complete text*

1. The user clicks the "Run MetaMap" button in the toolbar.
2. The MMTx program processes the text of the whole document by splitting it into sections, sentences, phrases, and concepts, and computing semantic and syntactic information for these constructs.
3. The MapFace GUI re-maps the MMTx results to the text of the guideline and provides access to the computed information.

**Alternative path:** *selected text*

1. The user selects parts of the guideline text.
2. The user clicks the "Run MetaMap" button in the toolbar.
3. The MMTx program processes the selected text only.
4. Go to Step 3 of the main path.

## A.2 Correct the results of the MMTx program

**ID:** UC2

**User:** Physician

**Goal:** To correct and complete the results of the MMTx program assigned to sections of the guideline text.

**Preconditions:** *editing mode, semantic types mode*, open XML document (partially) processed by the MMTx program.

**Main path:** *correct results*

1. The user corrects the text chunks representing phrases.
2. The user corrects the text chunks representing medical concepts.
3. The user assigns UMLS concepts to concept chunks.

## A.3 Correct phrase chunks

**ID:** UC2.1 (extends UC2: *Correct the results of the MMTx program*)

**User:** Physician

**Goal:** To correct (the boundaries of) all phrase chunks in the text of the guideline that were wrongly detected by the MMTx program.

**Preconditions:** *editing mode, semantic types mode, phrases mode*, open XML document (partially) processed by the MMTx program

**Main path:** *delete*

1. The user selects a wrong phrase chunk.
2. The user clicks the "Delete Concept / Phrase" button in the toolbar.
3. MapFace removes the "phrase tag" from the corresponding text of the guideline.

**Alternative paths:** Case 1: *create*

1. The user selects an appropriate part of the text of previously deleted phrase chunks.
2. The user clicks the "Create Concept / Phrase" button in the toolbar.
3. The selected text is tagged to be a phrase and associated with the syntactic and semantic information computed for this phrase by means of the MMTx program. In addition, each text chunk of the phrase matching at least one UMLS concept is tagged to be a concept and linked to the list of matching UMLS concepts provided by the MMTx program.

Case 2: *merge*

1. The user selects two adjacent phrase chunks, which need to be merged into one phrase chunk.
2. The user clicks the "Merge Phrases" button in the toolbar.
3. The two phrase chunks are merged into one phrase chunk; its syntactic and semantic information is refreshed.

## A.4 Correct concept chunks

**ID:** UC2.2 (extends UC2: *Correct the results of the MMTx program*)

**User:** Physician

**Goal:** To correct all concept chunks in the text of the guideline, which were wrongly detected by the MMTx program.

**Preconditions:** *editing mode, semantic types mode, concepts mode*, open XML document (partially) processed by the MMTx program.

**Main path:** *delete*

1. The user selects a wrong concept chunk.
2. The user clicks the "Delete Concept / Phrase" button in the toolbar.
3. MapFace removes the "concept tag" from the corresponding text of the guideline.
4. To delete another concept chunk, go to Step 1.

**Alternative path:** *create*

1. *delete*
2. The user selects an appropriate part of the previously deleted concept chunks.
3. The user clicks the "Create Concept / Phrase" button in the toolbar.
4. MapFace tags the selected text to be a medical concept and associates it with a list of matching UMLS concepts detected by the MMTx program.
5. To delete a concept chunk, go to Step 1 of *delete*.
6. To create another concept chunk, go to Step 2.



## A.5 Assign UMLS concepts to concept chunks

**ID:** UC2.3 (extends UC2: *Correct the results of the MMTx program*)

**User:** Physician

**Goal:** To assign the correct UMLS concept to each concept chunk in the text.

**Preconditions:** *editing mode, semantic types mode, concepts mode*, open XML document (partially) processed by the MMTx program.

**Main path:** *assign concepts*

1. If uncontrolled concept chunks exist in the guideline text, the user selects one of these concept chunks.
2. MapFace displays a list of matching UMLS concepts (if there is only one matching UMLS concept, it is assigned automatically).
3. If the assigned UMLS concept is correct, go to Step 1.
4. If no other candidate exists in the list, search for additional candidates (see Appendix A.7) and return to step 4.
5. The user selects a UMLS concept candidate from the list.
6. If the user is sure about the candidate, go to Step 10.
7. View the corresponding concept relations (see Appendix A.13).
8. If the user is sure about the candidate, go to Step 10.
9. View the corresponding semantic relations (see Appendix A.14).
10. If the selected candidate does not match the concept chunk, remove the candidate from the list (see Appendix A.6) and go to step 5.
11. If the selected candidate matches the concept chunk, go to step 12.
12. The user clicks the "Assign Candidate" button.
13. MapFace links the selected UMLS concept candidate to the concept chunk.
14. MapFace highlights the concept chunk according to the color scheme of the semantic type of the assigned UMLS concept candidate.
15. Go to Step 1.

