Faceted Information Representation

Uta Priss

School of Library and Information Science, Indiana University Bloomington,
upriss@indiana.edu

Abstract. This paper presents an abstract formalization of the notion of "facets". Facets are relational structures of units, relations and other facets selected for a certain purpose. Facets can be used to structure large knowledge representation systems into a hierarchical arrangement of consistent and independent subsystems (facets) that facilitate flexibility and combinations of different viewpoints or aspects. This paper describes the basic notions, facet characteristics and construction mechanisms. It then explicates the theory in an example of a faceted information retrieval system (FaIR).

1 Introduction

Facets are relational structures consisting of units, relations and other facets selected for a certain purpose. The term "facets" has been independently introduced in the 1950's in two separate fields. First, Ranganathan [11] used the term to denote aspects or viewpoints in library classification systems. One problem with classification systems is usually that items can be classed differently based on different purposes. For example, a piano is a musical instrument for the purpose of an abstract typology but a piece of furniture for the purpose of interior design. Ranganathan's idea was that class hierarchies for different purposes (or facets) can be built and combined. His top-level facets were personality, matter, energy, space and time. Second, independently of Ranganathan, Guttman [5] used the term "facet" for designing sociological surveys. In a survey it is important to cover the complete spectrum of every aspect of a population or topic in a "facet analysis". For example, if unemployed men age 20 to 30 and 30 to 40 are surveyed or interviewed then three facets are involved: employment status (employed, unemployed), gender (male, female) and age range. It is important for survey design to include all possible combinations of these, otherwise the results could be non-representative.

Apart from these two occurrences of the term "facets", the notion can be observed under different names in many areas throughout history. Ramon Lull's 13th century wheels [4] for combining principles of life and divine and human characteristics represent facets. Modern examples are scales in formal concept analysis (FCA) [3] and fields in relational databases. In each case, the notion of facets or whatever they are called is defined slightly differently but there is a strong overlap in meaning. For example, in each case a baseline facet consists of a small limited subset of values, concepts or isolates (Ranganathan). Facets of similar type can be aggregated to form larger facets. Different but related facets can be composed, which yields all possible combinations of values from different facets. Since the number of possible combinations grows quickly,

hardware such as Lull's wheels or software such as TOSCANA in formal concept analysis are required to manage composed facets. The need for operating with facets via software or hardware is probably the reason why in paper-based traditional library classification, Ranganathan's ideas have never been widely adapted. Another shared characteristic of facets is that they usually represent different viewpoints, aspects or levels of specification of a single domain. With respect to that characteristic facets relate, for example, to aspect-oriented programming, FCA contexts and multicontexts, AI frames and contexts, relational database views, and situations in situation theory.

There are a few systems that explicitly implement facets, such as a retrieval system for software reuse [8]. In that approach facets are distinguished but the facets are just lists of classes clustered by similarity without further structure. A very interesting system is HyTropes [2], an object-oriented web-based knowledge management system. It is somewhat similar to the FaIR (Faceted Information Retrieval) system described in this paper but it does not use lattices for its class hierarchies. Furthermore FaIR's formalism requires fewer primitives and FaIR's display is simpler. Among library access systems, Hibrowse [7] is an exceptional system that employs facets. Apart from these and possibly a handful of other, lesser known systems there is very little interest in facets, their applications and their theoretical foundation. This is unfortunate in the light of the advantages of using facets, such as the four-fold improvement of precision/recall described in [8].

This paper aims at describing the re-occurring principle of identifying, listing, distinguishing, aggregating and composing in an abstract notion of facets. The goal is to establish a theory of facets. As an example the applicability of the facet principle to information retrieval is demonstrated in the FaIR system.

2 Faceted Knowledge Representation

It should be noted that in the following description of facets some notions, such as purpose and representation, are not presented in a mathematical formalism because they are meta-mathematical. Other notions, such as interpretation and facet, can be described with mathematical symbols but their main properties, such as facet identity, depend on the purpose of a facet and can therefore not be denoted independently of specific systems or applications.

