Industrial Application of Ontologies

Real life examples

Dirk Malzahn
OrgaTech GmbH
Lunen, Germany
dm@orgatech.org

***Abstract*—The industrial application of ontologies is usually connected to a real life problem. Over the last 2 years we used ontologies to solve problems in the areas of retail article management, contract and Request for proposal (RFP) analysis, standard service catalogues and materials management. All these problems were either based on insufficient knowledge of the data and information by the data owner itself, or by semantic and constraints challenges, which could not be resolved due to the complexity and size of the data. In this paper, we will explain how ontologies have been set up and which algorithms have been used to resolve these problems. The combination of ontologies with the analysis of dependencies, text structures, outliers, patterns and similarities lead to an analysis approach and - in the very end - a tool, which on one hand is simple enough to be understood by an industrial expert and on the other hand mature enough to provide the analysis features described above. This paper is more field report than a research paper, but should give an impression that there are areas of application beside the academic world the urgently require ontology based knowledge management.**

Keywords- ontology; mapping; analysis features; service catalogues; article and material management.

I. INTRODUCTION

At eKNOW 2010 we presented a paper about a research project called OPTIKON [1]. The goal of OPTIKON is to develop an ontology based methodology that allows Small and Medium Enterprises (SME) to identify requirements from different standards and map these requirements into one combined set of requirements. This should allow a SME to satisfy a high number of standards with minimal – or at least reasonable – effort.

At the end of the presentation we promised to come back to eKNOW 2011 and explain what happened over the last 12 month with this approach, what worked and what did not.

But, since February 2010 our world changed significantly. Whenever one of our customers came up with a new data problem, our first idea always was “why not trying to solve it with an ontology?” And this is where we are today – solving data problems by thinking in “ontology terms” and trying to help our customers with this approach as much as possible.

We are no researchers – so please forgive us some simplifications. But we hope that our real life examples may give you some benefit and motivation that your work is valued, used and required every day to generate economic benefit.

In section II and III we will explain our initial and final approach. Section IV describes how the required data can be collected. Section V holds a description of the implemented analysis features, whilst section VI show how results are generated from these. In section VII we will show examples of the application in industry before we end with a conclusion in section VIII.

II. FROM MIND TO HAND TO MACHINE

In the beginning of our work, we tried to discuss with our customers (or better the problem owner) that we want to solve their problems by using ontologies. The reaction was opposition. For most of our customers, ontology was an over-the-top theoretical thing, good for universities and philosophers but never fit for purpose to solve their problems.

So we changed our approach – we asked them to draw bubbles. Each bubble should represent a set of information or data they are concerned about. Then we asked them to describe dependencies and interactions between these bubbles by drawing a line. In a next step we asked them whether there is more information required to understand their problem. So they draw more bubbles and connections. Once they had completed this “drawing” we asked them to describe their perfect data world with the same means: bubbles and lines. In the very end we tried to draw lines from the real world to the perfect world, which should represent procedures to resolve the problems in the perfect world.

Bringing it all together: we helped them drawing an as-is ontology, a to-be ontology and a mapping between both.

After this exercise we had some very good sheets of paper, but still one major problem: each bubble represented millions of data fields buried somewhere in a database waiting for being touched by the beauty of an ontology.

Coming from an IT background the solution was right at hand: we needed some kind of tool that allowed us to

- Draw the ontologies
- Assign data to concepts and relations
- Perform analysis on concepts and relations
- Generate results

 ISBN: 978-1-61208-121-2

- Correct identified inconsistencies in data

III. FROM MIND TO HAND TO MACHINE

The starting point for all work was to bring the ontology from a sheet of paper to a system.

A. *Drawing as-is and to-be*

The first required feature was to draw a very basic ontology – containing just concepts (called nodes) and relations (called connections). This feature allowed to draw an ontology of the current situation (as-is) as well as of the intended end stage (to-be).

B. *From as-is to to-be*

As-is and to-be ontologies only make sense if they can be mapped. For this reason three connectors have been created: the direct relation, the split and the combination.

A direct relation ensures that a concept of the as-is ontology can be mapped to a concept of the to-be ontology by applying some set of rules(1:1 mapping).

A split relation allows to divide a concept of the as-is ontology into more than one concept of the to-be ontology (e.g., address may be “splitted” into street, number, postcode, city).

A combination relation does the same as a split relation, only in the different direction (e.g., combines elements of an address).