## A.6 Remove a UMLS concept from the candidates list

**ID:** UC2.3.1 (extends UC2.3: *Assign UMLS concepts to concept chunks*)

**User:** Physician

**Goal:** Remove the selected UMLS concept from the candidates list of the concept chunk selected in the editor.

**Preconditions:** *editing mode, semantic types mode, concepts mode*, selected concept chunk in the editor, selected UMLS concept candidate in the *candidates view*.

**Main path:** *remove*

1. The user clicks the "Remove Candidate" button.
2. If the selected candidate is not assigned to the selected concept chunk as the matching UMLS concept, go to Step 4.
3. MapFace tags the selected concept chunk to be "undefined" (no UMLS concept assigned) and changes its background color in the editor to gray.
4. MapFace removes the selected UMLS concept from the candidates list of the corresponding concept chunk.

## A.7 Search for additional UMLS concept candidates

**ID:** UC2.3.2 (extends UC2.3: *Assign UMLS concepts to concept chunks*)

**User:** Physician

**Goal:** To find additional UMLS concept candidates matching an alternative expression for the selected concept chunk and add them to the list of candidates for this concept chunk.

**Preconditions:** *editing mode, semantic types mode, concepts mode*, selected concept chunk in the editor.

**Main path:** *search*

1. The user clicks the "Search for Candidates" button.
2. MapFace opens an input dialog.
3. The user enters an alternative expression for the concept chunk.
4. The MMTx program computes a list of UMLS concept candidates matching the alternative expression.

## A.8 Assign semantic types to phrase chunks

**ID:** UC3

**User:** Physician

**Goal:** To define a semantic type for each phrase chunk in the text.

**Preconditions:** *editing mode, semantic types mode, phrases mode*, open XML document (partially) processed by the MMTx program

**Main path:** *assign semantic types*

1. If uncontrolled phrase chunks exist in the guideline text, the user selects one of these phrase chunks.
2. MapFace displays a list of UMLS concepts matching the concepts included in the phrase. If there is only one matching UMLS concept, its semantic type is assigned automatically.
3. If the correct semantic type is assigned, or if the list is empty (no concept included in the phrase), go to Step 1.
4. The user selects a semantic type from the list.
5. If the user is sure about the selected semantic type, go to Step 11.
6. The user clicks the "Show Favorites" button.
7. MapFace displays a reduced list of the most likely semantic types for this phrase, computed with the help of IR\_Ex2.
8. If the user is sure about the selected semantic type, go to Step 11.
9. The user activates the *semantic relations view*.
10. MapFace displays a list of relations between the selected semantic type and the semantic types of phrases occurring in the same section of the text.
11. If the user is sure that the selected semantic type matches the phrase chunk, go to step 12, else go to Step 4.
12. The user clicks the "Assign Candidate" button.
13. MapFace assigns the selected semantic type to the phrase chunk and changes its background color in the editor to the color associated with the semantic type.
14. Go to Step 1.

## A.9 Information about concepts in the text

**ID:** UC4

**User:** Knowledge engineer

**Goal:** The knowledge engineer obtains all available information about medical concepts existing in the text of the guideline.

**Preconditions:** *information mode, semantic types mode, concepts mode*, open XML document processed with MapFace.

**Main path:** *concept information*

1. The user selects a medical concept in the text.
2. MapFace displays the matching UMLS concept, its semantic type, its semantic collection, and its semantic group.
3. MapFace highlights the selected concept according to its semantic collection.
4. The user activates the *semantic relations view*.
5. MapFace displays a list of semantic relations between the semantic type of the selected concept and the semantic types of other concepts in the same section of the text.
6. The user selects a semantic relation from the list.
7. Mapface highlights the concerned concepts in the guideline text.
8. If there are semantic relations left in the list, go to Step 6.
9. The user activates the *concept relations view*.
10. Mapface displays a list of relations between the selected concept and other concepts in the same section of the text.
11. The user selects a concept relation from the list.
12. MapFace highlights the concerned concepts in the guideline text.
13. If there are concept relations left in the list, go to Step 11.
14. If there are medical concepts left in the guideline text, go to Step 1.

