Electronic Transactions on Artificial Intelligence
Vol. 4 (2000), Section C: pp. *21-33*
http://www.ep.liu.se/ej/etai/2000/002/

Refereed Article

Uta Priss:

Faceted Knowledge Representation

Author's affiliation: Indiana University, Bloomington, IN, USA.

Abstract: *Faceted knowledge representation provides a formalism for implementing knowledge systems. The basic notions of faceted knowledge representation are "unit", "relation", "facet" and "interpretation". Units are atomic elements and can be abstract elements or refer to external objects in an application. Relations are sequences or matrices of 0's and 1's (binary matrices). Facets are relational structures that combine units and relations. Each facet represents an aspect or viewpoint of a knowledge system. Interpretations are mappings that can be used to translate between different representations. This paper introduces the basic notions of faceted knowledge representation. The formalism is applied here to an abstract modeling of a faceted thesaurus as used in information retrieval.*

Publication and review history

1. *Received* by the research area "Concept Based Knowledge Representation" of the Electronic Transactions on Artificial Intelligence, posted on its web site, and advertised in its Newsletter on 10.3.2000.

2. Available for *discussion* in the ETAI area of Concept Based Knowledge Representation until March–June, 2000. The discussion protocol is available and will be retained in the Article Interaction Page at
   http://www.etaij.org/ra/cbkr/004/

3. First version conditionally *accepted* after due refereeing and according to scientific journal standards by the Electronic Transactions on Artificial Intelligence (ETAI) on 22.9.2000.

4. Revised version *accepted* on 31.10.2000.

5. Revised version appears in Electronic Transactions on Artificial Intelligence, Volume 4 (2000), pages 21–33. The section and the annual volume are made permanently available at
   http://www.ep.liu.se/ej/etai/2000/

6. Paper editions of the ETAI issue and the ETAI volume containing this artice *republished* by the Royal Swedish Academy of Sciences.

7. First and final versions are also *published* as individual articles by Linköping University Electronic Press in December 2000 and January 2001, and are permanently available at
   http://www.ep.liu.se/ea/cis/2000/003/

The review policy and the quality requirements for acceptance are documented at http://www.etaij.org/info/

Article maintenance

The Electronic Transactions on Artificial Intelligence maintains an Article Interaction Page for this article at

http://www.etaij.org/ra/cbkr/004/

Besides the publication and review history, it contains links for contacting the author(s), as well as to amendments and other related information for the article. It is intended to keep these links up-to-date in the future.

Copyright conditions

Introduction

Faceted knowledge representation is a theory of knowledge representation that tries to overcome some of the difficulties involved in designing knowledge representation systems that are flexible enough to be applicable to a variety of domains and users. Facets are viewpoints or aspects, for example, the viewpoint of a specific user group or a specific organizational aspect, such as whether products are arranged according to departments, manufacturing processes or user needs. Faceted knowledge representation systems provide means for coding knowledge according to different facets and for displaying knowledge in a flexible, user-adaptable manner. There are many structures in conventional knowledge representation formalisms that facilitate flexibility and are similar to facets. Examples are classes in object-oriented design, interfaces in languages such as Java, entities and views in relational databases, situations in situation theory (Devlin & Rosenberg, 1996), contexts and scales in formal concept analysis (Ganter & Wille, 1999) and contexts in AI ontologies and in conceptual graphs (Sowa, 1984). Furthermore, a new extension of object-oriented programming called "aspect-oriented programming" (Kiczales et al., 1997) is essentially object-oriented programming with facets. Faceted knowledge representation identifies and formalizes the shared features that all these classes, interfaces, views, situations and contexts have in the definition of "facets". Such a meta-formalism can facilitate the combination of different knowledge representation systems and provide for translations among them.

Facets can be graphically displayed in visualizations, which are interpretations that map conceptual entities onto elements of graphical representations. Without visualizations, facets may be difficult to manage by human users. Faceted knowledge representation does not develop new tools for visualizing but instead facilitates the combination of conventional visualization tools by defining interpretations that translate between different representations. Combining visualization tools is not a new idea. For example, in the management literature, Kingston and Macintosh's (2000) "multi-perspective modelling" of organizational processes aims at combining different visualization and analysis methods. Faceted knowledge representation intends to provide the theoretical foundation for that.