Faceted knowledge representation (FKR) provides a framework for the definition of units, relations and facets. These three are loosely motivated by Peirce's Firstness, Secondness and Thirdness. *Units* are items characterized by their relations to other items but not by their internal features. Examples are objects and attributes of formal concept analysis (FCA). Complex items, such as FCA contexts or database tables, can also be considered as units if their internal features are ignored and statements about them as a whole and their relationships to other complex items are made. For example, in an entity relationship diagram, database tables can be treated as units although they are complex entities. FKR *relations* are abstract relations which describe relational properties but are independent of applications to specific sets of units. They are usually denoted as binary sequences or matrices or as graphs. For example, unlabeled graphs or FCA abstract scales, which are contexts or lattices with generic objects and attributes, can be

FKR relations. Higher n-ary relations or conceptual relations, such as "is part of" or "eat(John, apple)" are not modeled as relations but as facets.

Facets are relational structures consisting of finite sets of units, relations and/or other facets (called constituent facets) combined for a certain purpose. Simple examples are an unary relation r with a set N of units as its domain, which form a facet (N, r), or a binary relation r_1 with domain N_1 and codomain N_2, which form a facet (N_1, N_2, r_1). If a facet contains several relations or further sets of units, these are separated by semicolons such as $(N_1; N_2, N_3, r_1; N_3, N_4, r_2)$. More complex examples can be relational database tables, FCA contexts or concept lattices. Within a facet, relations must be concrete which means that they have a domain and codomain of units within the facet. The *purpose* of a facet can be denoted as a list of conditions, some of which are mathematical, such as a requirement for a relation to be transitive, others are meta-mathematical and refer to applications, semantics and pragmatics. The notion of purpose is weaker than the traditional notion of an "interpretation", which maps concepts or relations onto sets of units or tuples of units of a domain, because users can have a purpose for a facet in mind without thinking of specific sets or domains. In FKR, an *interpretation* is any mapping between facets.

The notion of facet and *representation* of a facet are not always distinguished in this paper because facets must be represented to be communicated. A single facet can be represented in different ways, for example, as graphs, sets of tuples, or logical formulas. A representation of a unit, relation or facet is *disambiguated* if it refers to exactly one unit, relation or facet, respectively, for a given purpose. In this paper all representations are assumed to be disambiguated. Some representations may not include all information that is required by the purpose of a facet in an explicit manner. *Full representations* explicitly contain the complete sets of units and relations of a facet required for a given purpose. For example, FCA contexts are not full representations because the set of concepts which is important for the purpose of FCA, is only implicitly contained. As another example, relational database tables are fully represented if all information about the table content and metadata, such as field names and datatypes, is provided. Without the metadata information all empty result sets would be full representations of a unique empty table. Two facets are *equivalent* if they serve the same purpose and a structure-preserving interpretation exists that maps a full representation of the first facet onto the second facet, and vice versa. The meaning of "structure-preserving" depends on the purpose of the facets. For example, structure-preserving means something different for FCA lattices and relational databases. Facets are *identical* if they are equivalent and their units, relations and constituent facets can be mapped via identity mappings. From this definition of facet identity follows that identical facets have identical interpretations.

The semantics of facets is two-sided involving extensional and intensional interpretations and representations. Extensional representations contain only sets of units and of tuples of units. For example, relational database queries return sets of tuples or - in FKR terminology - extensional representations. Intensional representations contain only logical formulas and expressions. For example, a FCA implication basis is an intensional representation of a concept lattice. In applications, most representations are a mixture of extensional and intensional representations. Extensional and intensional

interpretations map facets onto corresponding extensional and intensional representations. Denotative interpretations map facets onto sets, relations or logical structures of an external domain. Extensional denotative interpretations correspond to interpretations in traditional semantics.

In addition to the condition that facets are relational structures, further characteristics of facets are usually required but these depend on the purpose. Single facets are usually expected to be *exhaustive*, *regular* and *appropriate*. Exhaustivity means that a list of values is as complete as necessary for a given purpose. For example, if expected values of salary ranges for a certain application are "$10,000 - $20,000" and "$40,000 - $50,000", then the other ranges, such as "$20,000 - $30,000", should also be included. Furthermore, there should be a range for the highest possible values, such as "larger than $150,000", and the smallest possible values. In this example, the values are regular if the ranges are of equal size and appropriate if no impossible values, such as "$\$\sqrt{2}$", are included.

