Faceted Knowledge Representation

Uta Priss
School of Library and Information Science, Indiana University
Bloomington, IN 47405, upriss@indiana.edu

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.

Keywords Knowledge Representation, Formal Concept Analysis, Facets

Introduction

Faceted knowledge representation originates from an attempt to overcome the difficulties involved in designing a knowledge representation system that is applicable to a variety of domains and suits a variety of users. A facet represents a viewpoint or aspect. A faceted knowledge representation system provides a flexible means of coding and displaying knowledge structures depending on adjustable internal or user-defined facets. Facets can be interpreted in the context of conventional knowledge representation formalisms, such as object-oriented design, description logics, relational databases, formal concept analysis (Ganter & Wille, 1999), and conceptual graphs (Sowa, 1984). One goal of faceted knowledge representation is to provide a meta-formalism for combining different knowledge representation systems and for facilitating translations among them. Visualizations, which are interpretations that map sets, relations and facets onto elements of graphical representations, or other tools that traditional knowledge representation formalisms provide can be combined to allow for a display and analysis of knowledge structures under different viewpoints and pertaining to different applications.

Faceted knowledge representation is based on a small set of primitive notions, such as unit, relation and facet, and an open set of logical and relational operators that may differ among applications. An important aspect of faceted knowledge representation is that an extensional, set-oriented, and an intensional, relation-oriented approach are combined. For example, in traditional database retrieval, results are presented 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 closely related to their 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 whether the set contains a relational structure that corresponds to the query. Even if the result set is broken into smaller subsets that are ranked, the ranking is usually dependent on user profiles and heuristic or AI methods but not usually on structural properties of the dataset. In summary, traditional database retrieval presents the results extensionally without referring to intensional features. In faceted knowledge representation a result set is graphically displayed with its relational structure and in a context of related elements (almost hits). Faceted knowledge representation shares this idea with formal concept analysis which represents concepts in their relation to other concepts and to their extensions and intensions.

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 computerization, 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. On the other hand, the notion of facets is related to notions in other fields, such as "views" in relational databases, "polymorphism" in object-oriented programming and "multicontexts" (Wille, 1996) and "scales" in formal concept analysis. Furthermore, a new extension of object-oriented programming called "aspect-oriented programming" (Kiczales et al., 1997) is essentially object-oriented programming with 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. This paper defines the main components of faceted knowledge representation and demonstrates its usefulness in an application.

Uniformities, facets and interpretations

The basic elements (or uniformities) of faceted knowledge representation are *units*, *relations* and *facets*. Units are atomic elements or tuples of atomic elements. 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. Relations are understood as sequences of 0's and 1's (unary relations) or matrices of 0's and 1's (binary relations). They only obtain a meaning if they are applied to sets of units. Conceptual relations that are not unary or binary are modeled as facets (see below). They are not relations in the sense of the following definition.

Definition 1:
$\mathcal{U}$ denotes a set of *uniformities*. Elements of U are denoted by lower case letters, subsets by upper case letters. $\mathcal{N}$ denotes a set of *units* with $\mathcal{N} \subseteq \mathcal{U}$. $\mathcal{N}_{tf}$ denotes a set of two units: "true" and "false". $\mathcal{R}_u$ denotes a set of unary relations (sequences of 0's and 1's); $\mathcal{R}_b$ denotes a set of binary relations (binary matrices, arrays of 0's and 1's); and the union $\mathcal{R} := \mathcal{R}_u \cup \mathcal{R}_b$ denotes a set of relations. The length of a sequence and the dimension of a matrix can be indicated as r^a and $r^{a,b}$, respectively, but these are omitted if obvious in context. All sets in this paper are finite.

A *facet* is a viewpoint or aspect of given uniformities. For constructing facets, operators for sets and relations are required. The following list of operators is only a suggestion and can be modified depending on the requirements of applications. The operators are listed here without definitions and logical dependency considerations. For further details on the relational operators see Pratt (1992).

Definition 2:
Set operators for units and sets of units: $\in$, $\subset$, $\subseteq$, $=$, $\times$ (Cartesian product)
Relational operators: See table 1. They can be grouped as follows:
Operators of aggregation (logical operators): $\cup$, $\cap$, ^{c}.
Operators of composition: $\circ$, $\bullet$, ^{d}.
Operators of aggregation and composition: ${}^+$, *.
Comparison operators: $=$, $\subset$,$\subseteq$.
The sequence or matrix with only 0's is denoted by 0; the sequence or matrix with only 1's by 1; the square matrix with 1's on the diagonal and otherwise 0's by $1'$ and the square matrix with 0's on the diagonal and otherwise 1's by $0'$.