In order to formalize facets at an abstract level, the vocabulary used to describe them must be general enough to be applicable to such similar notions as context, situation, class and perhaps even fractal. Furthermore, the vocabulary should be concise. The three primary notions are unit, relation and facet. Other knowledge representation systems sometimes use different notions for entities that could potentially be formalized in one notion, such as Lenat's (1998) "context" and "dimension" and Devlin and Rosenberg's (1996) "situation" and "device". While such distinctions may be useful for providing a rich vocabulary in applications, concerning facets the focus is on identifying the common features so that facets are applicable to any kind of knowledge representation.

Another basic notion of faceted knowledge representation is the distinction between extensional, set-oriented and intensional, relation-oriented or rule-based representations which can be deliberately separated or combined. There are numerous examples demonstrating the usefulness of combining extensional and intensional representations. For example, logical arguments are often used to intensionally prove a statement, whereas a counter example is often sufficient to extensionally reject a statement. It is in many cases much more difficult to do the opposite: prove statements based on examples and disprove statements based on arguments. Another example is traditional database retrieval, which presents results extensionally as sets. In the case of an empty result set, users are given no clues as to what they need to change to retrieve elements that are based on intensional features closely related to the request, i.e., how to find the "almost hits". In the case of a very large result set, users are given no indication as to how the query can be narrowed down based on intensional features of the query and the database. Thus traditional database retrieval presents results extensionally without utilizing intensional features. A goal of faceted knowledge representation is to improve database retrieval by representing extensional elements of a result set within their intensional relational structures and within a context of related elements.

The notion of "facets" for knowledge representation was first mentioned by Ranganathan (1962) in the field of library classification schemes. Facets originate from Ranganathan's insight that it is usually not useful or feasible to design a classification scheme as a single tree-like hierarchy. This is especially true for universal classification systems in library science which should be applicable to any possible usage or user need. Instead he proposed to group and order classes according to facets. For example, consider classes, such as "American Poetry of the 19th Century". Does this class first belong under "America" and then "Poetry" and "19th Century", or first under the temporal class, then the geographical class and last the literary type class? In a faceted classification system this decision does not have to be made. The term would be classed under "America" in the geographical facet, under "19th century" in the temporal facet, and under "poetry" in the literary type facet. The facets would then be postcombined at the time of retrieval. The notion of "facets" in this paper is related to Ranganathan's but is broader.

Since it is difficult or impossible to implement a faceted classification scheme without software tools for creation, modification and display of the scheme, faceted classification or the notion of "facets" in general was not very well know outside of the field of library science for the last 50 years. The advent of software tools such as databases for flexible management of large data sets and graphical display software has opened numerous possibilities for the use of facets. This paper presents an attempt at defining a comprehensive formal theory of facets in knowledge representation systems that identifies facets as a unique principle or paradigm of knowledge structures. But while some major features of facets are described in this paper, this is still work in progress.

Uniformities, facets and interpretations

Uniformities are conceptual entities. Examples of uniformities are *units*, *relations*, and *facets*. Uniformities can usually be represented by a name, such as "prime number", by an extensional or set-based representation, such as $\{1, 2, 3, 5, 7, 11 \ldots\}$ or by an intensional or rule-based representation, such as "only divisible by 1 or itself". Both extensional and intensional representation assume a context of positive integers because otherwise the dots in the extensional representation would be meaningless and in the intensional representation the concept of "division" would be ambiguous. Other representations, such as a representation by a process or algorithm, or combinations of different kinds of representations are also possible.

Units are atomic uniformities or tuples of atomic uniformities. They are atomic because their properties are defined in terms of their relations to other uniformities but not in terms of their parts or internal features. Units can only be represented by names but there can be synonymous names for a single unit. Units can be elements of sets. Extensional representations usually consist of sets of units.

Relations are sequences of 0's and 1's (unary relations) or matrices of 0's and 1's (binary relations). Intensional representations contain primarily relations but sometimes also some units. Conceptual relations, such as "has part", "owns" or "eats" are not relations in this paper. Instead they are modeled as facets because they contain units and relations.

The sets in the following definition are not meant to be universal in that they would describe all actual or possible uniformities that humans could think of. Instead they are finite sets selected with respect to a context. That is not a limitation but a feature because contexts and facets can be changed, expanded, updated, deleted and combined in many ways. It is one goal of faceted knowledge representation to modularize knowledge representation systems.

Definition 1:
1.1 $\mathcal{U}$ denotes a set of *uniformities*.
1.2 $\mathcal{N}$ ($\subseteq \mathcal{U}$) denotes a set of *units*, $\mathcal{R}$ ($\subseteq \mathcal{U}$) a set of *relations*.
1.3 $\mathcal{N}_{tf}$ ($\subseteq \mathcal{N}$) denotes a set of two units: "true" and "false".
1.4 $\mathcal{R}_u$ ($\subseteq \mathcal{R}$) denotes a set of unary relations (sequences of 0's and 1's); $\mathcal{R}_b$ ($\subseteq \mathcal{R}$) denotes a set of binary relations (binary matrices, arrays of 0's and 1's).