Complex facets can be constructed from simpler facets. Operations on or constructions of facets cannot be described in mathematical terms without referring to specific types of representations and purposes. For example, mathematical operations on FCA contexts, such as union or direct product, do not result in the same operations on lattices. On the other hand, on an abstract level these operations can be characterized. In facet *aggregation* the structures of several facets with compatible sets, relations and purpose are combined in a union. *Composition* can be applied to facets that share some sets but have different relations on these sets. The resulting composed facet is a direct product of the original facets. *Restriction* refers to the complement of aggregation and *factorization* to the complement of composition. FCA examples for restriction are sublattices; for factorization are factor lattices or homomorphic images; and for composition are nested line-diagrams.

Facets can be arranged in two hierarchies: first, a hierarchy of constituent facets that results from the relation "is used for constructing" among facets. To avoid circularity, such as defining facet A in terms of facet B and B in terms of A, the relation "is used for constructing" should be acyclic. A second hierarchy is a generic hierarchy or hierarchy of specialization/generalization. A facet A is a specialization of a facet B, if the units and relations of A are specializations of the ones of B and the conditions in the purpose of A contain the conditions in the purpose of B. This requires that some other facets exist that identify generic hierarchies for units and relations. In addition to the feature of specialization, *filtering* or zooming can be achieved either via aggregation/restriction, if the level of complexity is changed, or composition/factorization, if the number of aspects or viewpoints is changed. Facets can be constructed theory-driven (intensional) or data-driven (extensional). Theory-driven and data-driven facets can be combined by aggregation, such as in FCA context composition, or by composition, such as in FCA conceptual scaling.

To ensure that the facet construction process is normalized, it is required that facets that are to be combined are independent of each other. Independence means that changing one facet does not affect the other facets. This is usually easy to achieve if all representations are disambiguated, which means that a single name or symbol is not used with different meanings in different facets, and the facet construction mechanisms

are in agreement with the purposes of the facets. Even if simple facets are exhaustive and regular, combined facets can together be incomplete and contain contradictory information. For example, if facets represent opinions of people, it can be expected that their combination yields contradictory information. This is part of the power of using facets: secure and complete information within each facet contrasts uncertain and complex information between facets.

An advantage of this abstract description of facets is that many systems and theories, such as formal concept analysis, object-oriented design or relational databases, can be interpreted as instantiations of FKR. Although formally describing these systems within FKR and identifying the interpretations from one system to the other may be a difficult task, for specific applications this can be achieved. For example, FCA concept lattices can be combined with geographical maps [9]. A second advantage is that FKR provides a two-sided semantics which facilitates shifting between extensional and intensional representations. This feature is already present in FCA. For example, when working with the FCA software tools users can switch between context and lattice. During an attribute exploration, implications (intensional) can be accepted or rejected. To reject an implication, a counter example (extensional) must be provided. Another example of combining extensional and intensional reasoning is Barwise & Etchemendy's Hyperproof [1]. Hyperproof distinguishes "diagrammatical" and "sentential" reasoning which correspond to extensional and intensional representations in FKR. The term "diagrammatical" can be misleading because there are also forms of intensional diagrammatical reasoning, such as Peirce's rules of inference. We believe, it would be helpful for many applications if a two-sided semantics is explicitly stated so that the advantage of being able to switch between extensions and logical formulas can be fully explored.

3 The FaIR System

The Faceted Information Retrieval (FaIR) system described in this section is an instantiation of FKR. A prototype of FaIR has been implemented as an interface for a small knowledge base (KB) of computing terms, which is developed by the information technology support center of Indiana University (http://kb.indiana.edu/). FaIR is somewhat similar to the FCA tool TOSCANA but while TOSCANA nests scales (or facets), FaIR presents them side by side. FaIR consists of a document database and a faceted thesaurus from which a document description language and a query language are derived.

There are two main types of representation for faceted thesauri: a representation as a term hierarchy and a representation as a concept hierarchy. Figure 1 shows a graphical representation of concept hierarchies of two thesaurus facets. Both are lattices but in this figure and the following figures, the bottom nodes of the lattices are always omitted. Figure 2 contains a term-based representation of the composition of the facets from figure 1. The representation in figure 2 corresponds to the ISO standard for thesauri [6] except that the NT (narrower term) symbol is omitted.