## A.10 Information about phrases in text

**ID:** UC5

**User:** Knowledge engingeer

**Goal:** The knowledge engineer obtains all available information about phrases existing in the text of the guideline.

**Preconditions:** *information mode, semantic types mode, phrases mode*, open XML document processed with MapFace.

**Main path:** *phrase information*

1. The user selects a phrase in the text.
2. MapFace displays the matching UMLS concept, its semantic type, its semantic collection, and its semantic group.
3. MapFace highlights the selected phrase according to its semantic collection.
4. The user activates the *semantic relations view*.

5. MapFace displays a list of semantic relations between the semantic type of the selected phrase and the semantic types of other phrases in the same section of the text.
6. The user selects a semantic relation from the list.
7. Mapface highlights the concerned phrases in the guideline text.
8. If there are semantic relations left in the list, go to Step 6.
9. If there are phrases left in the guideline text, go to Step 1.

## A.11 Highlight concepts with certain semantic types

**ID:** UC6

**User:** Knowledge engingeer

**Goal:** MapFace highlights all concepts in the text of the guideline corresponding to selected semantic types. The knowledge engineer obtains information about the occurrence of these semantic types in the text.

**Preconditions:** *information mode, semantic types mode, concepts mode*, open XML document processed with MapFace.

**Main path:** *highlight concept*

1. The user selects a semantic type in the *semantic collections view*.
2. MapFace highlights all medical concepts corresponding to this semantic type in the guideline text.
3. If the user wants to highlight another semantic type, go to Step 1.

## A.12 Highlight phrases with certain semantic types

**ID:** UC7

**User:** Knowledge engingeer

**Goal:** MapFace highlights all phrases in the text of the guideline corresponding to selected semantic types. The knowledge engineer obtains information about the occurrence of these semantic types in the text.

**Preconditions:** *information mode, semantic types mode, phrases mode*, open XML document processed with MapFace.

**Main path:** *highlight phrases*

1. The user selects a semantic type in the *semantic collections view*.
2. MapFace highlights all phrases corresponding to this semantic type in the guideline text.
3. If the user wants to highlight another semantic type, go to Step 1.

## A.13 Display concept relations

**ID:** UC2.3.3 (extends UC2.3: *Assign UMLS concepts to concept chunks*)

**User:** Physician

**Goal:** To display a list of relations between the selected UMLS concept candidate and the UMLS concept candidates assigned to concept chunks in the neighborhood of the concept chunk the selected candidate refers to.

**Preconditions:** *editing mode, semantic types mode, concepts mode*, selected concept chunk in the editor, selected UMLS concept candidate in the *candidates view*.

**Main path:** *concept relations*

1. The user activates the *concept relations view*.
2. MapFace displays a list of relations between the selected concept candidate and the UMLS concepts occurring in the same section of the text.
3. The user selects a relation from the list.
4. MapFace highlights the two concerned concept chunks in the editor.

## A.14 Display semantic relations

**ID:** UC2.3.4 (extends UC2.3: *Assign UMLS concepts to concept chunks*, and UC3: *Assign semantic types to phrase chunks*)

**User:** Physician

**Goal:** To display a list of relations between the semantic type of the selected UMLS concept candidate and the semantic types of UMLS concept candidates assigned to concept chunks in the neighborhood of the concept chunk the selected candidate refers to.

**Preconditions:** *editing mode, semantic types mode, concepts mode*, selected concept chunk in the editor, selected UMLS concept candidate in the *candidates view*.

**Main path:** *semantic relations*

1. The user activates the *semantic relations view*.
2. MapFace displays a list of relations between the semantic type of the selected concept candidate and the semantic types of UMLS concepts occurring in the same section of the text.
3. The user selects a relation from the list.
4. MapFace highlights the two concerned concept chunks in the editor.

# Appendix B

## FAQ

### B.1 How can I tell MapFace to structure my XML document?

You can define the names of XML tags of elements in your XML file which are important for structure and readability of your document, such as headlines and list-items. To do this you need to edit the "document\_structure.txt" file in the directory

`%MapFace_Path%/MapFace/plugins/at.ac.tuwien.ifs.ieg.mapface_1.0.0/`,

where you can define the name of the XML tag representing the first headline (h1), the second headline (h2), and the third headline (h3). The text of these elements will have a bigger font-size than the rest of the text. In addition it is possible to define the names of XML tags representing list-elements (iteration) and normal text (text).