The following definition contains the main operators for units and relations. The operators are listed here without details and logical dependency considerations. The set operators are the normal ones. For further details on the relational operators see Pratt (1992). Since $\mathcal{R}_u$ corresponds to sets, the operations on $\mathcal{R}_u$ also correspond to set operations. Table 1 contains a summary.

Definition 2:

2.1 For $n_1, n_2 \in \mathcal{N}$: $n_1 = n_2 :\Longleftrightarrow n_1$ and n_2 are synonymous in context.

2.2 $\in$ and $\times$ (Cartesian product) denote the usual set operations.

2.3 For a positive integer i, a linearly ordered set $N \subseteq \mathcal{N}$ and $n \in N$:
$n \sim i :\Longleftrightarrow n$ is at the ith position.

2.4 For $r \in \mathcal{R}_u$, $s \in \mathcal{R}_b$, positive integers i, j:
$ir :\Longleftrightarrow i$th position in r is 1; $isj :\Longleftrightarrow (i,j)$th position in s is 1.

2.5 For $r \in \mathcal{R}_u$, $s \in \mathcal{R}_b$, positive integers i, j:
$\neg(ir) :\Longleftrightarrow i$th position in r is 0;
$\neg(isj) :\Longleftrightarrow (i,j)$th position in s is 0.

2.6 Other operations for sets and relations are:
Operators of aggregation: $\cup$, $\cap$, ^{c}
Operators of composition (only for $\mathcal{R}_b$): $\circ$, $\bullet$, ^{d}
Operators of aggregation and composition (only for $\mathcal{R}_b$): $^+$, *
Comparison operators: $=$, $\subset$, $\subseteq$

2.7 Special notations: $\emptyset$ represents the empty set; 0 ($\in \mathcal{R}$) contains only 0's; 1 ($\in \mathcal{R}$) contains only 1's; $1'$ ($\in \mathcal{R}_b$) has 1's on the diagonal, 0's otherwise; $0'$ ($\in \mathcal{R}_b$) has 0's on the diagonal, 1's otherwise.

	name	mapping	definitions
$\cup$	union	$\cup : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$	
^{c}	negation (complement)	$^c : \mathcal{R} \to \mathcal{R}$	
0		$0 \in \mathcal{R}$	
$\cap$	intersection	$\cap : \mathcal{R} \times \mathcal{R} \to \mathcal{R}$	$r \cap s := (r^c \cup s^c)^c$
1		$1 \in \mathcal{R}$	$1 := 0^c$
$\circ$	product	$\circ : \mathcal{R}_b \times \mathcal{R}_b \to \mathcal{R}_b$	
^{d}	inverse (dual)	$^d : \mathcal{R}_b \to \mathcal{R}_b$	
$1'$		$1' \in \mathcal{R}_b$	
$\bullet$	de Morgan compl. of $\circ$	$\bullet : \mathcal{R}_b \times \mathcal{R}_b \to \mathcal{R}_b$	$r \bullet s := (r^c \circ s^c)^c$
$0'$		$0' \in \mathcal{R}_b$	$0' := 1'^c$
$^+$	transitive closure	$^+ : \mathcal{R}_b \to \mathcal{R}_b$	$r^+ := r \circ r \cup r \circ r \circ r \cup \ldots$
*	refl. trans. closure	$^\star : \mathcal{R}_b \to \mathcal{R}_b$	$r^\star := 1' \cup r^+$
$\subseteq$		$\subseteq: \mathcal{R} \times \mathcal{R} \to \mathcal{N}_{tf}$	$r \subseteq s :\Leftrightarrow r \cap s = r$
$=$	equality	$=: \mathcal{R} \times \mathcal{R} \to \mathcal{N}_{tf}$	$r = s :\Leftrightarrow r \subseteq s$ and $s \subseteq r$
$\subset$	containment	$\subset: \mathcal{R} \times \mathcal{R} \to \mathcal{N}_{tf}$	$r \subset s :\Leftrightarrow r \subseteq s$ and not $r = s$

Table 1

A *facet* is a uniformity that provides a viewpoint or aspect of other uniformities. Facets provide context for sets and relations and their combination. All uniformities involved in a single facet are disambiguated which means that within a facet they are unique and have exactly one meaning. If several different facets are to be combined, some of their uniformities may need to be renamed to achieve disambiguation. Formally, facets combine relations, units and other facets. The simplest facets are basic facets (as formalized in the following definition) that consist of one relation and its associated sets of units.

