Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 177 (2007) 4566–4597
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

On the quality of resources on the Web: An information retrieval perspective

B. van Gils *, H.A. Erik Proper, P. van Bommel, Th.P. van der Weide

Radboud University Nijmegen, Institute for Computing and Information Sciences, Tournooiveld 1, Nijmegen, Netherlands

Received 9 June 2006; received in revised form 20 April 2007; accepted 6 May 2007

Abstract

We use information from the Web for performing our daily tasks more and more often. Locating the right resources that help us in doing so is a daunting task, especially with the present rate of growth of the Web as well as the many different kinds of resources available. The tasks of search engines is to assist us in finding those resources that are apt for our given tasks. In this paper we propose to use the notion of *quality* as a metric for estimating the aptness of online resources for individual searchers.

The formal model for quality as presented in this paper is firmly grounded in literature. It is based on the observations that objects (dubbed artefacts in our work) can play different roles (i.e., perform different functions). An artefact can be of high quality in one role but of poor quality in another. Even more, the notion of quality is highly personal.

Our quality-computations for estimating the aptness of resources for searches uses the notion of linguistic variables from the field of fuzzy logic. After presenting our model for quality we also show how manipulation of online resources by means of transformations can influence the quality of these resources.

Keywords: Market models; Information retrieval; Quality on the Web

1. Introduction

The amount of information available to us has been increasing at an explosive rate over the last few years, especially with the enormous growth of the Web. Several tools and system have been developed to help us in dealing with this vast amount of *resources* such as indexes, search engines, catalogs and so on. The traditional information retrieval (IR) paradigm is introduced in Fig. 1. In this paradigm the main challenges are [41]:

Formulating needs – The formulation of information requests involves two important issues. First of all, it requires some formal language in which to express the query. Secondly, a precise formulation of the *true* information need is required. Obtaining such a formulation has proven to be a non trivial task [14].

* Corresponding author.
E-mail address: bas@van-gils.org (B. van Gils).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2007.05.035

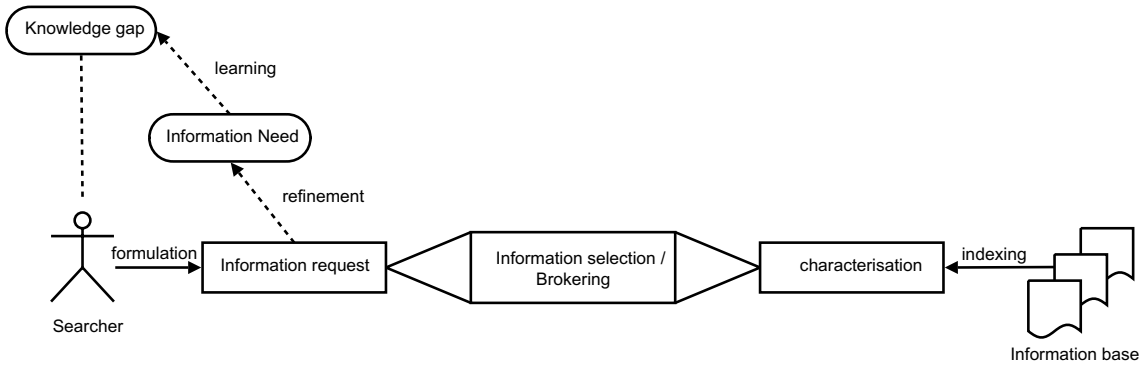

Fig. 1. The information retrieval paradigm.

Characterizing supply – Good characterization of information resources is imperative for effective information discovery, as poor characterizations inevitably leads to the retrieval of irrelevant information, or the missing of relevant information. An important question is of course which properties to include in a characterization. A useful property to include seems to be what an information resource is *about*. In addition, properties like authorship, price, medium, etc. may be included. In the literature standard attribute sets to characterize resources can be found in the context of meta-data standardization efforts [6,49].
Matching demand and supply – The selection of relevant information resources for a given query is a well understood problem. The field of information retrieval has developed a number of retrieval models.

In the past, our research group has studied several aspects of these challenges (formulating needs: [12,10,13,32,43], characterizing supply: [45,27,28], matching demand and supply: [2–4]).

The notion of *quality* is particularly important in this area as it is a driving force in the information market [26]. Relevant questions would be: *What is the quality of the characterization of resource space*? *What qualities do resources have*? *What is the quality of a query*? *How well is it formulated and how accurately does it describe the searchers information need*? *What is the quality of a search engine/match maker*? *What are its qualities*?

Since it is our ultimate ambition to explicitly reason about the matching of demand and supply in terms of offered and desired qualities, quality properties have to be made specific and precise in order to be able to reason about then. We will therefore provide a more formal elaboration of the notion of quality in this paper. More specifically, when an actor assesses the quality of an artifact then this assessment is based on (some of) the qualities that the artifact has. Which qualities play a role in a quality assessment depends on the (current) goals of the actor, his mental state etcetera. As such they are often implicit and hard to measure. Even more so, the quality assessment of an actor may vary over time as his goals or context changes!

It is often also difficult to (automatically) measure which properties an artifact has, or which values it has for a property. For example, different people may classify the *color* of an artifact differently (red versus orange, blue versus green). It seems impossible to even *express* quality in the sense of desirability. It does not make sense to state something like: ''The quality of this artifact is 10''. Quality of an artifact only makes sense in comparison with other (similar) artifacts. As such quality provides an ordering. Observe, however, that we (humans) may associate a judgment (reasonable quality, poor quality) to this comparison.

Given the above brief analysis, we feel that there are three aspects (or three 'layers', if you will) in assessing quality in the context of the information market:

1. **Measurement**: measuring the qualities that artifacts have is the first step. As we have observed already, there may be a great deal of uncertainty involved in these measurements.
2. **Calculus**: in order to be able to deal with (the uncertainty of) measurements a well-defined calculus must be developed specifically tailored for quality of resources on the Web.
3. **Ranking**: brings us back to the retrieval problem; somehow it must be possible to rank (topically relevant) resources according to their quality for a specific searcher in a specific context with specific goals.

In this article we will examine the notion of quality in a Web context. More specifically:

The goal of this article is to explore the notion of quality in the context of the Web; to explain what it is and how it can be used in practice.

In analogy with the famous OSI Protocol Stack, the following Web Services Protocol Stack has been designed:

1. Layer 1 technology handles the physical exchange of data.
2. Layer 2 takes care of reliably transmitting data, and consists of two sub-layers, the lower sub-layer being the media access control (MAC), and the higher sub-layer being the logical link control (LLC).
3. Layer 3 is the addressing and routing layer, within the Internet protocol stack implemented by the internet protocol (IP).
4. Layer 4 handles security and the way message patterns are supported.
5. Layer 5, the coordination layer, provides some transaction control.
6. Layer 6 is called the vocabulary layer, and is responsible for mapping the application's data model into a form than can be transmitted between communicating peers.

Quality aspects basically are introduced at the vocabulary layer, but may also involve aspects from lower layers, for example, reliability information (level 2) may be an attribute taken into account. There are also other potential application areas for the ideas developed in this paper, for example Virtual reality and scenedescription languages.

When considering the limitations of our work, it is important to realize that our treatment of quality is not particularly suitable for *data quality based on data semantics*. In case our approach is to be used in that context, the chosen model of data semantics has to be embedded in terms of quality functions.

In the remainder of this paper, we will first provide a brief discussion of the notion of quality. This is followed by the introduction of a formal model for quality in Section 3. This model aims to combine the two views on quality (properties and desirability). In Section 4 we will make this high-level model for quality more specific for the resources on the Web. That is, we will introduce a set of concepts with which we can model/represent the qualities of resources on the Web. We will use the same set of qualities to also introduce a language with which those properties that are used in a quality assessment can be expressed. The models presented in this section will assume that there is no uncertainty about the property assignments and the quality assessments. Uncertainty will be added to these models in Section 5. Last but not least, in Section 6 we will show how our findings can be operationalized on the Web by means of transformations.

2. Quality

In this section we study the notion of quality based on literature from several fields. A thorough investigation of how this term is used in literature seems particularly useful as [21] points out that "Well-founded and practical approaches to assess or even guarantee a required degree of the quality of data are still missing". To fuel the discussion we start with a definition from the *Webster's third new international dictionary, unabridged* (1981). The noteworthy headings in the entry are:

> Peculiar and essential character; a distinct, inherent feature; degree of excellence; inherent or intrinsic excellence of character or type social status; a special or distinguishing attribute the character in a logical proposition of being affirmative or negative something that serves to identify a subject of perception or thought in respect in which it is considered something from the possession of which a thing is such as it is manner of action.