In the term-based representation, a thesaurus baseline facet $(T, T, r_T; t_t)$ is defined as a set T of units called "terms", a relation r_T among the terms called "generic relation", and a top term t_t. The conditions are that r_T is reflexive, antisymmetric and transitive; the top term is in T ($t_t \in T$) and broader than any other term in the

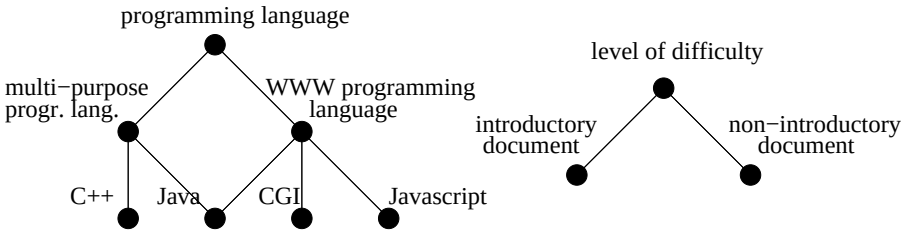

Fig. 1. Two thesaurus facets

facet ($\forall_{t \in T} : tr_T t_t$). In the concept-based representation, a thesaurus baseline facet $(C, C, r_C; c_t)$ is defined as a set C of units called "concepts" which form a lattice. The ordering relation r_C is called "subconcept-superconcept" relation and c_t is the top concept of the lattice. The term-based and concept-based representations are related via an interpretation that embeds the term hierarchy into the smallest possible lattice. In mathematical terms that means that the Dedekind closure is calculated for the term hierarchy. Other representations of thesaurus facets, such as as FCA concept lattices with lists of prototypical objects and attributes, are possible.

In the FaIR system, a faceted thesaurus consists of thesaurus baseline facets that are combined in facet composition. Composition of baseline facets yields the direct product of the concepts from the different baseline facets. Every concept in the left facet of figure 1 can be composed with any concept in the right facet, such as "introductory document C++". Concepts that are within one facet cannot be composed. As indicated above, facet construction mechanisms are not necessarily visible if the representations are not full representations. Although figures 1 and 2 represent facet composition this is not directly visible in the representations. In the term-based representation, facet composition corresponds to a union of the hierarchies of the original facets. The top terms of the original facets are surrounded by angle brackets and preceeded by the word "by" (compare figure 2). In the concept-based representation, the direct product can be explicitly represented or as a nested-line diagram, such as done by the FCA software TOSCANA. In FaIR the facets are represented side by side. Users explore the facet composition interactivly by selecting concepts from different facets and observing the impact which that has on the set of retrieved documents (see below). For further details on facet composition in a faceted thesaurus see [10].

For the assignment of documents it is assumed that documents are represented in a document facet (D, K, r_{DK}) that consists of a set D of documents, a set K of keywords and a relation r_{DK} among them. A second facet (K, C, r_{KC}) maps the keywords to concepts. It consists of a set K of keywords of documents, a set C of concepts of a faceted thesaurus and a relation r_{KC} among them. A condition is that r_{KC} is a mapping. Since documents can contain several keywords, it is not necessarily obvious how the two facets (D, K, r_{DK}) and (K, C, r_{KC}) can be combined. Several strategies have been suggested in the literature. The strategy that FaIR uses identifies at most one concept in each baseline facet to which the document is assigned. This concept is the most general concept that encompasses the concepts to which the keywords of the document correspond. In case a document has general and specific keywords assigned, such as

```
KB document
        <by level of difficulty>
                introductory document
                non-introductory
        <by programming language>
                multi-purpose programming language
                        C++
                        Java
                WWW programming language
                        Java
                        CGI
                        Javascript
```