Due to complexity reasons, a n:m relation was not modeled. Nevertheless it is possible for most cases by a two-step approach. First combination relations (n:1) are used, then the to-be ontology becomes an as-is ontology, on which then split relations are applied (1:m).

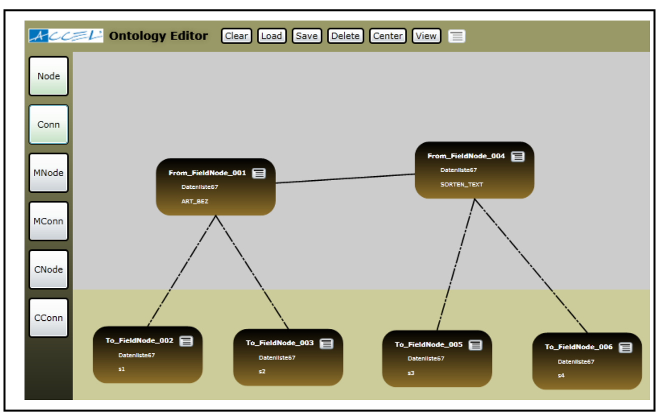


Figure 1. A very basic as-is and to-be ontology (for enlarged picture see Appendix)

IV. GETTING THE DATA

Even though the ontologies build the basis for all work, the user is more interested to see the real world picture of the individuals.

A. *Database input*

The easiest way of assigning individuals is the database input. A concept is assigned to a database column and by this all fields of this column become individuals of a concept.

B. *Text input*

A more sophisticated approach is to use text as an input. If one wants to retrieve data from text there usually are two approaches: if the text is unstructured, the user tags words or text elements that represent a concept or individual; if the text is structured, the user tries to identify the pattern of the structure and afterwards assigns concepts and individuals based on the pattern.

Both approaches have been implemented. By a tag function the user identifies words or text elements that represent a concept. Individuals are tagged in the same way and then assigned to an already existing - or newly tagged – concept.

The implementation of patterns was more challenging. If a text is structured in a pattern, it might contain some table structures, integrated pictures, etc. Therefore it was decided to analyze structured texts by graphical means. Each element of a text structure became a “box”. Based on the size and distance of the boxes, related text elements can be identified as well as maybe missing elements.

1.2.223	**Software as a service**	
	delivered Mon-Fri, based on SLA by customer	12,43 €
	on Saturday	21,00 €
1.2.224	**Software on demand**	
	agile approach	
	structured approach	10,54 €

Figure 2. Example of a structured text

Based on the approach described above, the text will be “boxed” as follows:

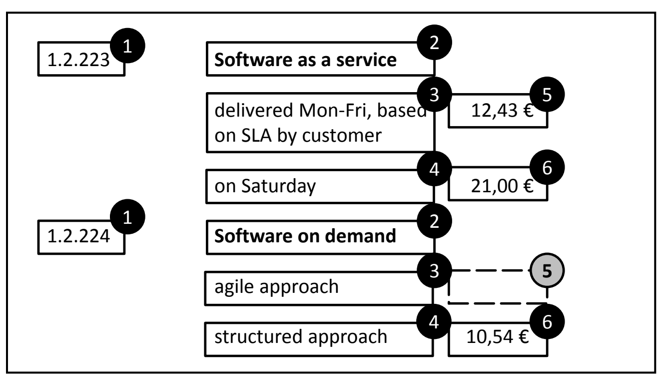


Figure 3. Boxes a structured text

C. Supporting Tables

Text and database input assume that individuals have to be generated from a text or database. But in some cases it might also be possible that the required set of individuals is already know and available. In this case a table is assigned to a concept, which holds only required and valid values.

As this kind of table in most cases has been used to support the validation of individuals from other tables (or concepts), it has been named "supporting table".

V. ANALYSIS FEATURES

By now, all effort has been made to structure information (as-is / to-be ontology, relations, concepts) and assign data (individuals). But the main reason for this effort is to prepare required and intended analysis. Given our current realm of experience, the set of analysis features must at least cover the elements below.

A. Text structure analysis

Text structure analysis looks after the structure and content of the individuals of a concept. If an activity e.g., should consist of a verb, a noun and the number of occurrences, "wash hands twice" would be a valid text structure. "Hands wash twice" will be as well invalid as "wash twice", "twice wash hands" and "wash clean twice".