This is how the "document\_structure.txt" file initially looks like:

h1 = guideline\_title

h2 = null

h3 = section\_title

text = ElementText

iteration = ElementList

If you don't want to define a certain element, please set it to null, f. e. "h2 = null". Furthermore, you should take care that there is no new line after the last element. Bear in mind that MapFace only displays text between XML tags (no tag-names or -attributes) to keep the text tidy.

### B.2 How can I change the colors MapFace uses to highlight the text?

MapFace reads the colors to use for highlighting from the "colors.txt" file at the directory

`%MapFace_Path%/MapFace/plugins/at.ac.tuwien.ifs.ieg.mapface_1.0.0/`.

This file contains the RGB-values together with a color-name for each color. If you want to

delete, add or change colors, you need to edit this file accordingly. Be sure to have the separation symbol "|" at the beginning and end of each line, as well as after the color-name and between the RGB-values.

For example, a line of the "colors.txt" file can look like this:

```
|blue_velvet|214|181|255|
```

For reasons of readability, don't use very dark colors, since the font-color is black. And remember: no new line after the last element!

### **B.3 How can I make MetaMap process only certain parts of the document?**

If you choose the *text selection cursor* from the toolbar, you can select any text in the editor and then click the "Run MetaMap" button to process the selected text.

### **B.4 How can I make MetaMap process the whole text of the document at once?**

If there is no text selected in the editor, you can click the "Run MetaMap" button to process the whole text of the document at once. Depending on the size of the document, this may take up to a few minutes.

### **B.5 How can I find out which medical concepts were detected by the MetaMap program?**

After the text has been processed by means of the MetaMap program, you can highlight all found concepts or phrases (depending on whether the *prases mode* or the *concepts mode* is active) by clicking the "Highlight All" button in the *semantic collections view*.

### **B.6 How can I find out to which concepts / phrases an UMLS concept / semantic type could not unambiguously assigned?**

Gray background of a concept chunk or phrase chunk in the editor indicates that an UMLS concept / semantic type could not unambiguously assigned. If you click the "No Highlighting" button in the *semantic collections view*, only concepts / phrases with more than one candidate are still highlighted by means of their gray background.

### **B.7 Do I need to control only the concepts with more than one candidate?**

It is necessary to control all detected and assigned concept candidates. If only one candidate has been detected by MetaMap, it was automatically assigned to the concept, but you can search



for additional UMLS concept candidates by clicking the "Search for Candidates" button in the *candidates view* and entering an alternative expression for the text of the concept.

## **B.8 What does it mean, if the background of a concept/phrase is gray?**

The gray background indicates that MetaMap has detected more than one UMLS concept candidate for these concept chunks and phrase chunks. If you double-click (using the *arrow cursor*) the concept chunk or phrase chunk in the editor you can see the list of possible candidates in the candidates view.

## **B.9 How can I highlight all concepts/phrases of the same semantic type?**

By selecting a certain semantic type in the *semantic collections view*.

## **B.10 How can I find out which candidate is assigned to the concept/phrase?**

A candidate that is assigned to a concept/phrase is selected in the list of candidates in the *candidates view*, if you double-click the concept/phrase in the editor. If no candidate is assigned yet, the background color of the concept/phrase is gray.

## **B.11 How can I highlight certain XML elements of the XML document?**

You can select any XML element occurring in the document in the XML elements view (except for the XML elements added by MapFace, whose tag-names start with "mf\_", e.g., the XML tag "mf\_phrase").

## **B.12 How can I choose and assign a UMLS concept to a concept chunk of the guideline text?**

After the text has been processed by MetaMap, you can select a concept in the editor by double-clicking it with the *arrow cursor*. A list best matching UMLS concepts is shown in the candidates view, where you can choose a candidate by selecting the radio button on the left; then click the "Assign Candidate" button to assign it to the concept.

### **B.13 How can I change the UMLS concept candidate assigned to a concept chunk?**

You need to select another candidate from the list in the *candidates view* and simply click the "Assign Candidate" button again. If the correct candidate does not appear in the list of candidates, you can search for additional candidates by clicking the "Search for Candidates" button and entering an alternative text.