From this definition we can derive that in essence two main interpretations of quality exist. The first aspect refers to the fact that quality can be considered synonymous for the word "attribution". In terms of the dictionary definition: an artifact may possess a certain inherent feature. The second aspect to the notion of quality refers to the fact that the notion of quality is used to express how "good" some artifact

is. Note that this is both personal and dependent on the present situation of an actor. As such the term quality is used to refer to the desirability of properties or characteristics of some artifact. We refer to the former interpretation as "quality as in attribution" and the latter interpretation as "quality as in desirability".

It is interesting to observe that the attribution-interpretation of the quality notion has been studied since the ancient philosophers. For example, in his work on *the Philosophy of Nature* Aristotle defined the word quality as the category according to which artifacts are said to be like or unlike (see e.g., [34]). Other great philosophers such as Descartes, Bacon, Newton, and Galileo opposed to Aristotle's view on (the quality of) matter (see e.g., [19]). In their view a distinction must be made between the artifactive qualities of matter and its largely subjective qualities. This observation must also be reflected in our theory for quality: it is essential to note that only some qualities of artifacts can be measured objectively. This influences the determination (and hopefully: computation) of the quality in the sense of desirability of the artifact under consideration. In the context of Web resources this means that we must take into account that not every quality (property) can be measured objectively. As such we must cater for a "situational" view of quality.

A second interesting issue with respect to the notion of quality is *uncertainty*. In [48] the problem of *quality uncertainty* is discussed. This problem boils down to the observation that in E-Commerce (loosely defined as doing business via the Web) customers often have difficulty accepting products or services from 'strange vendors' that may not even have a bricks and mortar back office. Two methods to deal with this problem are mentioned: *provide free samples* and *return if not satisfied*. The former, however, is difficult in case of digital products since they are consumed when they are viewed by customers.

Finally there are some other issues with respect to the notion of quality that should be mentioned here without further elaboration:

- High quality *process* does not necessarily imply that high quality *artifacts* are produced by this process [37]. This is also stressed by [42] where a distinction is made between the teleological point of view and the causal point of view with respect to quality.
- The perceived quality of an artifact (i.e., quality as desirability) can be dependent on different factors. In some fields, such as operations research, attempts have been made to standardize these factors: product attributes, product performance, service characteristics, warranty, service availability, and total price [30]. Total Quality Management (TQM), then, is a concept that makes quality the responsibility of all people within an organization [38].
- In many fields, such as software engineering, quality is defined as "conformance to specification". See e.g., [46,18,20]. Three related principles are:
  The principle of unambiguous quality specification: all quality requirements can and should be stated unambiguously.
  Kelvin's principle: when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind.
  Shewhart's measurable quality principle: The difficulty in defining quality is to translate future needs into measurable characteristics so that a product can be designed and turned out to give satisfaction at a price the user will pay.
- Particularly in the field of Web information, many authors have tried to capture *principles* and *guidelines* which should help achieve a certain level of quality (desirability) of Web data. In e.g., [40] this list is comprised of the following: Unused data cannot remain correct for very long; Data quality in an information system is a function of its use, not its collection; Data quality will be no better than its most stringent use; Data quality problems tend to become worse as the information system ages; The less likely some attribute (element) is to change, the more traumatic it will be when it finally does change; and Laws of data quality apply equally to data and meta-data.

Even though the above list of issues are interesting in their own right, they do not influence our model of quality as presented in the next Section.

3. A model for quality

Upon closer examination, the above definitions and applications of quality show that there are two main views on quality:

Property: the 'qualities of something'. At some level of abstraction this view on quality can be considered objective. However, deciding whether something has a property or not can also lead to philosophical discussions. It remains to be seen if an 'objective reality' exists or not.
Desirability: has to do with 'how good' something is (in comparison to other things). This is a subjective view on quality.

It would be desirable to be able to make quality SMART (Simple, Measurable, Applicable, Repeatable, and Trainable) and to unify/use both views on quality. The properties an artifact may exhibit relate to the supply side of the information market, while the desirability can be positioned at the demand side.

3.1. Quality and properties

As stated previously, the main goal of this section is to introduce a (formal) model for quality. This requires a two-pronged approach. Firstly, the intuition behind our model has to presented. We will use motivating examples for this. Secondly, we will present a formalism. Fig. 2 shows our model using the Object Role Modeling (ORM) notation,[1] which provides the signature for the formalism. In the remainder of this section we will use the terminology introduced in this figure. for an overview of this notation. The first observation that we must make is that the artifacts can play different roles. For example, a mug can be seen as a device from which you can drink tea; it can be seen as an art object or even as a place to store pens in. The quality of some artifact depends on which role this artifact plays. Continuing the above example: a mug can be great as a drinking device but be horrible as an art object. We will model this as follows: Let $\mathscr{AF}$ be the set of all artifacts that may have certain qualities (properties) and let $\mathscr{RO}$ be the set of all roles that these artifacts can fulfill. The combination of an artifact and a role is dubbed an *fulfillment* (i.e., a fulfillment denotes an artifact in a role): $\mathscr{FL}$. The artifacts and roles that participate in a fulfillment can be found using the functions $\mathsf{Artifact} : \mathscr{FL} \to \mathscr{AF}$ and $\mathsf{Role} : \mathscr{FL} \to \mathscr{RO}$ respectively. Since a fulfillment denotes an artifact in a role we know that an artifact and a role combination uniquely determines a fulfillment:

Axiom 1 (*Unique fulfillment*)

$$\mathsf{Artifact}(e_1) = \mathsf{Artifact}(e_2) \wedge \mathsf{Role}(e_1) = \mathsf{Role}(e_2) \Rightarrow e_1 = e_2$$

For convenience of notation we introduce the following abbreviation for a fulfillment;

$$\langle a, r\rangle \triangleq e \text{ such that } \mathsf{Artifact}(e) = a \wedge \mathsf{Role}(e) = r$$

This allows us to write $\langle \text{MyMug}, \text{drinking device}\rangle$ for a specific fulfillment. The following example illustrates the use of artifacts, roles and fulfillments in our model.

Example 3.1. Let Mug (denoted by a) be an artifact that can play two roles. It either plays the role of type: something to drink from (denoted by r_1) or the role of type: art object (denoted by r_2). Both $e_1 = \langle a, r_1\rangle$ and $e_2 = \langle a, r_2\rangle$ are entities such that:

$$\mathsf{Artifact}(e_1) = a \quad \mathsf{Role}(e_1) = r_1$$
$$\mathsf{Artifact}(e_2) = a \quad \mathsf{Role}(e_2) = r_2$$

Recall that the quality (desirability) of an artifact depends on its qualities (properties). Furthermore, observe that properties should not be coupled to artifacts as such, but to the roles that these artifacts play. To see why this is the case one only needs to realize that, for example, all mugs have a volume; that all vehicles have a maximum speed; that all storage devices have a capacity etcetera. Furthermore, properties such as speed,

[1] See e.g. [29] for an overview of this notation.

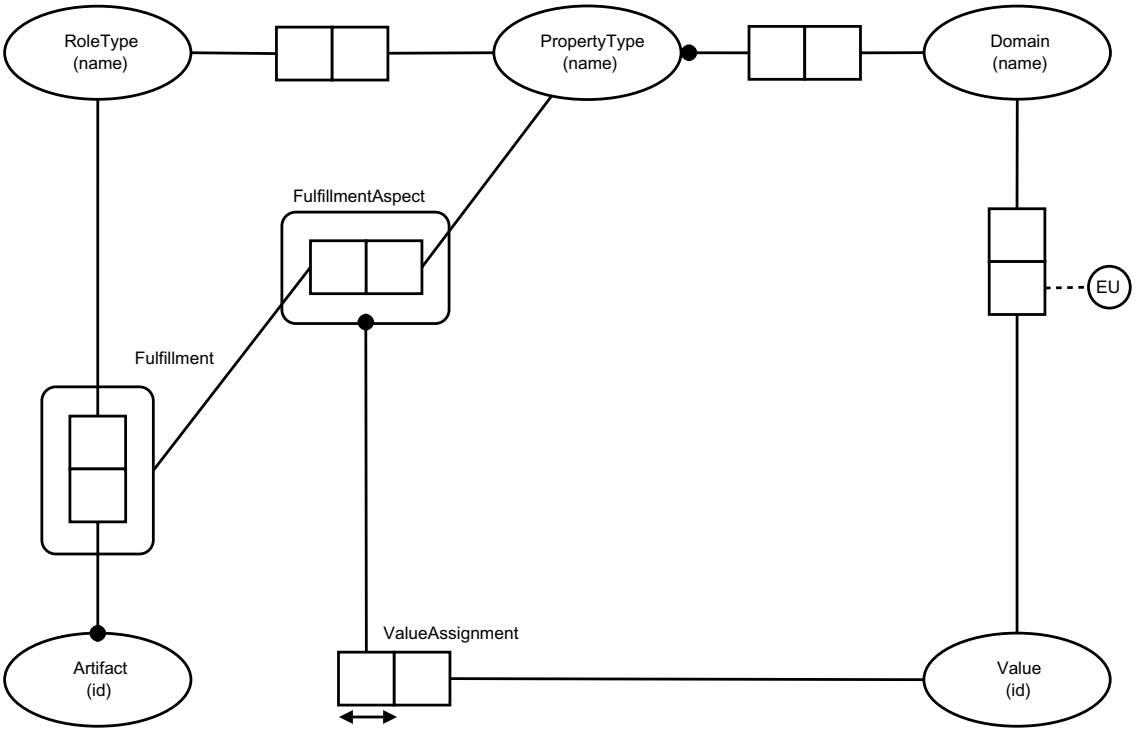

Fig. 2. Properties of artifacts.