Definition 3:

3.1 A *basic facet* is a uniformity consisting of a relation and a set/sets of units.

3.2 $\mathcal{F}_B$ ($\subset \mathcal{U}$) denotes a set of basic facets.

3.3 $\mathcal{F}_{Bu}$ ($\subset \mathcal{F}_B$) denotes a set of unary basic facets of the form $f = (N; r)$.

3.4 $\mathcal{F}_{Bb}$ ($\subset \mathcal{F}_B$) denotes a set of binary basic facets of the form $f = (N_1, N_2; s)$.

The following conditions must be fulfilled:

3.5 All uniformities within a facet are disambiguated.

3.6 The number of elements in N and N_1, N_2 correspond to the matrix dimensions of r and s, respectively.

3.7 N, N_1 and N_2 are linearly ordered and for $n \in N$, $n_1 \in N_1$, $n_2 \in N_2$, positive integers i, i_1, i_2:
$nr :\Longleftrightarrow n \sim i$ and ir; $\quad n_1 r n_2 :\Longleftrightarrow n_1 \sim i_1$ and $n_2 \sim i_2$ and $i_1 r i_2$.

3.8 The *extension* $ext(f)$ of a basic facet $f \in \mathcal{F}_{Bu}$ consists of N and the set $\overline{N} := \{n \mid n \in N, nr\}$. The *extension* $ext(f)$ of a basic facet $f \in \mathcal{F}_{Bb}$ consists of N_1 and N_2 and the set $\overline{N_1 \times N_2} := \{(n_1, n_2) \mid n_1 \in N_1, n_2 \in N_2, n_1 r n_2\}$.

3.9 The *intension* $int(f)$ of a basic facet is the set of finite (first order) logic formulas without free variables that are true for the facet.

An example of a formula without free variables is "$\forall_{n \in N} : nrn$", which expresses the intensional property "reflexivity" of r within the context of a facet $f = (N, N; r)$. Because of $n \in \overline{N} \Longleftrightarrow nr$ and $n \in \mathcal{N} \backslash \overline{N} \Longleftrightarrow \neg(nr)$ for $f = (N; r)$ and $(n_1, n_2) \in \overline{N_1 \times N_2} \Longleftrightarrow n_1 r n_2$ and $(n_1, n_2) \in \mathcal{N} \backslash \overline{N_1 \times N_2} \Longleftrightarrow \neg(n_1 r n_2)$ for $f = (N_1, N_2; r)$, it follows that with the exception of the order of rows and columns of the relation and of elements in the linearly ordered sets, a basic facet can be re-constructed both from its extension and its intension. Two basic facets are thus equal if they have equal extensions or, which is an equivalent condition, equal intensions.

More complex facets can be built by applying set and relational operators to basic facets. These operations can be described extensionally or intensionally. Table 2 shows examples. All facet constructions in the table require that the original facets share some sets. Each example is assumed in a context that assures that all sets are linearly ordered in a corresponding manner. For example in 1a), the common elements of the sets must be at the same positions. Furthermore, rows and columns with 0's may need to be added to r_1 and r_2 so that they are of equal dimension.

1a) intensional representation: $n(r_1 \cup r_2) :\Leftrightarrow nr_1$ or nr_2 b) extensional representation: $n \in \overline{(N_1 \cup N_2)} :\Leftrightarrow n \in \overline{N_1}$ or $n \in \overline{N_2}$ c) facet: $f_1 = (N_1; r_1)$, $f_2 = (N_2; r_2)$ yield $f_3 = (N_1 \cup N_2; r_1 \cup r_2)$
2a) intensional representation: $n_1 r^c n_2 :\Leftrightarrow \neg(n_1 r n_2)$ b) extensional representation: $(n_1, n_2) \in \overline{\mathcal{N} \backslash N_1 \times N_2} :\Leftrightarrow (n_1, n_2) \in \mathcal{N} \backslash \overline{N_1 \times N_2}$ c) facet: $f_1 = (N_1, N_2; r)$ yields $f_2 = (N_1, N_2; r^c)$
3a) intensional representation: $n_1(r_1 \circ r_2)n_2 :\Leftrightarrow \exists_{n_3} : n_1 r_1 n_3$ and $n_3 r_2 n_2$ b) extensional representation: $(n_1, n_2) \in \overline{N_1 \circ N_2} :\Leftrightarrow$ $\exists_{n_3 \in N_3} : (n_1, n_3) \in \overline{N_1 \times N_3}$ and $(n_3, n_2) \in \overline{N_3 \times N_2}$ c) facet: $f_1 = (N_1, N_3; r_1)$, $f_2 = (N_3, N_2; r_2)$ yield $f_3 = (N_1, N_2; r_1 \circ r_2)$
4a) intensional representation: $n_2 r^d n_1 :\Leftrightarrow n_1 r n_2$ b) extensional representation: $(n_2, n_1) \in \overline{N_2 \times N_1} :\Leftrightarrow (n_1, n_2) \in \overline{N_1 \times N_2}$ c) facet: $f_1 = (N_1, N_2; r)$ yields $f = (N_2, N_1, r^d)$