Fig. 2. The facets from figure 1 as term hierarchy

"dog" and "poodle", the general keywords are ignored. This is achieved by first computing the minimum above the bottom concept of the kewyord concepts in the lattice and then calculating its supremum in the lattice. Formally, the construction is described as follows. The conventions are $d^r := \{c \mid drc, d \in D, c \in C\}$ for $r \subseteq D \times C$; the minimum of a set of concepts in a lattice after exclusion of the bottom element is denoted by $min(C_1)$; and $\circ$ denotes the relational composition. A facet (D, C_f, r_f) of documents and concepts is computed for every baseline facet f in the faceted thesaurus with relation r_f defined as $dr_f c :\Longleftrightarrow c = \bigvee(min(d^{r_{DK} \circ r_{KC}}))$. For non-baseline facets, a facet (D, C, r_{DC}) is computed with relation r_{DC} defined as $dr_{DC}c :\Longleftrightarrow c = (c_1, \ldots, c_n)$ and $dr_{f1}c_1, \ldots, dr_{fn}c_n$.

Figure 3 shows the assignments of documents to the composition of three facets from the KB. The first two are identical to the ones in figures 1 and 2. The third facet contains information on the relevance of the documents to the user communities: everywhere, only at IU or only at certain IU campuses (IUB, IUK, etc). The numbers under the nodes represent document counts. To be counted, every document must have a keyword that is mapped to a concept in each of the three facets. There is a total of 110 documents that fulfill this condition. Users can click on the document counts and retrieve further information on the documents. Or they can click on concepts from different facets to restrict the retrieved document set. Figure 4 shows an example where a user clicks on "introductory document" and thus restricts the retrieval set to introductory documents about programming languages related to any user community. In contrast to many traditional information retrieval interfaces FaIR's graphical display provides hints for the user on how to broaden a search in case of small retrieval sets and how to narrow a search in case of large retrieval sets.

Formally, the selection of facets, which is done via a menu display, and the selection of concept sets within each facet, which is done by mouse click on the concept nodes, correspond to a query language. The query language consists of the set C of concepts of the faceted thesaurus and the operators "AND", "OR" and "NOT". While "AND" can be applied to concepts from a single facet or concepts from different facets, "OR" and "NOT" can be applied only to concepts within single facets. This is not a real limitation

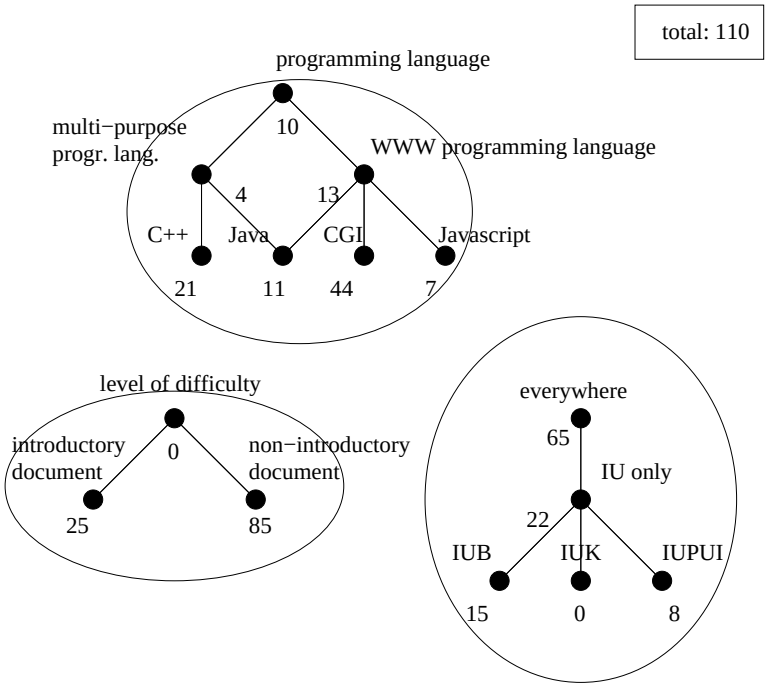

Fig. 3. Thesaurus facets with assigned documents

because users normally do not formulate queries such as “programming language OR introductory document”. Within each facet, concepts and their assigned document sets can be selected in an exclusive or inclusive mode. Exclusively, only the documents that are directly assigned to a concept are relevant to the concept. Inclusively, the filter and ideal of a concept in the lattice yield further relevant documents for a concept. According to the strategy used for the assignment of documents to concepts, documents that cover multiple topics within one facet are assigned to high level concepts. For example in the first facet in figure 3, a document that covers Java and C++ is assigned to multi-purpose programming language. All documents relating to Java can thus only be retrieved if the inclusive mode is selected. Figure 5 shows several examples of queries in that facet. The circles and dotted lines are only included because of the black and white print in this paper. They correspond to highlighting in the interface.