capacity can be expressed in different domains. For example, consider the property type color. This can be expressed in the domain *RGB color* but also as *CMYK color*.

We model this as follows: Role types can have properties, the value of which are expressed in a property domain. Let $\mathscr{PT}$ be the set of property types and $\mathscr{PD}$ be the set of all property domains. The properties that can be played by a certain role type are given by the relation $\mathsf{Props} \subseteq \mathscr{RO} \times \mathscr{PT}$ and the domain in which (values of) a property can be expressed is given by the function $\mathsf{PrDom} \subseteq \mathscr{PT} \times \mathscr{PD}$. We continue the above mentioned example to illustrate the use of our model further.

Example 3.2. Role type *art object* (r_2) can have the property type *color* (denoted by p) which can be expressed in the domain *RGB-colors* (denoted by d_1) and the domain *CMYK-colors* (denoted by d_2) such that: $\mathsf{Props}(r_2) = \{p\}$ and $\mathsf{PrDom}(p) = \{d_1, d_2\}$.

Note that property types and domains are at the typing level. We still need to assign values to entities having a certain property type. The first step to achieve this is to create a link between $\mathscr{PD}$ and the values from this domain. The set $\mathscr{VL}$ consists of sets of values for a certain domain. In other words, an element from $\mathscr{PD}$ is the *names* of a certain domain and an element of $\mathscr{VL}$ consists of its values. In the ORM-schema (Fig. 2) the extensional uniqueness constraint denotes the fact that the values uniquely determine the domain(name). The functions $\mathsf{Value} : \mathscr{PD} \to \mathscr{VL}$ and $\mathsf{VlDom} : \mathscr{VL} \to \mathscr{PD}$ are used to find the values of a domain or the name of a set of values respectively. For example:

Example 3.3. The domain RGB-colors (d) has the values $v = \{\#000000 \ldots \#FFFFFF\}$. More specifically: $\mathsf{Value}(d) = v$ and $\mathsf{VlDom}(v) = d$.

Last but not least we should introduce notation for expressing the fact that a fulfillment has an associated value for a certain property. For example, we should be able to express that a mug has a volume of 20 cm^3. The property type of a fulfillment is denoted in our model by a *fulfillment aspect*. The set of these fulfillment aspects is denoted by $\mathscr{FA} \triangleq \mathscr{FL} \times \mathscr{PT}$ such that

$$\langle f, p \rangle \in \mathscr{FA} \Rightarrow p \in \mathsf{Props}(\mathsf{Role}(f))$$

The intended meaning is as follows:

Example 3.4. Let $f = \langle \text{mug,drinking device} \rangle$ denote the fulfillment of a mug in its role as drinking device and let color $\in \mathscr{PT}$ be a property type. Then $\langle f, \text{color} \rangle$ is a fulfillment aspect denoting the color of mugs in their role as drinking device.

This notion of fulfillment aspects may seem somewhat unnatural. We introduce this concept here mainly to make the remainder of our formalisation more elegant. In our model we will use the predicate $\mathsf{ValAss} : \mathscr{FL} \to \mathscr{VL}$ to denote the observation that a fulfillment has a certain value for a property type. Continuing our example:

Example 3.5. The fact that the mug (a) as an art object ($r2$) has the color (p) red (#*FF*0000) is expressed as: $\mathsf{ValAss}(\langle a, r_2 \rangle, p) = \#FF0000$.

In our model we have to ensure that the observations on the instance level do not conflict with the typing level, something that is 'obvious' in the real world. For example, if a fulfillment is said to have a value assignment for a property then, obviously, one of the roles of this fulfillment must at least have this property. Similarly, consider the observation: $\mathsf{ValAss}(\langle \text{mug}, \text{drinking device} \rangle) = 20\,\text{cm}^3$. To be able to make this observation, the value $20\,\text{cm}^3$ must be in $\mathscr{VL}$ and it must be of the correct domain. That is, it must be of the domain in which the property type can be expressed. The following axiom enforces that the typing level and instance level stay in sync. Let f be a fulfillment, p a property type and v a value:

Axiom 2 (*Conformance*)

$$\mathsf{ValAss}(f, p) = v \Rightarrow p \in \mathsf{Props}(\mathsf{Role}(f)) \wedge \mathsf{PrDom}(p) = \mathsf{VlDom}(v)$$

In order to be able to operationalize this model for quality properties, a measuring method has to be developed for:

- Measuring the roles that an artifact can play;
- Measuring the property types that exists;
- Measuring the value assignment of a fulfillment.

Devisings such measuring methods is a problem in itself. In [1]. Ken Alder writes "Our methods of measurement define who we are and what we value". In his book, Alder describes the quest or a universal measure for distance in the late 1790's by two astronomers. Their task was to establish a new measure (the meter) as one ten-millionth of the distance from the North Pole to the equator. This is, obviously, by the standards deployed in these days, as well as by modern standards, a daunting task to say the least.

As this example illustrates: agreement of stakeholders is important. Sufficiently many people involved should agree on the roles that an artifact can play, and the properties that exist etcetera. For example, if two stakeholders can not agree on the color(s) of a mug or the roles that this mug can play: what good will the measuring system be then? Note that, in essence, there are two ways a measuring system will be able to, or forced to, operate when assigning values:

Objective: some value assignments can be measured objectively. For example: the number of characters in a file, or the weight of an artifact,
Subjective: other value assignments are, really, dependent on humans. For example: is an artifact expensive, or is it pretty?

We will return to this issue in the upcoming sections. More specifically, the objective value assignments will be the topic of Section 4. The subjective assignment will be addressed in Section 5 by looking at the uncertainties they introduce in matching objectively measured properties with subjectively formulated/measurable desired properties.

3.2. Quality and desirability

To be able to assess the quality (in the sense of desirability) of an artifact for a user, his/her actual desires must be made explicit. The question is how to do this. One of the main problems is to choose a domain in which quality is expressed. To be more precise, it doesn't seem to make sense to say: "The quality of this artifact is 24". The notion of quality is, in that respect, similar to the notion of *value* as discussed in [8]: it is an abstract notion and can be used to compare artifacts.

Quality, in the sense of desirability, depends on the desires of people (actors). However, these actors are not always aware of their desires, or may not know how to express them. Such issues also arise in other fields such as:

- Software engineering: stakeholders have to, somehow, express requirements with regard to a system. See e.g., [36,46,7].
- Search on the Web: searchers must try to specify their information need. See e.g., [5,27,32].

Furthermore, a distinction must be made between *hard* and *soft* desires with regard to artifacts. These can be compared, to some extent, to functional and non-functional requirements or hard goals and soft goals in requirements engineering (See e.g. [16]). In requirements one often tries to *make soft goals hard*. In our opinion, a goal/requirement is considered to be *soft* if a human opinion is needed for the value assignment. Otherwise, it is considered to be *hard*. In other words, hardness or softness of a requirement depends on the way of measurement. The following are examples of hard goals and soft goals:

Hard goals: Price may not exceed €20. Contents of 25 l. Made of stainless steel.
Soft goals: Cheap. Pretty. Low. Hard. Strong.