Table 2

General facets are formalized in the following definition. Obviously, some restrictions are required to ensure that the process of constructing facets from other facets does not result in contradictory constructions. For example, it should be prohibited to construct facet f_1 from f_2 and at the same time construct f_2 from f_1. The list of conditions is not complete. Only some necessary conditions are provided.

Definition 4:
4.1 A *facet* is a relational structure consisting of uniformities and/or sets of uniformities and rules that constrain the uniformities and that are formed using uniformities and operators, such as but not limited to the operators in definition 2.
4.2 The set of facets is denoted by $\mathcal{F}$ $(\subseteq \mathcal{U})$.
4.3 A facet f_1 that is used for constructing another facet f_2 is called a *subfacet* of f_2, denoted by $f_1 \sqsubset f_2$.
The following necessary conditions must be fulfilled:
4.4 $\sqsubset$ is acyclic and transitive.
4.5 Facets that do not contain subfacets are basic facets.
4.6 Only units, sets of units and relations of a facet or its subfacets can be used in rules.
4.7 For every relation in a facet there is at least one basic facet that contains that relation.

Conditions 4.4 and 4.5 ensure that facets form a hierarchy. Basic facets are the building blocks of facets. Condition 4.6 can be further refined, for example by restricting rules to first or second order logic. Rules can be distinguished as to whether they apply to relations and sets of units or to units and specific elements of relations. This distinction is similar to class methods and instance methods in object-oriented programming. The *extension* of a facet refers to the units that are described by the facet. Essentially the extension consists of a set of sets N_i and $\overline{N_i}$. The *intension* refers to first order logic formulas without free variables that are true for the facet. Two facets are equal if they have the same extension or, and this is an equivalent condition, the same intension.

In traditional semantics, an interpretation is defined as a mapping from concepts or conceptual relations onto sets of units or tuples of units of a domain. In other words, traditionally an interpretation maps from a theory onto a model. In faceted knowledge representation that would correspond to a mapping from intensional representations onto extensional representations. But interpretations can also map in the opposite direction or between two extensional or two intensional representations. Interpretations are useful tools for checking consistency and applicability of a facet because translating from one kind of representation, such as an intensional representation, to another kind of representation, such as an extensional representation can make features apparent that are otherwise implicit. In the case of interpretations between extension and intension of a single facet, no information is lost or added. On the other hand, interpretations between different facets can produce a loss of information. In that sense interpretations are similar to homomorphisms in mathematics or Barwise & Seligman's (1997) infomorphisms.

Definition 5:
5.1 An *interpretation* ι is a mapping between uniformities or sets of uniformities.
5.2 The set of interpretations is denoted by $\mathcal{I}$.
5.3 An interpretation ι from D to C is denoted by $\iota : D \to C$.

Table 3 shows some standard interpretations. The interpretation ι_{atm} maps each uniformity or set of uniformities onto a unit. An example is an entity relationship diagram in which entities (i.e., database tables or facets) and relations (facets) are represented as nodes (units). The interpretation ι_{den} corresponds to the notion of an interpretation in description logic. A subset $\mathcal{N}_{\text{den}}$ of $\mathcal{N}$ is identified as a set of denotata, i.e. units that correspond to objects in an application. Each uniformity of the formal description is mapped onto a set of elements of the application. In other words the formal description is assigned a meaning in the context of an application. The interpretations ι^{+}_{den} and $\iota^{\star}_{\text{den}}$ can be used to compute the transitive or transitive and reflexive closure of a relation. The last interpretation in the list, $\iota_{N_1 N_2}$, enables the changing of the dimension of a relation. The relation can be restricted to a subset or extended to a superset by inserting 0's among the new elements.