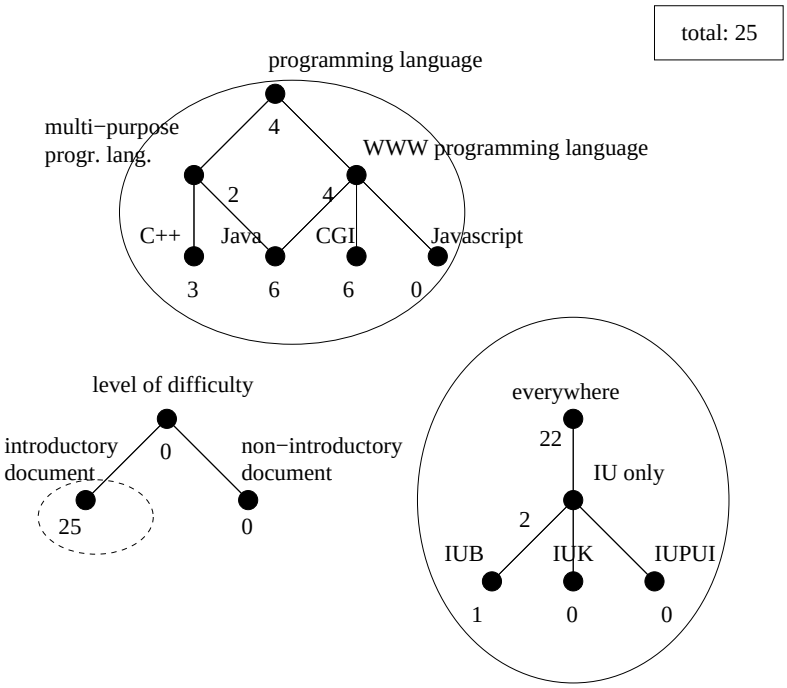

Fig. 4. Restricting the result set of documents

There are several possible choices concerning how intra-facet “AND”, “OR” and “NOT” can be implemented. In FaIR’s current implementation, exclusive “AND” yields the join and meet of the concepts in the lattice; inclusive “AND” yields the filter and ideal above (below) the join and meet, respectively; exclusive “OR” results in the union of the intervals between the selected concepts and their join and meet; inclusive “OR” results in the union of filters and ideals above and below all selected concepts; exclusive and inclusive “NOT” are implemented as set-theoretical subtractions of the correspond-

ing sets. Inter-facet “AND” is represented in the composition of facets by highlighting the selected concepts in the single facets and restricting the selected documents to these. In the query language the queries in figures 3 and 4 are represented as “’programming language’ (incl) AND ’everywhere’ (incl) AND ’level of difficulty’ (incl)” and “’programming language’ (incl) AND ’everywhere’ (incl) AND ’introductory document’ (excl)”, respectively.

The FaIR interface has been implemented for a small subset of the documents and has been tested in a usability study. Several issues still need to be addressed and improved but they all seem to relate to the graphical design, to the selection of terms, and to other implementation issues. FaIR represents a controlled vocabulary interface because users cannot select their own terms but are restricted to the terms provided by the system. A closer comparison with other controlled vocabulary interfaces and with the FCA software TOSCANA should be interesting and may be conducted in the future.

The goal of this paper is not to discuss another information retrieval system but instead to show how FaIR instantiates faceted knowledge representation. So far, only some FKR features are implemented: facet aggregation is implicit in the construction of thesaurus baseline facets, facet composition is achieved interactively via the selection of facets by the user. Next steps will be to implement filtering by allowing users to select the vocabulary and thus the complexity of the facets depending on their level of expertise and to incorporate facets that are not lattices, such as flow diagrams or maps.