Quality in the sense of desirability depends on the *requirements* of an individual. More specifically: these requirements have to do with value assignments; the quality of some fulfillment increases if properties have 'the right value'. Putting it differently, value assignments are *constrained*. Consider the following examples of a requirement for a fulfillment:

Example 3.6

- *The price may not exceed* €10
  In this example, *price* is a property type which is expressed in the domain €'s. Furthermore, 10 is a value and *may not exceed* is a constraint.
- *The price in euros must be as low as possible*
  In this example, *price* is a property type which is expressed in the domain €'s. Furthermore, *must be as low as possible* is a constraint.
- *The price in euros may not exceed the price of cup c*
  In this example, *price* is a property type which is expressed in the domain €'s. Furthermore, *may not exceed the price of cup c* is a constraint involving an assignment.

Observe that the former requirement has a property type, a constraint and a value and the latter requirement does not specify a value. We model this as follows: Let $\mathscr{R}\mathscr{Q}$ be the set of all requirements and $\mathscr{C}\mathscr{S}$ be the set of all constraint operators.[2] A requirement adheres to a property type (mandatory), a constraint (mandatory) and possibly an *expression* (optional).

Expressions can either be values or value assignments, as illustrated by the above examples. In the first example the expression is a, value whereas in the latter example the expression is another value assignment. Traditionally, expressions are often modelled in terms of base expressions (literals) which can be combined

[2] In the following text we will abbreviate "constraint operator" with the simpler, and more readable "constraint".

by operators and possibly some logical connectors. Consider example, the expression $P(x) \wedge Q(x,y)$. This expression has a unary operator P and a binary operator Q. Even more, the expressions are coupled using a logical and. In terms of our model we need only a subset of this full approach. Therefore we model expressions as follows.

In our model: $\mathscr{EX} \triangleq \mathscr{VL} \cup \mathsf{ValAss}$[3] denotes the set of all expressions. Let $\mathsf{Prop} : \mathscr{R2} \to \mathscr{PT}$, $\mathsf{Constr} : \mathscr{R2} \to \mathscr{CS}$, and $\mathsf{Expr} : \mathscr{R2} \rightarrowtail \mathscr{EX}$. We introduce the following shorthand notation:

$$r_1 = \langle p, c, e\rangle \triangleq \mathsf{Prop}(r_1) = p \wedge \mathsf{Constr}(r_1) = c \wedge \mathsf{Expr}(r_1) = e$$
$$r_2 = \langle p, c\rangle \triangleq \mathsf{Prop}(r_2) = p \wedge \mathsf{Constr}(r_2) = c$$

The previous examples can now be written more formally as:

Example 3.7

- *The price may not exceed* €10
  $\langle price, <, €10\rangle$
  Requirement on Property Type ''Price'' by Constraint Operator ''may not exceed'' is Value ''10 euro''
- *The price in euros must be as low as possible*
  $\langle price, \min\rangle$
  Requirement on Property Type ''Price'' is Constraint Operator ''minimize''
- *The price in euros may not exceed the price of cup c*
  Letting g denote the fulfillment of cup c in some role:
  $\langle price, <, \mathsf{ValAss}(g, price)\rangle$
  Requirement on Property Type ''Price'' by Constraint Operator ''may not exceed'' is the Value of Artifact ''c'' with respect to Property Type ''price''

Fig. 3 illustrates how requirements are positioned in our quality-model. Note that a requirement with respect to a fulfillment is of a certain actor/individual. Let $\mathscr{AC}$ be the set of actors and $\mathsf{Req} : \mathscr{AC} \times \mathscr{FL} \to \wp(\mathscr{R2})$ denote the requirements of an actor with regard to a fulfillment. For example:

$$\mathsf{Req}(a, f) = \{r_1, r_2\}$$

denotes the observation that actor a has requirements r_1 and r_2 with regard to fulfillment f.

Last but not least we will point out the relation between quality assessment and choice. To this end, consider the following example situation in which you want to buy a mug (in its role of a 'drinking device'):

Example 3.8. The decision space is summarized by:

	Property type		
	Color	Volume (cm^3)	Price
m_1	Red	20	€3
m_2	Red	25	€3
m_3	Blue	25	€2

Depending which mug is best (i.e. has highest quality for an actor a) depends on the requirements of the actor. Let f denote the fulfillment of a mug artifact in its role as a drinking device and $\mathsf{Req}(a, f) = \{r_1, r_2, r_3\}$ where $r_2 = \langle \text{color}, =, \text{red}\rangle$, $r_3 = \langle \text{volume}, \geqslant, 25\ \text{cm}^3\rangle$ and $r_1 = \langle \text{price}, \leqslant, \text{e3}\rangle$. In this case, it seems apparent that m_1 not feasible: for this actor it is over priced and too small. m_2 and m_3 seem equally feasible for 2 out of 3 requirements are matched. Furthermore, if the price attribute is more important than the color then m_3 will be chosen, if color is more important then m_2 will be chosen.

[3] Note: ValAss is defined as a function which can also be considered a set.

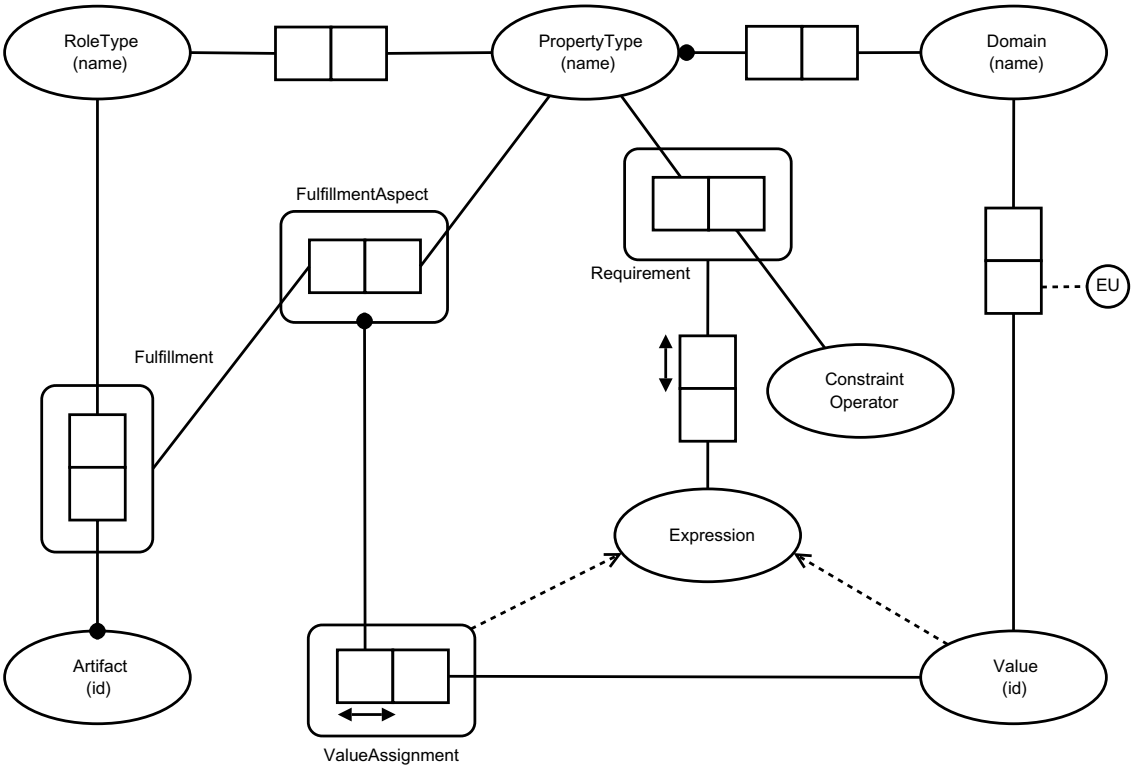

Fig. 3. Requirements and constraints.

Literature suggests numerous ways to deal with these kinds of selection/optimization problems such as Operations Research [35,47] and multi-objective decision making [17,9]. For example, one may opt to model this using a relative prioritization of the requirements (weighing of the requirements or using several objective functions). Discussing these approaches in detail is beyond the scope of this paper. Observe that it is important to decide what kind of problem is under consideration: finding the fulfillment which conforms to all constraints is a completely different problem than finding a fulfillment that is best, given these constraints!

However, to conclude the above example, as well as this section, we will show how the above *selection problem* may be solved adding weights to the requirements.

Example 3.9. Suppose that the following weights are added to the requirements:

Requirement	Weight
r_1	0.4
r_2	0.3
r_3	0.3

It is easy to verify that a considers m_2 to be of the highest quality (color is more important than price).

This concludes our exploration of the (general) notion of quality. In the upcoming sections we will explore the quality of *resources* on the Web as well as the quality of transformations on these resources. The following questions will guide our explorations:

- What does the notion of quality imply on the Web?
- Which role types, property types and domains can be used to describe resources and transformations?