Name and domains	explanation or definition
$\iota_{\text{atm}} : \mathcal{U} \cup \mathcal{PU} \rightarrow \mathcal{N}$ $\iota_{\text{den}} : \mathcal{U} \rightarrow \mathcal{PN}_{\text{den}}$	interprets uniformity or set of unif. as a unit interprets uniformity as a set of denotata
$\iota^{+}_{\text{ext}} : \mathcal{F}_{Bb} \rightarrow \mathcal{PN}$ $\iota^{\star}_{\text{ext}} : \mathcal{F}_{Bb} \rightarrow \mathcal{PN}$	the extension of a basic facet after replacing r with r^{+} the extension of a basic facet after replacing r with $r^{\star}$
$\iota_{N_3 N_4} : \mathcal{F}_{Bb} \rightarrow \mathcal{F}_{Bb}$	with $f_1 = (N_1, N_2; r_1)$ a facet $f_2 = (N_3, N_4; r_2)$ is created with $\forall_{n_3 \in N_3, n_4 \in N_4} : n_3 r_2 n_4 :\Leftrightarrow n_3 \in N_1, n_4 \in N_2, n_1 r_1 n_2$

Table 3

Operators for facets, such as "facet union", should probably not be defined independently of applications because there are many possibilities. For example, with $f_1 = (N_{11}, N_{12}; r_1)$ and $f_2 = (N_{21}, N_{22}; r_2)$ a "union" could be defined as $f_1 \cup^1 f_2 := (N_{31}, N_{32}; r_3; f_1, f_2; r_3 = r_1 \cup r_2)$ if $N_{31} = N_{21} = N_{11}, N_{32} = N_{22} = N_{12}$. If the sets are not identical, the union could be $f_1 \cup^2 f_2 := \iota_{N_1 \cap N_3 N_2 \cap N_4}(f_1) \cup \iota_{N_1 \cap N_3 N_2 \cap N_4}(f_2)$ or $f_1 \cup^3 f_2 := \iota_{N_1 \cup N_3 N_2 \cup N_4}(f_1) \cup \iota_{N_1 \cup N_3 N_2 \cup N_4}(f_2)$. But $f_3 := (f_1, f_2)$ is also a candidate for facet union. Since it may be difficult to list all possible or sensible operators for facets, it may be useful to at least discuss classes of such operators:

Facet aggregation: a facet is constructed by using only interpretations and operators of aggregation. The four suggested candidates for a union of facets are examples of facet aggregation.

Facet composition: a facet is constructed by using interpretations and operators of composition. This usually means that one facet is considered under the application or aspect of another facet. Selection and filtering are examples of facet composition. Selection can be represented by relational composition with a column vector. Filtering can be represented as relational composition with a matrix that can have 1's only on the diagonal.

Facet abstraction is based on the use of interpretations that group uniformities into sets and then transform the original relations to relations among those sets. Facet abstraction could also be called facet homomorphism. Classification is a form of facet abstraction.

Facet expansion is the counterpart to facet abstraction. Diverse products (Cartesian product, tensor product) for facets are types of facet expansion.

One purpose of interpretations is to analyze the formalized knowledge in a system by shifting it from one representation to another and thereby exploring it under different viewpoints. One advantage is that tools, especially tools for visualization from different existing knowledge representation formalisms can be utilized to graphically represent implicit and explicit knowledge structures. Interpretations that map units, relations or facets onto a graphical representation are called *visualizations* and are defined as follows.

Definition 6:
A *visualization* of a facet is an interpretation that maps uniformities of the facet onto elements of a graphical representation, such as nodes and lines of a graph.

Depending on the application different graphical representations are appropriate: basic facets $f = (N_1, N_2; r)$ can be interpreted as a bipartite graph if $N_1 \cap N_2 = \{\}$ or as an adjacency matrix of a graph if $N_1 = N_2$. A visualization can then graphically represent the graphs. Basic facets $f = (N_1, N_2; r)$ can also be interpreted as a formal context in the sense of formal concept analysis. In that case they are visualized as line diagram of concept lattices (Ganter & Wille, 1999) and the notion of facets is then similar to the notion of multicontexts (Wille, 1996). If ι^{+}_{ext} is applied or the relation is transitive, transitive lines can be omitted in the graphical display. If $\iota^{\star}_{\text{ext}}$ is applied or the relation is reflexive and transitive, transitive and reflexive lines can be omitted in the display. Several displays can be generated for the same facet.