### **B.14 What information about concept candidates is available to facilitate my decision?**

In the candidates view, you can see the semantic type, the semantic collection, and the semantic group of each possible candidate for the selected concept. By selecting one candidate from the list and activating the *concept relations view*, you can see a list of existing relations between this concept and other concepts in the same section. To display all existing relations between the semantic type of the selected candidate and the semantic types occurring in the same section, proceed the same way activating the *semantic relations view*.

### **B.15 What, if the correct candidate for a concept does not appear in the candidates list?**

You can search for other candidates by clicking the "Search For Candidates" button in the *candidates view* and entering an alternative text for the concept chunk.

### **B.16 How can I find out to which phrase a concept chunk belongs?**

If you are working in the *concepts mode*, but still want to see the delimitation of phrases, click the "Mark Phrases" button in the toolbar of the MapFace window to surround all phrase chunks with brackets.

### **B.17 Why is it not possible to look for additional candidates for phrases?**

The candidates list for phrases consists of all candidates for the concepts included in this phrase. If a UMLS concept has been assigned to a concept chunk, other UMLS concept candidates matching this concept chunk don't show up in the candidates list for the corresponding phrase.

If you want to look for additional candidates for a phrase, you will have to search for other UMLS concept candidates for one of its concepts in the *concept mode*.

## B.18 How can I modify the beginning and end of a concept/phrase?

In the *concepts mode*, select a concept chunk in the editor (using the *arrow cursor*); then click the "Delete Concept" button to delete this concept. Then you can change the function of the cursor to *text selection* and select the text you want to create a concept chunk from in the editor and click the "Create Concept / Phrase" button. In the *phrases mode* you can delete and create phrase chunks in a similar way.

You can create concept chunks or phrase chunks only from text that has already been processed by MetaMap.

In the *phrases mode*, it is also possible to merge two adjacent phrase chunks by selecting both phrase chunks with the *text selection cursor* and clicking the "Merge Phrases" button.

## B.19 What, if a message "No valid text selected to add a concept/phrase" appears, when I want to create a concept chunk/phrase chunk?

With the "Create Concept / Phrase" button, you can create concept chunks or phrase chunks only from text that has already been processed by MetaMap. If this is not the case, you can add phrases and concepts to the text by selecting it and then running the MetaMap program.

To create a new concept chunk from a certain part of the text, it must not contain any concept chunks. If it does, delete the concept chunks first and then create the new concept chunk.

To create a new phrase chunk, proceed the same way in the *phrases mode*.

## B.20 Will all the information computed with the help of the MMTx program still be available when I save the document and continue work later?

Yes. If you save the processed document, a file of the same name having the extension ".idoc" is saved to the same directory as your XML document. This file contains all information obtained by the MetaMap program.

If you open the XML document again, MetaMap assigns the information from the "\*.idoc" file to the corresponding XML elements, so you can continue your work. Make sure that the "\*.idoc" file is located in the same directory as the corresponding XML file.

## B.21 Why are the buttons to edit the concepts and phrases disabled?

If you don't choose the *editing mode*, all buttons to edit the concepts and phrases are disabled. Clicking the "Editing Mode" button in the toolbar will enable them.

## **B.22 What does it mean, if the semantic types don't show up in the *semantic collections view* after opening the MapFace editor?**

If your system cannot connect to the server on which the umls-database is located, or if the server is down at the moment, MapFace neither load the semantic types nor is able to process the document.