The major means for text structure analysis are the split between as-is and to-be ontology (to split the text structure into its elements) and the supporting tables (e.g., to limit the number of occurrences).

B. Dependencies

Dependency analysis restricts the number of individuals per concepts. For a to-be ontology it may be required that only individuals containing a specific text or value, starting with a specific number… are allowed to be assigned to a concept. The power of this analysis is driven by the text, mathematical and pattern features applicable to a dependency.

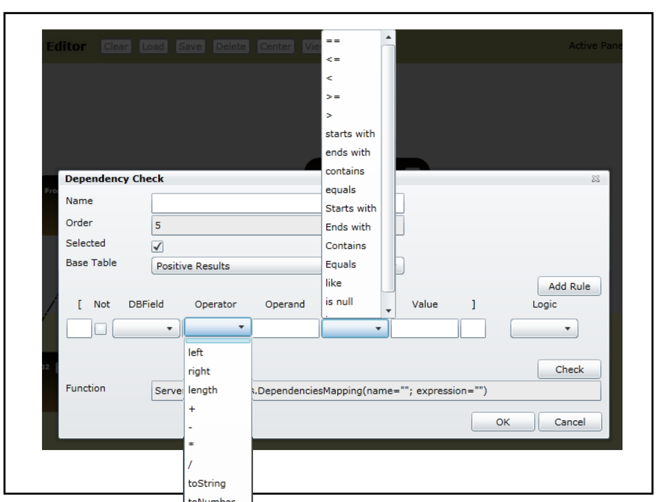


Figure 4. Some elements for dependency definition

C. Outliers

If a high number of individuals are assigned to a concept, it should be checked whether these individuals contain unwanted outliers, which may impact the overall result. Therefore a defined set of top or bottom elements (with regard to the mean or median, based on value or %-age) must be identifiable.

D. Pattern

In extension to the outlier analysis, one should be able to restrict the number of individuals based on a specific pattern. If, for example, a valid date always follows the patter 99. XXX 9999 (where 9 stands for number and X for letter), all individuals not following this pattern should be marked for rework.

E. Duplicates

If individuals are assigned to a concept from a database or text, usually duplicates are generated (e.g., if an address database is analyzed by the split function as described above, most cities will be assigned to the concept "city" multiple times).

To resolve this, it must be possible to identify and delete duplicates.

F. Syntactical similarities

A specific type of duplicates can occur if individuals had been collected or modified manually. Typos or different abbreviations may lead to different syntax for semantically identical elements (e.g., "number of elements" vs. "no of elements" vs. "nr of elements" vs. "numb of elements").

In these cases algorithms like Levenshtein distance [2], with its extension by Ukkonen [3], the Baeza-Yates-Gonnet Algorithm [4] and others may be used.

G. Automated correction

Once the analysis features have identified the correct set of individuals, one has to cope with the incorrect individuals. Again here one will find 2 groups: intended incorrect and unintended incorrect.

Intended incorrect individuals are individuals that break a defined rule and for no applicable other rule this individual is valid. Unintended incorrect individuals still break a defined rule but may deliver a valid result for a parallel or corresponding rule.

If, for example, there is a rule that an address should always consist of street, number, postcode and city, "Lunen, D44536, Zum Pier, 73" has 4 unintended incorrect values as based on the rule and its split into 4 elements, none of the element delivers a valid value at its position; but by resorting the values, a valid structure can be reached.

On the other hand "Zum Pier, 73, D44536, Germany" has an intended incorrect individual as "Germany" does in no case fulfill the requirements of the rule.

To optimize the benefit of the user, for each identified incorrectness it has to be checked, whether the incorrectness can be resolved by either resorting or replacment by a valid value.

H. *Using supporting tables and majorities*

Whilst the automated correction by resorting of values (individuals / parts of individuals) can easily be realized, replacement requires a second look. To replace a value two approaches have been implemented.

The most trustful way is to use supporting tables. If a supporting table contains a value with a high syntactical similarity the original value may be replaced by this, if the calculated syntactical similarity is high enough (based on a threshold, which should be dependent on text length).

If no supporting table is available, the correct value may be identified by majority observation. If, for example, in a database two values have a high syntactical similarity, both are correct (or incorrect) at first sight, and one value occurs e.g., 10 times more often than the other value, it may be assumed that the value with the highest number of occurrences should become the correct value. This should be considered carefully, as e.g., low thresholds for syntactical similarity usually lead to inconsistent replacements.