An application: a faceted thesaurus

A faceted thesaurus is a faceted, hierarchical structured set of terms and/or concepts that can be used in information retrieval. The thesaurus provides a controlled vocabulary for assigning terms to documents. The faceted hierarchical display ensures maximum efficiency and control in the design and use of the thesaurus. For further details on faceted thesauri see for example Soergel (1985). The notion of "facet" in faceted thesaurus refers to Ranganathan's use of "facet". The modeling in this paper demonstrates that traditional facets are facets in terms of faceted knowledge representation. The modeling can directly be translated into an object-oriented implementation. Such an implementation can provide a graphical interface for thesaurus construction (in contrast to many currently existing thesaurus tools that are text-based). The faceted modeling ensures that the thesaurus can be designed in a modular fashion and that the modules (facets) are not circular and do not duplicate terms or concepts. The modular approach eases future updates and maintenance of the system. Last but not least, a graphical interface based on the faceted structure can provide user-friendly access to a database with documents that users are interested in searching and browsing.

A thesaurus consists of a generic (IS-A) relation among terms (or concepts), a synonymy relation among terms and possibly other relations, such as a part/whole relation. In this paper only the generic relation is formalized. Details on the other relations are left for future publications. A less technical description of this modeling with further examples can be found in Priss & Jacob (1999). The underlying idea of this modeling is that a relatively small fixed set of terms is used to generate a flexible, possibly large set of concepts. The larger set of concepts provides the vocabulary that is assigned to documents as descriptors but only the smaller set of terms needs to be stored in a database. The main building blocks of the thesaurus are base-line facets. They are defined with the help of an interpretation.

Definition 7:
$\iota_l : \mathcal{F}^g_{Bb} \rightarrow \mathcal{F}_{Bb}$ is an interpretation that maps every facet f_1 in $\mathcal{F}^g_{Bb}$, the set of basic facets for which $N_1 = N_2$ and r_1 is an order relation, onto a basic facet f_2 whose relation r_2 forms a lattice that is the Dedekind closure of the ordered set in f_1.

Definition 8:
A *base-line facet* is a facet of the following form:

$$\begin{aligned} f \;=\; & (C, T, \{t_t\}; r_g, r_c; (T, T, r_g), (C, C, r_c); \\ & (C, C, r_c) = \iota_l((T, T, r_g)), \\ & t_t \in T, \forall_{t \in T} : t r_g t_t, \\ & 1' \leq r_g, r_g \cap r_g^d \leq 1', r_g^+ = r_g) \end{aligned}$$

Definition 8 states that a base-line facet contains two basic facets: (T, T, r_g), a set of terms with a generic relation, and (C, C, r_c), a set of concepts that form a (concept) lattice. The set of terms has one *head term* t_t that is more general than any other term. The conditions relating to r_g state that the relation is reflexive, antisymmetric and transitive. Furthermore the concept lattice is the Dedekind closure of the ordered set of terms.

For the construction of higher-level facets, two operators are defined. The notions of term aggregation and term composition are in analogy to De Morgan's notions of aggregation and composition (compare Priss & Jacob (1999)).

Definition 9:
Let $c(t_t)$ denote the concept that corresponds to the head term and 1^t the matrix that has 1's in the column that corresponds to the head term and 0's otherwise. Let $\iota_{\times N_3 N_4} : \mathcal{F}_{Bb} \rightarrow \mathcal{F}_{Bb}$ an interpretation that maps a facet $f_1 = (N_1, N_2; r_1)$ onto a facet $f_2 = (N_3, N_4; r_2)$ where N_3 is a direct product of N_1 and other sets, N_4 is a direct product of N_2 and other sets and $(\ldots, n_1, \ldots) r_2 (\ldots, n_2, \ldots) :\Leftrightarrow n_1 r_1 n_2$. For $f_i := (C_i, T_i, \{t_{ti}\}; r_{gi}, r_{ci}; \ldots); 1 \leq i \leq n$ the following two methods of facet construction are defined:

Facet construction by term aggregation:

$$\begin{aligned} \oplus(f_1, \ldots, f_n) \;=\; & (C, T, \{t_t\}; r_g, r_l; f_1, \ldots, f_n; \\ & T := (\bigcup_{1 \leq i \leq n} T_i) \cup t_t, \\ & C := (\bigcup_{1 \leq i \leq n} C_i) \cup c(t_t), \\ & r_g := (\bigcup_{1 \leq i \leq n} \iota_{TT}(r_{gi})) \cup 1^t, \\ & r_l := (\bigcup_{1 \leq i \leq n} \iota_{CC}(r_{li})) \cup c(1^t)) \end{aligned}$$