Symbol	name	domain and codomain	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

Relations are meaningless unless they are applied to sets of units as domain and codomain. The purpose of facets is to facilitate the combination of relations and units in a flexible manner. The simplest facets are basic facets (as formalized in the following definition) that consist of one relation and its domain and codomain. For combining relations and sets of units it is essential that the sets are linearly ordered and have the same number of elements as the length of the sequence or the dimension of the matrix.. In order to extend the relational operators to sets of units, it is also required that two sets are ordered in a corresponding manner. For example, if two sets are intersected or joined, their common elements must be at the same positions. In what follows it is always assumed that the sets are "as required" but this is not explicitly mentioned in avoidance of an overly complicated notation.

Definition 3:
A *basic facet* consists of a relation and a set of units as domain and, in the case of binary relations, a set of units as codomain. The notation is $f = (N; r)$ or $f = (N_1, N_2; r)$, respectively. With $N(i)$ denoting the i'th element of set N in f and $r(i)$ (or $r(i,j)$) denoting the i'th (i,j'th) element of a sequence (matrix), the relation is written as nr_f iff $n = N(i), r(i) = 1$ and $n_1 r_f n_2$ iff $n_1 = N_1(i), n_2 = N_2(j), r(i,j) = 1$, respectively. The index "f" can be omitted in context. The *extension* of a basic facet is defined as $ext(f) := \{n \mid n \in N, nr_f\}$ and $ext(f) := \{(n_1, n_2) \mid n_1 \in N_1, n_2 \in N_2, n_1 r_f n_2\}$, respectively. 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. The set of basic facets (denoted by $\mathcal{F}_B$) is a subset

of $\mathcal{U}$. $\mathcal{F}_{Bu}$ and $\mathcal{F}_{Bb}$ denote the subsets of $\mathcal{F}_B$ that contain the unary and binary relations, respectively.

An example of a formula without free variables is "$\forall_{n \in N_1} : nrn$", which expresses reflexivity if $N_1 = N_2$. With the exception of the order of rows and columns (or of the elements in the sequence), the relation can be re-constructed both from the extension and the intension of a basic facet. The intension of a basic facet is influenced by facts about the relation of the sets to each other, such as $N_1 = N_2$ and $N_1 \cap N_2 = \{\}$, but not by the individual units in the sets, which cannot be determined from the intension. The extension on the other hand depends on specific elements in the set of units but information about units that correspond to 0's in a sequence or 0 rows or columns in a matrix is lost. Renaming units usually changes the extension of a facet but not the intension. Two facets with equal extensions can have different intensions.

The relational operators can be used to create facets by applying them to relations from different basic facets that share sets of units. The operators are thus extended to statements about elements in the extension of facets. Table 2 shows examples. The first line in the table corresponds to two basic facets $f_1 := (N_{11}, N_{12}; r_1)$ and $f_2 := (N_{21}, N_{22}; r_2)$ that generate $f_3 := (N_{31}, N_{32}; r_3; f_1, f_2; r_3 = r_1 \cup r_2, N_{31} = N_{21} = N_{11}, N_{32} = N_{22} = N_{12})$. A unit is in $ext(f_3)$ iff it is in $ext(f_1)$ or $ext(f_2)$. The line in the table is part of the intension of f_3. The following general definition of facets formalizes the process of creating facets from other facets. The definition is on purpose somewhat vague. For specific applications more restricted versions of this definition may be useful.

$n_1(r_1 \cup r_2) \Leftrightarrow n_1 r_1$ or $n_1 r_2$
$n_1(r_1 \cup r_2) n_2 \Leftrightarrow n_1 r_1 n_2$ or $n_1 r_2 n_2$
$n_1 r^c n_2 \Leftrightarrow$ not $n_1 r n_2$
$n_1(r_1 \circ r_2) n_2 \Leftrightarrow \exists_{n_3} : n_1 r_1 n_3$ and $n_3 r_2 n_2$
$n_1 r^d n_2 \Leftrightarrow n_2 r n_1$
$n_1 1' n_2 \Leftrightarrow n_1 = n_2$

Table 2

Definition 4:
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. 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 set of facets (denoted by $\mathcal{F}$) is a subset of $\mathcal{U}$. The following conditions must be fulfilled:
a) $\sqsubset$ is acyclic and transitive.
b) Facets that do not contain subfacets are basic facets.

c) Only units, sets of units and relations of a facet or its subfacets can be used in rules.
d) For every relation in a facet there is at least one basic facet that contains that relation.