# Bibliography

- [1] Douglas E. Appelt (1999): Introduction to information extraction, in: *AI Communications* 12, 161-172.
- [2] Alan R. Aronson (2001): Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program, in: *Proc. of the Annual AMIA Symposium 2001*, 17-21.
- [3] Alan R. Aronson (2006): MetaMap: Mapping Text to the UMLS Metathesaurus.  
<http://skr.nlm.nih.gov/papers/index.shtml#MetaMap> (last assessed: March 26, 2008)
- [4] Allen C. Browne, Alexa T. McCray, Suresh Srinivasan (2000): The Specialist Lexicon, in: *The SPECIALIST Lexicon Technical Report, 6/2000*, Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, MD.
- [5] Zong Chen, Yehoshua Perl, Michael Halper, James Geller, Huanying Gu (2002): Partitioning the UMLS Semantic Network, in: *IEEE Transactions on Information Technology in Biomedicine, Number: 2, Volume: 6*, 102-108.
- [6] John W. Chinneck (1999): How to Organize your Thesis, Carleton University, Dept. of Systems and Computer Engineering.  
<http://www.sce.carleton.ca/faculty/chinneck/thesis.html>  
(last assessed: March 26, 2008)
- [7] Commission on Professional and Hospital Activities (1978): International Classification of Diseases, Ninth Revision, with Clinical Modifications (ICD-9-CM), United States National Center for Health Statistics, Ann Arbor.
- [8] Roger A. Côté, David J. Rothwell, James L. Palotay, Ronald S. Beckett, Louise Brochu (1993): The Systematized Nomenclature of Medicine, SNOMED International, College of American Pathologists, Northfield, IL.
- [9] Joshua C. Denny, Plomarz R. Irani, Firas H. Wehbe, Jeffrey D. Smithers, Anderson Spickard (2003): The KnowledgeMap project: development of a concept-based medical school curriculum database, in: *Proc. of the Annual AMIA Symposium 2003*, 195-199.  
<http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1480333>  
(last assessed: March 26, 2008)
- [10] The Dutch Institute for Healthcare Improvement CBO.  
<http://www.cbo.nl> (last assessed: March 26, 2008)

- [11] Peter L. Elkin, Mor Peleg, Ronilda Lacson, Elmer Bernstam, Samson W. Tu, Aziz Boxwala, Robert Greenes, Edward H. Shortliffe (2000): Toward Standardization of Electronic Guideline Representation, in: *MD Computing*, 17(6): 39–44.
- [12] Marilyn J. Field, Kathleen N. Lohr (ed.) (1990): Clinical Practice Guidelines: Directions for a New Program, National Academies Press, Institute of Medicine, Washington DC.  
<http://www.nap.edu/books/0309043468/html/index.html>  
 (last assessed: March 26, 2008)
- [13] Robert Gaizauskas, Mark Hepple, Neil Davis, Yikun Guo, Henk Harkema, Angus Roberts, Ian Roberts (2003): AMBIT: Acquiring Medical and Biological Information from Text, in: *S. J. Cox (ed.), Proc. of the Second UK e-Science All Hands Meeting*, Nottingham.
- [14] GATE Information Extraction, The University of Sheffield, Natural Language Processing Group.  
<http://gate.ac.uk/ie/> (last assessed: March 26, 2008)
- [15] Stefan Gindl, Katharina Kaiser, Silvia Miksch (2008, forthcoming): Syntactical Negation Detection in Clinical Practice Guidelines, in: *Conference on Medical Informatics Europe (MIE'08)*.
- [16] William R. Hersh, Robert A. Greenes (1990): SAPHIRE—an information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval, and hierarchical relationships, in: *Computers and biomedical research, an international journal*, 23(5): 410–425.
- [17] Jim Heumann (2001): Generating Test Cases From Use Cases, in: *The Rational Edge, E-zine for the Rational Community*, Rational Software.  
<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf>  
 (last assessed: March 26, 2008)
- [18] Katharina Kaiser (2007): Medical Terminology Systems. Technical Report, in: *Asgaard-TR-2007*, Vienna University of Technology, Institute of Software Technology and Interactive Systems.
- [19] Katharina Kaiser, Cem Akkaya, Silvia Miksch (2006): How Can Information Extraction Ease Formalizing Treatment Processes in Clinical Practice Guidelines? A Method and its Evaluation, in: *Artificial Intelligence in Medicine*, 39:151–163.
- [20] Katharina Kaiser, Silvia Miksch (2005): Information Extraction, A Survey, in: *Asgaard-TR-2005-6*, Vienna University of Technology, Institute of Software Technology and Interactive Systems.
- [21] Donald A. Lindberg, Betsy L. Humphreys, Alexa T. McCray (1993): The Unified Medical Language System, in: *Methods of Information in Medicine, Number 4, Volume 32*, 281–291.
- [22] Ruth Malan, Dana Bredemeyer (2001): Functional Requirements and Use Cases, in: *White Paper 8/3/01*, Bredemeyer Consulting.  
[http://www.bredemeyer.com/pdf\\_files/functreq.pdf](http://www.bredemeyer.com/pdf_files/functreq.pdf) (last assessed: March 26, 2008)