- Which measurement methods can we use (for deciding whether an instance has a role type, a property type or is of some domain)?
- How can constraints be formulated?
- What kind of problem are we dealing with? Should we find the perfect resource or the best resource?

4. Quality of resources

In the previous section we have presented a framework for quality in two senses: quality in the sense of 'properties' and in the sense of 'desirability'. In the context of the Web these notions play an important role as well. This is particularly obvious in the context of *searching* on the Web: which resources (documents, pictures, movies, Web services) have a high quality for which searcher? As such, quality is synonymous to *aptness*.

In earlier work ([22,25]) we have extensively researched *information supply*. This resulted in a model with which we can characterize information supply. As such it can be used as a basis for describing quality in the sense of properties. In Section 4.1 we will introduce those parts of the model of interest for the discussion here.

This *reference model* for information supply is only part of the quality equation, however. From the previous section we know that from a user-perspective, quality is also expressed in terms of (constraints on) these properties. Therefore we propose to introduce a *formal language* with which we can express the requirements of searchers with regard to resources. This *query language* is introduced in Section 4.2.

4.1. Concepts

In this section we will present an overview of our model for information supply. The core concepts in this model are summarized in Fig. 4. We will firstly present a short formalization of our model. After that we will illustrate its use by means of a small example.

4.1.1. Formalization

Data resources are the central concept in our model as they represent the entities that can be found on the Web. We presume that data resources are identified by means of a URI [6]. Data resources can be a lot of

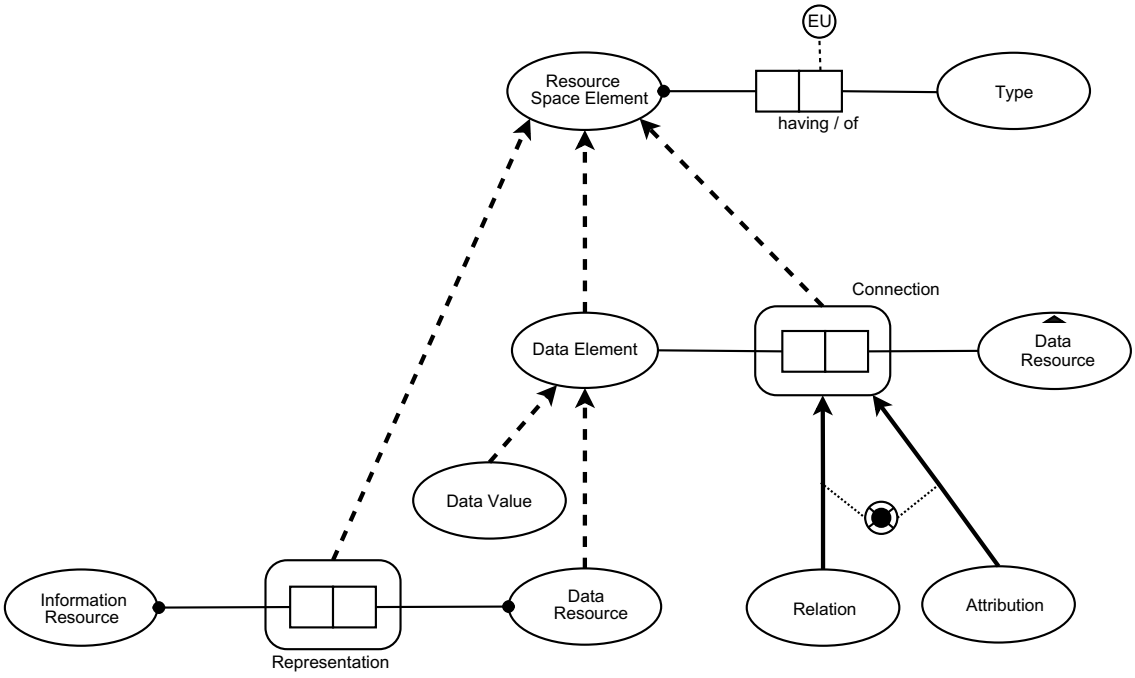

Fig. 4. A reference model for information supply.

things, such as Web pages, E-services, online databases or even people. Obviously many different data resource types exist.

We assume that data resources are always about something. To distinguish the raw data conveyed by data resources, and the 'things' they are about we introduce the concept of information resources. Information resources are the (real-world) objects that data resources may be about. We require that each data resource is about at least one information resource. Similarly, each information resource that we know about has at least one data resource associated to it.

Since different data resources can be about the same information resource, albeit in a different way, we introduce the concept of representations. In essence, representations represent the combination of a data resource and the information resource it is about and representation types model *how* aboutness is implemented. This allows us to model, for example, that one data resource is a picture of the Mona Lisa, whereas another is a detailed textual description of this famous painting.

Similar to the RDF approach (see e.g. [39]) we also make the distinction between data resources on the one hand, and data values on the other. Data values are literals that can not be addressed directly, that do not have meaning without an associated data resource. Examples would include the string €20 or *Dutch*. Data values are also typed.

The concept *data element* is a generalization of data resources and data elements. The distinction between these two leads to two different kinds of connections. On the one hand there are connections from data resources to data resources, which are dubbed relations. The most prominent example of such connections is the notion of hyperlinks [11,15] but other types of relations exist as well. On the other hand there are connections from data resources to data values, which are dubbed attributions. These allow us to model, for example, the price of a data resource, or its resolution. As such, attributions are also typed.

In our formalization we assume the following base sets:

Information Resource	$\mathcal{IR}$	Data Resource	$\mathcal{DR}$
Representation	$\mathcal{RP}$	Data Value	$\mathcal{DV}$
Relation	$\mathcal{RL}$	Attribution	$\mathcal{AT}$

Firstly, we require these sets to be disjoint:

Axiom 3 (*Disjoint Base Sets*). $\mathcal{IR}, \mathcal{DR}, \mathcal{RP}, \mathcal{DV}, \mathcal{RP}$ and $\mathcal{AT}$ are disjoint sets.

Collectively, the data resource and data values were dubbed data elements: $\mathcal{DE} \triangleq \mathcal{DR} \cup \mathcal{DV}$. Similarly, connections are either attributions or relations: $\mathcal{CN} \triangleq \mathcal{AT} \cup \mathcal{RP}$. This allows us to introduce a uniform way of modeling connections. Let $\mathsf{Src}, \mathsf{Dst} : \mathcal{CN} \to \mathcal{DE}$. As an abbreviation we introduce:

$$s \overset{c}{\rightsquigarrow} d \triangleq \mathsf{Src}(c) = s \wedge \mathsf{Dst}(c) = d$$

$$s \rightsquigarrow d \triangleq \exists_c [s \overset{c}{\rightsquigarrow} d]$$

To make the distinction between relations and attributions we must enforce that the destinations of connections point to the right elements:

Axiom 4 (*Relations*). $r \in \mathcal{RP} \Rightarrow \mathsf{Dst}(r) \in \mathcal{DR}$.

Axiom 5 (*Attributions*). $r \in \mathcal{AT} \Rightarrow \mathsf{Dst}(r) \in \mathcal{DV}$.

The aboutness of data resources is given shape using information resources and representations, which form the bridge between the abstract world of information resources on the one hand, and data resources on the other. Hence we define $\mathsf{IRes} : \mathcal{RP} \to \mathcal{IR}$ and $\mathsf{DRes} : \mathcal{RP} \to \mathcal{DR}$. The observation that each information resource should have some representation and each data resource should be involved in a representation is enforced by the following axioms:

Axiom 6. IRes is a subjective function.

Axiom 7. DRes is a surjective function.

Recall from the informal introduction of our model that data resources, data values, representations, relations and attributions are typed. To introduce a uniform typing mechanism over these base sets, let TP be the

set of all types and $\mathscr{RE} \triangleq \mathscr{DE} \cup \mathscr{CN} \cup \mathscr{RP}$ be the resource space elements that form the basis for the typing mechanism; then $\mathsf{HasType} \subseteq \mathscr{RE} \times \mathsf{TP}$ denotes the relation for typing. Observe that a $t \in \mathsf{TP}$ is both a type and an instance: it is the type in the real world, but an instance in the model. Furthermore, observe that resource space elements can have more than one type. This is, for example, the case with sub-typing (i.e. an *Xhtml* file is also an *Xml* file is also an *Ascii* file). To reason about types and instances we introduce:

$$\pi(t) \triangleq \{e | e \mathsf{HasType} t\} \tau(t) \triangleq \{t | e \mathsf{HasType} t\}$$
$$\pi(T) \triangleq \bigcup_{t \in T} \pi(t) \tau(E) \triangleq \bigcup_{e \in E} \tau(e)$$

In the above, π gives the population of a type (or set of types) and τ gives the types of an instance (or a set of instances). If $X \subseteq \mathscr{RE}$, in particular one of the basic sets such as $\mathscr{RP}$ or $\mathscr{DR}$, then we will abbreviate $\tau(X)$ with X_τ.