For completeness, a third approach should be mentioned: whenever two values have a syntactical similarity that allows replacement and one of them is in the set of correct results, this value always replaces the other.

I. *Using meta notes*

Even though not mentioned explicitly yet, all analysis features are limited to 2 concepts, as the basic structure is that one relation only connects 2 concepts.

As in some cases dependency, similarity or other analysis have to be performed on more than 2 concepts at the same time (e.g., only those addresses where postcode start with “45”, city starts with “L” and street starts with “Z”) we invented a so called meta-node. A meta-node is a node that collects information from more than one relation. If the concept address from the as-is ontology is connected to street, number, postcode and city in the to-be ontology by a “has” relation, the meta node itself is connected to the 4 relations and by this allows analysis on these relations and its assigned concepts.

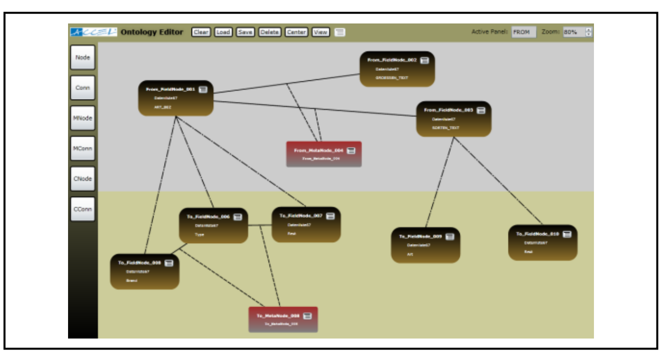


Figure 5. Ontology with meta nodes (for enlarged picture see Appendix)

VI. CREATING RESULTS

By the ontology editor, supporting tables, analysis and correction features and meta-nodes, a powerful set of analysis can be performed – which on the other hand can lead to a typical mismatch.

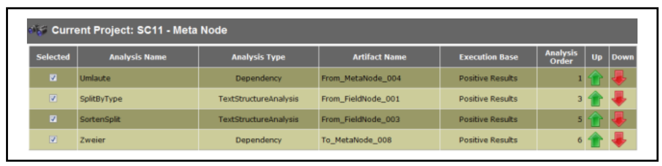
If, for example, one performs a dependency analysis on the first letters of a concept, afterwards splits this concept and then performs a resort, he may receive a different set of results than by first performing the split and resort and then the dependency analysis.

A. *Running and re-running*

To avoid this problem, each analysis should first be run separately to ensure that the results are created as intended. By running and re-running analysis and sequences of analysis, mismatches can be easily identified and a powerful analysis sequence can be developed.

B. *Analysis sequencing*

Once the analysis sequence has been properly set up, one might still wish, to change the analysis sequence (“what happens if…” approach). For this case an analysis sequencer has been implemented to allow not only the resort of values but also a resort of the analysis.



Current Project: SC11 - Meta Node

Selected	Analysis Name	Analysis Type	Artifact Name	Execution Base	Analysis Order	Up	Down
☑	Umlaute	Dependency	From_MetaNode_004	Positive Results	1		
☑	SplitByType	TextStructureAnalysis	From_FieldNode_001	Positive Results	3		
☑	SortenSplit	TextStructureAnalysis	From_FieldNode_003	Positive Results	5		
☑	Zweier	Dependency	To_MetaNode_008	Positive Results	6		

Figure 6. Sequence of 4 analysis (orig. 6, no. 2 and 4 have been deleted)

VII. REAL LIFE EXAMPLES

By now we described approach, procedures and tool on theoretical level. But what is this used for? In this chapter we have collected 4 examples to show the field of application and benefits.

A. *Contract and RFP analysis*

RFPs and contracts are usually longish and full of legal terms.

Lots of companies have made the experience to be closed out from a bidding process just because of formal errors or incomplete proposals. By the analysis features described above it is possible to identify words, elements and rules from a text that describe a must-be, could-be and nice-to-have requirement. This ensures that the company develops a proposal that has all required elements in the right weight and content.

Contracts often have elements that are read over in the beginning (e.g., the usual small print) but lead to severe problems once these elements come to live. If problematic terms from prior contracts are collected in a supporting table, affected elements in new contracts could be identified and a

 ISBN: 978-1-61208-121-2

replacement can be proposed based on elements from successful contracts.