- [23] Alexa T. McCray (1989): UMLS Semantic Network, in: *Proc. of the 13th Annual Symposium on Computer Applications in Medical Care (SCAMC'89)*, 503–507.
- [24] Alexa T. McCray, Anita Burgun, Olivier Bodenreider (2001): Aggregating UMLS Semantic Types for Reducing Conceptual Complexity, in: *Proceedings from the Medinfo 2001 World Congress on Medical Informatics, Volume: 84 Studies in Health Technology and Informatics*, 216–220, IOS Press.
- [25] MetaMap Transfer (MMTx) Information, webpage.  
<http://www.nlm.nih.gov/research/umls/mmtx.html> (last assessed: March 26, 2008)
- [26] Prakash M. Nadkarni(1997): Concept locator: a client-server application for retrieval of UMLS metathesaurus concepts through complex boolean query, in: *Computers and biomedical research, an international journal*, 30(4): 323–336.
- [27] National Institute of Health (2008): Computer Retrieval of Information on Scientific Projects (CRISP).  
<http://crisp.cit.nih.gov> (last assessed: March 26, 2008)
- [28] National Library of Medicine (updated annually): Medical Subject Headings, The Library.
- [29] Lawrence H. Reeve (2007): Semantic Annotation and Summarization of Biomedical Text, Dissertation, Drexel University, College of Information Science and Technology, Philadelphia.
- [30] Lawrence H. Reeve, Hyoil Han (2007): CONANN: An Online Biomedical Concept Annotator, in: *S. Cohen-Boulakia and V. Tannen (Eds.): DILS 2007, LNBI 4544, Springer-Verlag Berlin Heidelberg*, 264–279, Drexel University, College of Information Science and Technology, Philadelphia.
- [31] Ellen Riloff (1999): Information Extraction as a Stepping Stone toward Story Understanding, in: *Ashwin Ram and Kenneth Moorman (ed.), Understanding Language Understanding: Computational Models of Reading*, MIT Press.
- [32] Peri L. Schuyler, William T. Hole, Mark S. Tuttle, David D. Sherertz (1993): The UMLS Metathesaurus: representing different views of biomedical concepts, in: *Bulletin of the Medical Library Association, April, Number: 2, Volume: 81*, 217–222, Medical Subject Headings Section, National Library of Medicine, Bethesda, MD.
- [33] The SPECIALIST NLP Tools Web Page.  
<http://lexsrv3.nlm.nih.gov/SPECIALIST/index.html> (last assessed: March 26, 2008)
- [34] Suresh Srinivasan, Thomas C. Rindflesch, William T. Hole, Alan R. Aronson, James G. Mork (2002): Finding UMLS Metathesaurus concepts in MEDLINE, in: *Proc. of the Annual AMIA Symposium 2002*, 727–731.  
<http://skr.nlm.nih.gov/papers/references/FindingUMLSinMEDLINE.pdf>  
 (last assessed: March 26, 2008)
- [35] Sara Twaddle (2005): Clinical Practice Guidelines, in: *Singapore Medical Journal*, 46(12): 681–687, Scottish Intercollegiate Guidelines Network, Edinburgh.

- [36] U.S. National Library of Medicine (2007): UMLS<sup>®</sup>Knowledge Sources, November Release 2007AC DOCUMENTATION.  
<http://www.nlm.nih.gov/research/umls/umlsdoc.html> (last assessed: March 26, 2008)
- [37] Dennis Wollersheim, Wenny Rahayu, James Reeve (2002): Evaluation of Index Term Discovery in Medical Reference Text, in: *Proc. of the International Conference on Information Technology and Applications*, Bathurst.  
[http://homepage.cs.latrobe.edu.au/dewollershei/papers/ICITA.2002\\_DEWollersheim.PDF](http://homepage.cs.latrobe.edu.au/dewollershei/papers/ICITA.2002_DEWollersheim.PDF) (last assessed: March 26, 2008)
- [38] Yuri L. Ziemann, Howard L. Bleich (1997): Conceptual mapping of user's queries to medical subject headings, in: *Proc. of the AMIA Annual Fall Symposium 1997*, 519–522.
- [39] Qinghua Zou, Wesley W. Chu, Craig Morioka, Gregory H. Leazer, Hooshang Kangarloo (2003): IndexFinder: A Method of Extracting Key Concepts from Clinical Texts for Indexing, in: *Proc. of the AMIA Annual Symposium 2003*, 763–767.  
<http://www.cobase.cs.ucla.edu/tech-docs/zou/INDEXFinder-Final.pdf>  
(last assessed: March 26, 2008)