In our model we assume that *types follow population*, which means that the instances define which types exist in our world. This is in contrast with, for example, the world of relational databases where a schema is designed first and populated consecutively. As a consequence, if we have never encountered a document of type t then, in our model, type t does not even exist. As a consequence, we assume that all elements have a type and that all types have a population:

Axiom 8 (*Total typing*). $\tau(e) \neq \emptyset$.

Axiom 9 (*Existential typing*). $\pi(t) \neq \emptyset$.

Obviously, two types are equal when their populations are equal:

Axiom 10 (*Equal types*). $\pi(s) = \pi(t) \Rightarrow s = t$.

Last but not least, the partitioning of elements from resource space over $\mathscr{DR}, \mathscr{DV}, \mathscr{AT}, \mathscr{RL}$ and $\mathscr{RP}$ should be obeyed by their types as well:

Axiom 11 (*Partitioning of types*). $\mathscr{DR}_\tau, \mathscr{DV}_\tau, \mathscr{AT}_\tau, \mathscr{RL}_\tau$ and $\mathscr{RP}_\tau$ form a partition of TP.

4.1.2. Example

In this subsection we will present a small example population to illustrate the working of our model. It can be seen as a description of the value assignments in the quality-model as introduced in Section 3.2.

Let us assume that there are only two data resources in the world, each with only one type (we ignore subtying in the example):

davinci.html	HasType	*Html*
monalisa.eps	HasType	*Eps*

In other words, we already know that $\mathscr{DR} = \{davinci.html, monalisa.eps\}$ and that $\mathscr{DR}_\tau = \{Html, Eps\}$. The aboutness of the resources is given by:

$\mathsf{IRes}(r_1) = Leonaro\ DaVinci$ and $\mathsf{DRes}(r_1) = davinci.html$ and $r_1 \mathsf{HasType}\ Website\ about$
$\mathsf{IRes}(r_2) = The\ Mona\ Lisa$ and $\mathsf{DRes}(r_2) = davinci.html$ and $r_2 \mathsf{HasType}\ Website\ about$
$\mathsf{IRes}(r_3) = The\ Mona\ Lisa$ and $\mathsf{DRes}(r_3) = monalisa.eps$ and $r_3 \mathsf{HasType}\ Picture\ of$

From the above we can deduce that $\mathscr{RP} = \{r_1, r_2, r_3\}$ and that $\mathscr{RP}_\tau = \{Webiste\ about, Picture\ of\}$. The observation that the picture is included in the website (which is a special form or a hyperlink) is modeled using a relation r:

$$monalisa.eps \overset{r}{\rightsquigarrow} davinci.html \quad \tau(r) = \{Included\ in, hyperlink\}$$

Since there is only one relation we know that $\mathscr{RL} = \{r\}$ and that $\mathscr{RP}_\tau = \{Included\ in, hyperlink\}$. Last but not least, we know several attributions of both the website and the picture:

$$
\begin{aligned}
&monalisa.eps \overset{a_1}{\rightsquigarrow} 1024 \times 768 && a_1 \mathsf{HasType} resolution\\
& && 1024 \times 768 \mathsf{HasType} ResolutionString\\
&monalisa.eps \overset{a_2}{\rightsquigarrow} 24-06-2003, 10:12 && a_2 \mathsf{HasType} creationdate\\
& && 24-06-2003, 10:12 \mathsf{HasType} DateString\\
&davinci.html \overset{a_3}{\rightsquigarrow} 24-06-2003, 16:45 && a_3 \mathsf{HasType} modificationdate\\
& && 24-06-2003, 16:45 \mathsf{HasType} DateString
\end{aligned}
$$

In other words, the picture has a resolution and a creation date. The website has a modification date associated to it. Both dates are of (data value) type *DateString* which can be defined elsewhere, for example by means of a regular expression. We know that:

$$
\begin{aligned}
&\mathscr{AT} = \{a_1, a_2, a_3\}\\
&\mathscr{AT}_\tau = \{resolution, creation\ date, modification\ date\}\\
&\mathscr{DV} = \{1024 \times 768, 24-06-2003, 10:12, 24-06-2003, 16:45\}\\
&\mathscr{DV}_\tau = \{ResolutionString, DateString\}
\end{aligned}
$$

Last but not least, the populations of the generalizations $\mathscr{DE}$, $\mathscr{CN}$ and $\mathscr{RE}$ is straight forward. Note that, in terms of the quality model introduced in the previous section, several *value assignments* can be derived. For example: the observation that the picture (artifact) in its role as an element on the Web (role type) has a certain resolution (property type). Specifying ones *requirements* with regard to the properties of resources on the Web can, however, be tedious. In order to facilitate this we will introduce a quality language in the next section. This language is specifically tailored to the above model.

4.2. Language

In this section we will present a quality-language. More specifically, we will present a language that makes use of the concepts as introduced in Section 4.1 with which user goals can be represented. To this end we must first introduce LISA-D, a query/constraint language for NIAM/ORM like information structures. In the discussion here we will discuss the *Predicator Set Model* (PSM) flavor of NIAM. In the following we will introduce the relevant parts of the PSM and LISA-D based on the discussions in [33,31,44].

4.2.1. PSM and LISA-D

In this section we will introduce PSM and LISA-D. We will make use of the example schema presented in Fig. 5. Information structures capture the syntax of PSM. An information structure consists of the following basic components:

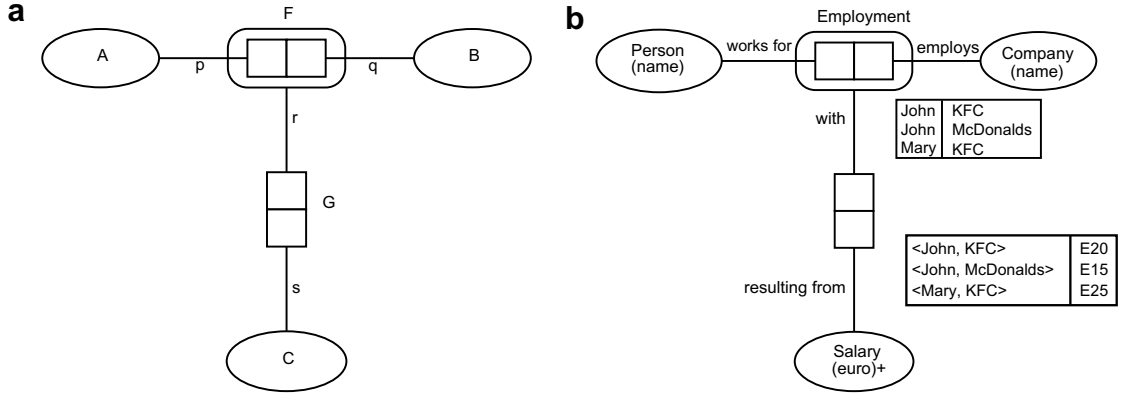

Fig. 5. Example: information structure and names.

- A finite sit $\mathscr{P}$ of *predicators*. In Fig. 5a: $\mathscr{P} = \{p, q, r, s\}$.
- A nonempty set $\mathscr{O}$ of *object types*. In Fig. 5a: $\mathscr{O} = \{A, B, C, F, G\}$.
- A partition $\mathscr{F}$ of $\mathscr{P}$. Elements of $\mathscr{F}$ are called *fact types*, which are also object types. In Fig. 5a: $\mathscr{F} = \{F, G\}$.
- The functions $\mathsf{Fact} : \mathscr{P} \to \mathscr{F}$ and $\mathsf{Base} : \mathscr{P} \to \mathscr{O}$ relate predicators to their respective fact types and object types. For example, in Fig. 5a: $\mathsf{Fact}(p) = F$ and $\mathsf{Base}(p) = A$. Note that the Fact relation is derivable, it is defined as follows:

$$\mathsf{Fact}(p) = f \iff p \in f$$

- In PSM a distinction is made between specialization (Spec, denoted as a bold arrow in PSM schema) and generalization (Gen, denoted as a dotted arrow in PSM schema). A full discussion of this topic is beyond the scope of this paper. The interested reader is referred to [33].