B. Standard Service Catalogues

Standard Service Catalogues (SSC) are used to have a valid and efficient basis for contract negotiations. In large companies, SSC usually have a significant size, and sometimes it is hard to identify the correct service.

This opens the door for fraud. If the service contains e.g., to complete some work measured in inch, and the service provider uses centimeter as basis and "forgets" to add the dimension, the customer pays nearly 4 times of the correct price. This can easily brought to light by building a dependency between length and dimension of contract and invoice.

Another typical trick is to invoice extras. As a SSC cannot cover all possible services, extra charges may be applicable for non-SSC services, and these usually are higher than the SSC charges. For services, which seldom occur, some suppliers use "creative" abbreviations. If the SSC e.g., contains "pipe welding DN25 100 mm" and the supplier charges "DN25 weld pp 100" the identity may not be visible at first sight, but is identifiable by a combination of split, resort, dependency and similarity analysis.

C. Article and Material Management

The standard problem in article and material management is always the text. As article and material text are provided by more than one supplier, are manually extended or changed, not only text structure rules are violated but also inconsistencies to other concepts (e.g., between size and weight) may occur. These problems can be resolved by a combination of text structure analysis with supporting tables, splits and dependency analysis.

D. OPTIKON – still running

That last example should be known by all who attended eKNOW 2010. OPTIKON is a research project that tries to resolve the problem of applying a high number of standards on a project performed by a small or medium enterprise (SME). In reality a SME tries to develop a new locomotive. To get the approval for a new locomotive, several hundred different – but in most cases overlapping – standards have to be fulfilled. A SME is overwhelmed, not only by fulfilling but already by reading and understanding all these standards.

The idea of OPTIKON in brief is to identify the rules defined in each standard (using the text input as described in IV.B and the text structure analysis and patterns as described in V.A and V.D), search for duplicates between the rules from different standards (as described in V.E and V.F), limit the rules based on applicable dependencies and not applicable outliers (as described in V.B and V.C) and correct inconsistencies (see V.G and V.H).

After this exercise a minimum set of required rules are available. Now this set can be compared against the internal rules and standards of the SME – still a lot of work but reduced to a minimum must and cleaned from all ballast.

The result will be the – again smaller – set of rules that have to be fulfilled by the SME in addition to its own procedures.

VIII. Conclusion

In this paper, we have described how very basic elements from the field of ontologies can be used to generate benefits in industry.

In our experience lots of companies – especially in the SME area – are very conservative in using ontologies as they still assign these methodologies only to the academic world.

Nevertheless if ontologies are used in a fit-for-use approach ("just need a sheet of paper and a pencil") and then further developed from this hands-on approach to a structured design supported by sufficient analysis features, it can resolve problems with a complexity the user never was able to handle before.

References

[1] D. Malzahn: Standard compliant process improvement with ontologies - the OPTIKON project. eKNOW 2010, St. Maarten.

[2] www.levenshtein.de. [retrieved: December 9, 2010]

[3] www.psue.uni-hannover.de/wise2009_2010/apzkett/mat/kap4.pdf. [retrieved: December 9, 2010]

[4] www2.informatik.hu-berlin.de/mac/lehre/SS06/Tag_2_Duplikaterkennung.pdf. [retrieved: December 9, 2010]