Conditions a) and b) ensure that facets form a hierarchy. Odd constructions such as $f_1 = (f_2)$ and $f_2 = (f_1)$ are prohibited. Basic facets are the building blocks of facets. Condition c) can be further refined, for example by restricting rules to first or second order logic. The *extension* of a facet refers to the units that are described by the facet. The *intension* refers to first order logic formulas without free variables that are true for the facet. Formulating a precise definition of extension and intension of facets without further conditions of facets in a specific application, is difficult. In the example above the extension of f_3 can be defined as $ext(f_3) := \{(n_1, n_2) \mid n_1 \in N_{31}, n_2 \in N_{32}, n_1 r_3 n_2\}$. But in general an extension of a facet can consist of several unrelated sets of units. No matter how extension and intension are defined in an application, facets are called *extensionally equal (intensionally equal)* if they have the same extensions (intensions). They are called *equal* if they are extensionally and intensionally equal. Facets are therefore not equivalent to concepts as defined in formal concept analysis because concepts are uniquely defined by either extension or intension. Furthermore the set of concepts of a formal context forms a lattice whereas the hierarchy of facets is not usually a lattice.

Extensions are special kinds of *interpretations*, which translate from one representation, such as a facet, to another, such as a set of units (compare the following definition). Intensions in general are not interpretations but are similar to a mapping from facets to relations, which is an interpretation. Interpretations can map units onto elements of a domain or relations onto a graphical representation. Basic facets $f = (N_1, N_2; r)$ can be interpreted as a bipartite graph if $N_1 \cap N_2 = \{\}$ and as an adjacency matrix of a graph if $N_1 = N_2$. They can also be interpreted as a formal context in the sense of formal concept analysis. In that case the notion of facets is similar to the notion of multicontexts (Wille, 1996).

Definition 5:
An *interpretation* ι is a relation that fulfills the condition $(1'^c \circ \iota^d)^c = \iota^d$ (which means that it is a mapping). Domain and codomain of an interpretation are subsets of $\mathcal{U} \cup \mathcal{PU}$ (where $\mathcal{PU}$ denotes the power set of $\mathcal{U}$). The set of interpretations is denoted by $\mathcal{I}$. With domain D and codomain C the notation $\iota : D \to C$ can be used.

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 logics. A subset $\mathcal{N}_{\text{den}}$ of $\mathcal{N}$ is identified as a set of denotata, i.e. external 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, $\iota_{N_1N_2}$, enables the changing of domain and codomain of a relation. Using this interpretation the dimension of a relation can be changed: 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
$\iota_{\text{atm}} : \mathcal{U} \cup \mathcal{PU} \to \mathcal{N}$ $\iota_{\text{den}} : \mathcal{U} \to \mathcal{P}\mathcal{N}_{\text{den}}$	interprets uniformity or set of unif. as a unit interprets uniformity as a set of denotata
$\iota^{+}_{\text{ext}} : \mathcal{F}_{Bb} \to \mathcal{PN}$ $\iota^{\star}_{\text{ext}} : \mathcal{F}_{Bb} \to \mathcal{PN}$	the extension of a basic facet after replacing r with r^{+} the extension of a basic facet after replacing r with $r^{\star}$
$\iota_{N_3N_4} : \mathcal{F}_{Bb} \to \mathcal{F}_{Bb}$	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 can be to analyze the formalized knowledge in a system by shifting it from one modeling 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: a binary relation can be visualized as a bipartite graph, as a concept lattice in the sense of formal concept analysis, or as an adjacency graph. If ι^{+}_{ext} is applied or the relation is transitive, transitive lines can be omitted in the graphical display. If $\iota^{\star}_{\mathrm{ext}}$ 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* 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:

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

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. 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

Ganter, Bernhard; Wille, Rudolf (1999). *Formal Concept Analysis: Mathematical Foundations.* Springer Verlag, Berlin-Heidelberg. (English translation of: Formale Begriffsanalyse: Mathematische Grundlagen. Springer Verlag, Berlin-Heidelberg, 1996.)

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.

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.* Proc. of ASIS 1999 Annual Meeting. (to appear)

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.