References

1. Barwise, John; Etchemendy, John: Hyperproof. CSLI, Stanford, CA, 1994.
2. Euzenat, J.: HyTropes: a WWW front-end to an object knowledge management system, Actes 10th knowledge acquisition workshop demonstration track, Banff (CA), 1996.
3. Ganter, Bernhard; Wille, Rudolf: Formal Concept Analysis. Mathematical Foundations. Berlin-Heidelberg-New York: Springer, 1999.
4. Gardner, Martin: Logic Machines and Diagrams. McGraw-Hill, New York, 1958.
5. Guttman, Louis: An Outline of Some New Methodology for Social Research., Public Opinion Quarterly, 1954.
6. ISO 2788-1986. International Standard 2788: Documentation -Guidelines for the establishment and development of monolingual thesauri. International Organization for Standardization, 1986.
7. Pollitt, Steven: Interactive Information Retrieval based on Faceted Classification using Views. Proc. of the 6th Int. Study Conf. on Class., London, June 1997, FID, 51-56.
8. Prieto-Diaz, R.: Implementing Faceted Classification for Software Reuse. Communications of the ACM, 34(5), 1991, p. 88-97.
9. Priss, Uta; Old, John: Information Access through Conceptual Structures and GIS. In: Information Access in the Global Information Economy. Proceedings of the 61st Annual Meeting of ASIS, 1998, p. 91-99.
10. Priss, Uta; Jacob, Elin: Utilizing Faceted Structures for Information Systems Design. Proceedings of the 62st Annual Meeting of ASIS, 1999.
11. Ranganathan, S. R.: Elements of library classification. Asia Publishing House, Bombay, 1962.

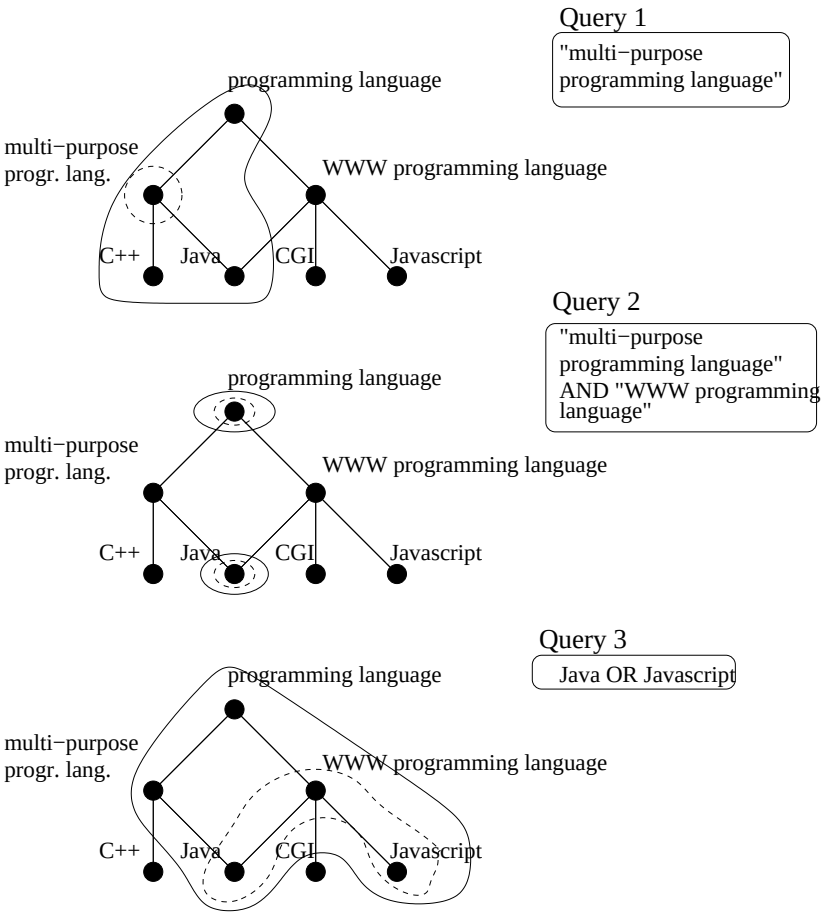

Fig. 5. Several queries in a single facet