 ISBN: 978-1-61208-121-2

APPENDIX

Enlarged versions of Figure 1 and Figure 5

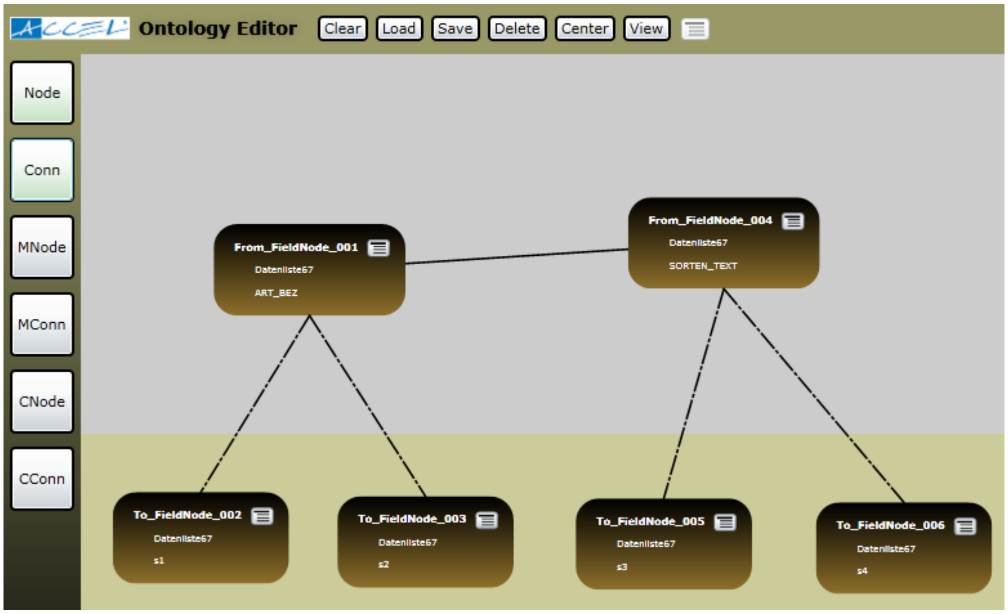


Figure 1. A very basic as-is and to-be ontology

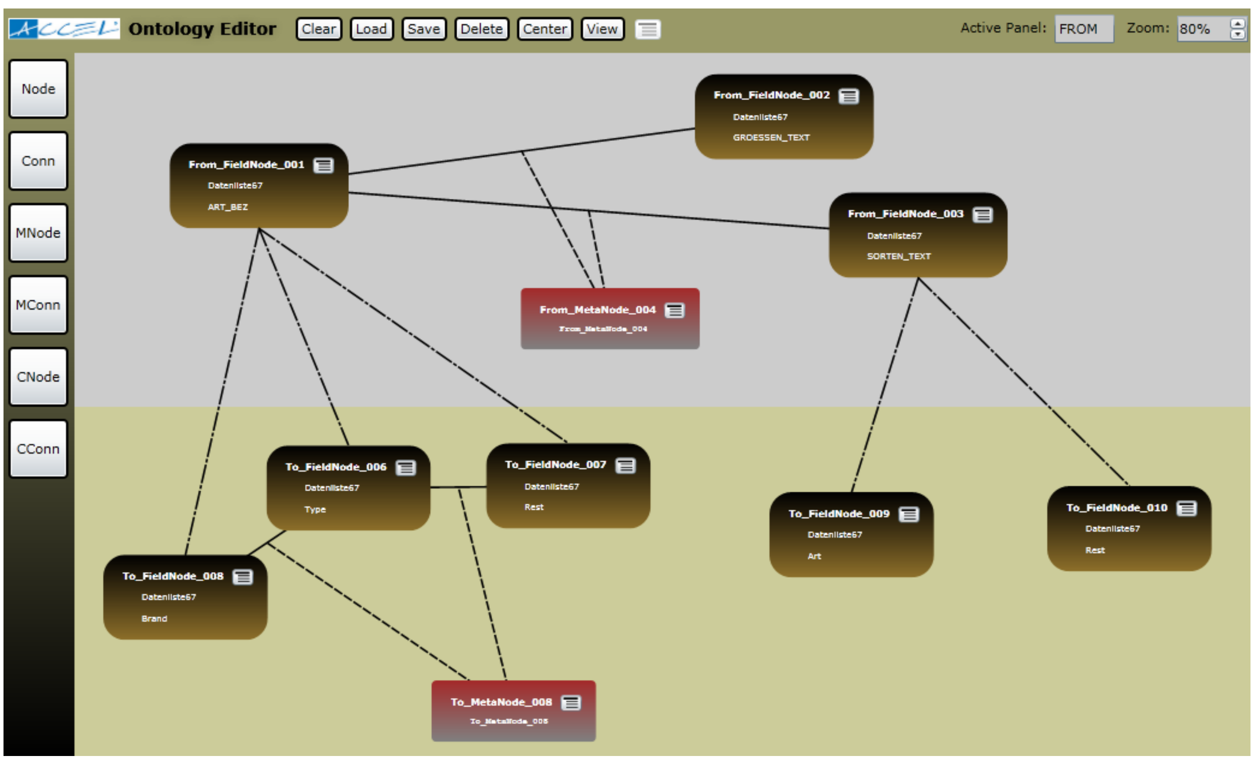


Figure 5 Ontology with meta nodes