An information structure such as Fig. 5a is used as a frame for some part of the world, the universe of discourse (UOD). The state of the UOD corresponds to a population of the information structure. The population POP of an information structure $\mathscr{I}$ is the assignment of sets of instances to the object types in $\mathscr{O}$; $\mathsf{Pop} : \mathscr{O} \to \wp(\Omega)$, where Ω denotes the universe of all instances. Observe that the population of a fact type can thus be seen as a mapping from its predicators to a value of the population of their respective bases. Often, an ordering of the predicators is obvious from the representation of the scheme. In those cases we can denote such a mapping as a tuple.

Path expressions ($\mathscr{PE}$) correspond to a (directed) path through the information structure. Such path is interpreted as describing a relation between beginning and ending point. The semantics of a path expressions are defined as binary, inhomogeneous, tuple-oriented multi-relations over object types. They are built around constants, multisets, object types ($\mathscr{O}$) and predicators ($\mathscr{P}$). Let $\mu : \mathscr{PE} \to \Omega$ denote the semantics of a path expression. Before we can elaborate on μ we need to introduce the following auxiliary functions for the concatenation and reverse of multisets:

$$N \circ M \triangleq \lambda\langle x, y\rangle. \bigcup_{a \in X} N(x, a) \times M(a, y)$$

$$N^{\leftarrow} \triangleq \lambda\langle x, y\rangle. N(y, x)$$

Using these auxiliary functions we can now introduce the semantics of path expressions in two steps: atomic path expressions and composed path expressions:

Atomic path expressions:

Name	Expression	Semantics
Empty path	$\emptyset$	$\mu[\![\emptyset]\!] = \emptyset$
A constant	c	$\mu[\![c]\!] = \{[c, c]\}$
Multiset	X	$\mu[\![X]\!] = \{[\langle x, x\rangle\uparrow^1 — x \in X]\}$
An object type	x	$\mu[\![x]\!] = \{[\langle x, x\rangle\uparrow^1 \mid x \in \mathsf{Pop}(x)]\}$
A predicator	p	$\mu[\![p]\!] = \{[\langle v(p), v\rangle\uparrow^1 \mid v \in \mathsf{Pop} \cdot \mathsf{Fact}(p)]\}$

Composed path expressions:

Name	Expression	Semantics
Concatenate	$P \circ Q$	$\mu[\![P \circ Q]\!] = \mu[\![P]\!] \circ \mu[\![q]\!]$
Intersection	$P \cap Q$	$\mu[\![P \cap Q]\!] = \mu[\![P]\!] \cap \mu[\![q]\!]$
Union	$P \cup Q$	$\mu[\![P \cup Q]\!] = \mu[\![P]\!] \cup \mu[\![q]\!]$
Minus	$P - Q$	$\mu[\![P - Q]\!] = \mu[\![P]\!] - \mu[\![q]\!]$

Furthermore, several operators can be defined for path expressions such as counting, summarizing etcetera. For our purposes the *front* operator is important. For path expression P the operator $\not\vdash P$ isolates the front elements of a path.

There are many more calculations on multisets and path expressions that we ignore in this article. For our purposes the above will suffice. Recall that the path expressions enable us to reason about the population of the PSM schema. We will now introduce LISA-D with which we can add a 'syntactical sugar layer' on top of path expressions which would lead to natural, readable expressions. This is achieved by adding names to the PSM schema in the following manner:

- Let $\mathscr{N}$ be the set of all names.
- Object types are referenced by a unique name: $\mathsf{ONm} : \mathscr{O} \rightarrowtail \mathscr{N}$.
- Predicators are referenced by a unique name: $\mathsf{PNm} : \mathscr{P} \rightarrowtail \mathscr{N}$.
- Role names correspond to special connections (in the form of path expressions) through (binary) fact types: $\mathsf{RNm} : \mathscr{P} \rightarrowtail \mathscr{N}$.

The actual naming is administered by the function $\mathsf{Path} : \mathscr{O} \times \mathscr{O} \times \mathscr{N} \rightarrowtail \mathscr{PE}$ that assigns, in a given context, a path expressions to a name. For optimization purposes, beginning and endpoints of the paths are registered in the dictionary. That is, in case of $\mathsf{Path}(x, y, N) = P : N$ describes a path from x to y that should be interpreted as P. Naming works as follows:

- The name $\mathsf{ONm}(x)$ of object type x stands for path expression $x : \mathsf{Path}(x, x, \mathsf{ONm}(x)) = x$
- If p a predicator then PNm describes a path from the base of p to its corresponding fact type: $\mathsf{Path}(\mathsf{Base}(p), \mathsf{Fact}(p), \mathsf{PNm}(p)) = p$
- If predicator p of a binary fact type $f = \{p, q\}$ has a role name then this role name corresponds to the path through the fact type: $\mathsf{Path}(\mathsf{Base}(p), \mathsf{Base}(q), \mathsf{RNm}(p)) = p \circ q$
- Constants do not, in essence, form paths. As such $\mathsf{Path}(*, *, c) = c$

LISA-D is built around *information descriptors* which boil down to the names of the paths as shown above. The function $\mathbb{D} : \mathscr{N} \rightarrow \mathscr{PE}$ translates information descriptors to paths. The lexicon Path contains all atomic information descriptors:

$$\mathbb{D}[\![N]\!] = \bigcup_{\mathsf{Path}(x,y,N)!} \mathsf{Path}(x, y, N)$$

Single object types, predicator names and role names are atomic information descriptors. More fruitful information descriptors emerge by making combinations by means of concatenation:

$$\mathbb{D}[\![P_1 P_2]\!] = \mathbb{D}[\![P_1]\!] \circ \mathbb{D}[\![P_2]\!]$$

LISA-D supports several path constructors which can be grouped into two classes: constructors that are head-oriented (i.e. that only take the heads of paths into account) and head-tail constructors. In this paper we only need the former class, most notably:

$$\mathbb{D}[\![\mathsf{PAND} - \mathsf{ALSOQ}]\!] = \not\vdash\mathbb{D}[\![P]\!] \cap \not\vdash\mathbb{D}[\![Q]\!]$$
$$\mathbb{D}[\![\mathsf{POR} - \mathsf{ELSEQ}]\!] = \not\vdash\mathbb{D}[\![P]\!] \cup \not\vdash\mathbb{D}[\![Q]\!]$$
$$\mathbb{D}[\![\mathsf{PBUT} - \mathsf{NOTQ}]\!] = \not\vdash\mathbb{D}[\![P]\!] - \not\vdash\mathbb{D}[\![Q]\!]$$

Using the above mechanism we are able to present the details of the example presented in Fig. 5. We start by adding names to the object types and predicators in Fig. 5a which results in Fig. 5b. Part of the 'dictionary' is:

- $\mathsf{Path}(A, A, \mathsf{Person}) = A$
- $\mathsf{Path}(A, B, \mathsf{works\ for}) = p \circ q^{\leftarrow}$
- $\mathsf{Path}(B, A, \mathsf{employs}) = q \circ p^{\leftarrow}$
- $\mathsf{Path}(A, F, \mathsf{having}) = p$

- $\mathsf{Path}(F, A, \mathsf{of}) = p^{\leftarrow}$
- $\mathsf{Path}(*, *, \text{“KFC”}) = \text{“KFC”}$

Observe that Fig. 5a also presents a population for the schema, showing how People work for companies to earn their respective salaries. To see how the translation from LISA-D queries to path expressions and finally to answering the query in terms of the population works, we will work out two example queries:

The first query is to try to answer the question: which persons work for ''KFC''. This translates to the following path: Person works for Company with name ''KFC''. However, for purposes of this example we abbreviate this as follows:

$$\mathbb{D}[\![\mathsf{Person\ works\ for}\text{“KFC”}]\!] =$$
$$\mathbb{D}[\![\mathsf{Person}]\!] \circ \mathbb{D}[\![\mathsf{works\ for}]\!] \circ \mathbb{D}[\![\text{“KFC”}]\!] =$$
$$A \circ p \circ q^{\leftarrow} \circ \text{“KFC”}$$

We can now calculate which part of the population conforms to this path:

$$\mu[\![A \circ p \circ q^{\leftarrow} \circ \text{“KFC”}]\!] =$$
$$\mu[\![A]\!] \circ \mu[\![p]\!] \circ \mu[\![q^{\leftarrow}]\!] \circ \mu[\![\text{“KFC”}]\!] =$$
$$\mu[\![p]\!] \circ \mu[\![q^{\leftarrow}]\!] \circ \mu[\![\text{“KFC”}]\!]$$