Facet construction by term composition:

$$\odot(f_1, \ldots, f_n) \;=\; (C, T, \{t_t\}; r_g, r_l; f_1, \ldots, f_n;$$

$$
\begin{aligned}
T &:= (\bigcup_{1 \leq i \leq n} T_i) \cup t_t, \\
C &:= (\bigotimes_{1 \leq i \leq n} C_i) \times c(t_t), \\
r_g &:= (\bigcup_{1 \leq i \leq n} \iota_{TT}(r_{gi})) \cup 1^t, \\
r_l &:= (\bigcap_{1 \leq i \leq n} \iota_{\times CC}(r_{li})) \cup c(1^t))
\end{aligned}
$$

In facet construction by term aggregation the union of terms (or concepts) is formed. A new head term (concept) is added so that every facet has a unique head term. An example, are terms such as "cat", "dog", "rat" that can be aggregated in a facet "mammal". In facet construction by term composition the union of terms is formed and the direct product of concepts. An example, are facets, such as "mammal", "color of fur", "size", because the resulting concepts are composed of the terms from each facet, such as "large white cat", "small grey rat", and so on. Concerning the example mentioned in the introduction: a literary type facet, a geographical facet and a temporal facet can each be constructed by term aggregation and then be combined by term composition. This process can be applied recursively and complicated structures can result. For example, only a subfacet of "mammal" might be composed with a facet "typical pet" resulting in term composition inside of term aggregation inside of term composition. The following definition ensures that in a faceted thesaurus, facets are not duplicated and that terms are unique.

Definition 10:
A *faceted thesaurus* is a facet created by recursive use of term aggregation and term composition with the conditions that the subfacet/facet relation forms an ordered set and that for each term there is exactly one minimal facet so that the term is an element of that facet.

Conclusion

Faceted knowledge representation is an attempt at providing a common formal framework for different types of knowledge representation systems and enhancing them with the notion of facets. Using faceted knowledge representation existing knowledge representation systems can be combined and compared. New aspects or viewpoints can be added. The example of a faceted thesaurus demonstrates the applicability of faceted knowledge representation to the construction of term and concept hierarchies, which is sketched in this paper. A previous publication (Priss & Jacob, 1999) contains further details and examples of faceted thesauri. A future publication will demonstrate its usefulness for information retrieval by expanding the term and concept hierarchy with facets for document descriptors and query terms resulting in a graphical representation of queries. Other applications of faceted knowledge representation are visualizations of SQL-based database queries and graphical interfaces for knowledge management and data mining. Future publications will explore these.

References

[Barwise, Jon; Seligman, Jerry (1997)] *Information Flow. The Logic of Distributed Systems.* Cambridge University Press.

[Devlin, Keith; Rosenberg, Duska (1996)] *Language at Work. Analyzing Communication Breakdown in the Workplace to Inform Systems Design.* CSLI Lecture Notes, CSLI Publications, Stanford.

[Ganter, Bernhard; Wille, Rudolf (1999)] *Formal Concept Analysis: Mathematical Foundations*. Springer Verlag, Berlin-Heidelberg.

[Kiczales et al.] Kiczales, G.; Lamping, J.; Mendhekar, A.; Maeda, C.; Videira Lopes, C.; Loingtier, J.; Irwin, J. (1997). *Aspect-Oriented Programming.* In: Proc. of the European Conf. on Object-Oriented Programming, Finland, Springer, LNCS 1241.

[Kingston, John; Macintosh, Ann (2000)] *Knowledge management through multi-perspective modelling: representing and distributing organizational memory.* In: Knowledge Based Systems Journal, Special Issue.

[Lenat, Doug (1998)] *The Dimensions of Context Space.* Cycorp. Available on-line at http://www.cyc.com/publications.html.

[Pratt, Vaughan (1992)] *Origins of the calculus of binary relations.* In: Proc. 7th Annual IEEE Symp. on Logic in Computer Science, Santa Cruz, CA, p 248-254.

[Priss, Uta; Jacob, Elin (1999)] *Utilizing Faceted Structures for Information Systems Design.* Proceedings of the 62st Annual Meeting of ASIS, 1999, p. 203-212.

[Ranganathan, S. R. (1962)] *Elements of library classification.* Asia Publishing House, Bombay, p 45-70.

[Soergel, Dagobert (1985)] *Organizing information.* San Diego, CA, Academic Press.

[Sowa, John (1984).] *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley.

[Wille, Rudolf (1996)] *Conceptual structures of multicontexts.* In: Eklund; Ellis; Mann (eds.). Conceptual structures: representation as interlingua. Springer Verlag, Berlin-Heidelberg-New York, p 23-39.