In the remainder we will use quoted names to refer to the strings (i.e. ''John'') and omit the quotes when referring to the objects. That is, we use John as an abbreviation for Person with name ''John''. Working out the joins leads to:

from	*to*
John	$\langle John, KFC\rangle$
John	$\langle John, McDonalds\rangle$
Mary	$\langle Mary, KFC\rangle$

$\circ$

from	*to*
$\langle John, KFC\rangle$	KFC
$\langle John, McDonalds\rangle$	McDonalds
$\langle Mary, KFC\rangle$	KFC

$\circ$

from	*to*
KFC	“KFC”

$=$

from	*to*
John	“KFC”
Mary	“KFC”

A second example query concerns finding all people working for ''KFC'' with a Salary of 20 euro. This is verbalized by the expression following expression, which is illustrated in Fig. 6:

$$\mathbb{D}[\![\mathsf{Person\ having\ employment\ (with\ company}\ \text{“KFC”}\ \mathsf{AND-ALSO\ earning\ salary}\ \text{“E20”})]\!] =$$
$$\mathbb{D}[\![\mathsf{Person}]\!] \circ \mathbb{D}[\![\mathsf{having}]\!] \circ \mathbb{D}[\![\mathsf{Employment}]\!] \circ$$
$$(\mathbb{D}[\![\mathsf{with}]\!] \circ \mathbb{D}[\![\mathsf{Company}]\!] \circ \mathbb{D}[\![\text{“KFC”}]\!] \circ \mathbb{D}[\![\text{“AND-ALSO”}]\!]$$
$$\mathbb{D}[\![\mathsf{earning}]\!] \circ \mathbb{D}[\![\mathsf{Salary}]\!] \circ \mathbb{D}[\![\text{“E20”}]\!]) =$$
$$p \circ (\not{=}(q^{\leftarrow} \circ \text{“KFC”}) \cap \not{=}(r \circ s^{\leftarrow} \circ \text{“E20”}))$$

The expressions $\mu[\![q^{\leftarrow} \circ \text{“KFC”}]\!]$ and $\mu[\![r \circ s^{\leftarrow} \circ \text{“20”}]\!]$ result in:

from	*to*
$\langle John, KFC\rangle$	KFC
$\langle MARY, KFC\rangle$	KFC

and

$\langle John, KFC\rangle$	E20

respectively. The remainder of the calculation is straightforward. Taking the heads and performing the intersection leads to a path expression from $\langle John, KFC\rangle$ to $\langle John, KFC\rangle$. After joining with $\mu[\![p]\!]$ we get the answer to the query which is:

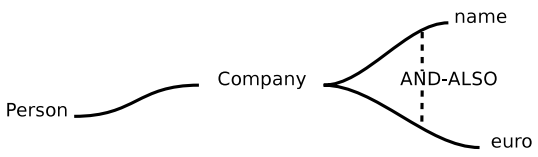

Fig. 6. Example path.

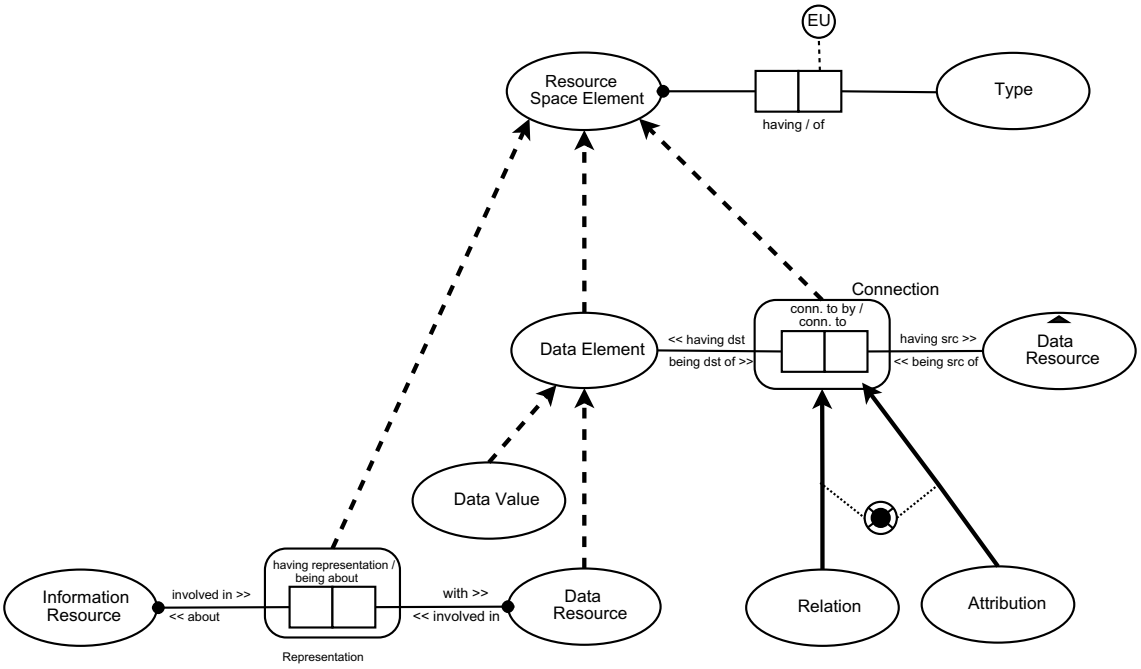

Fig. 7. A reference model for information supply.

from	*to*
John	⟨John,KFC ⟩

4.2.2. Language for resource space

In the previous subsection we have introduced PSM and LISA-D. Furthermore, we have shown how the semantics of LISA-D statements can be calculated in terms of the population of a PSM-schema. In this section we will present the LISA-D on top of the model for resource space which was already presented as a PSM-schema in Fig. 4.

In that figure we already added names to all object types and the role-names. However, we did not include the names for the paths from object types directly to other object types (For example, for the path from *Data Resource* to *Representation*). These names are included in Fig. 7.

This allows us to create, for example, the following expressions:

Aboutness:

- Data resource involved in Representation
  Finds the data resources that are involved in a specific representation.
- Data resource involved in Representation having type ''webpage''
  Finds all data resources that are webpages.
- Data resource involved in Representation (having type ''webpage'' AND-ALSO about ''Van Gogh'')
  Finds all data resources that are webpages about Van Gogh.

Relations:

- Data resource being src of relation having type ''hyperlink''
  Finds all data resources with outgoing hyperlinks.
- Data resource being src of relation having destination ''vangogh.html''
  Finds all data resources that are, somehow, connected to the data element (in this case: data resource) *vangogh.html*.

- Data resource being Dst of relation (having src ''vangogh.html'' AND-ALSO having type ''hyperlink'')
  Finds all data resources that have hyperlink-relations to *vangogh.html*.

Attributions:

In order to make the attribution-related LISA-D statements more readable we introduce two aliases: having ≙ being src of and with value ≙ having dst data value.

- Data resource having attribution of type ''version''
  Finds all data resources that have a version attribute. This would expand to Data resource being src of connection having type ''version''.
- Data resource having attribution (with value ''2.0'' AND-ALSO of type ''version'')
  Finds all data resources that have a version attribute with value ''2.0''.

These expressions can, in turn, be combined again to make even more complex expressions thus forming a language for specifying requirements (of a searcher) with regard to resource space. A typical example of a query that combines the above would be:

Data resource (having type ''EPS''
AND-ALSO involved in representation (about ''Mona Lisa'' AND-ALSO having type ''picture-of'')
AND-ALSO being dst of relation having dst ''davinci.html'')

This would find all pictures of the *Mona Lisa* in the *Eps* format that are, somehow, related to the webpage *davinci.html*.

5. Uncertainty in the real world

Assessing the quality (desirability) of some artifact for an actor is tricky, to say the least. As we have explained in Section 3, actors make quality assessments based on goals/constraints. These constraints are, usually, a *linguistic statement* such as: *I will assess the quality of this car to be high if its topspeed is high*, where it is unclear how high is to be interpreted. In other words, the quality assessment system has to deal with uncertainty about the constraints posed by the searcher.

A second kind of uncertainty has to do with the observations/measurements made by the system. For example:

- The fact that a resource has (outgoing) hyperlinks can be measured with near 100% certainty.
- The language of a resource is more difficult to measure. For example, consider the subtle differences between American English and British English, or between Dutch and Flemish, for that matter. It is possible that a quality assessment system can only establish the language of a resource with only 90% certainty.

In other words, the quality assessment system has to take different kinds of uncertainty into account as illustrated by Fig. 8. Quality assessment systems have to somehow deal with the uncertainty involved with measuring whether or to what degree a resource has a certain property, as well as determine the constraints that the actor uses for quality assessment. In order to come to a quality assessment of an artifact for an actor,

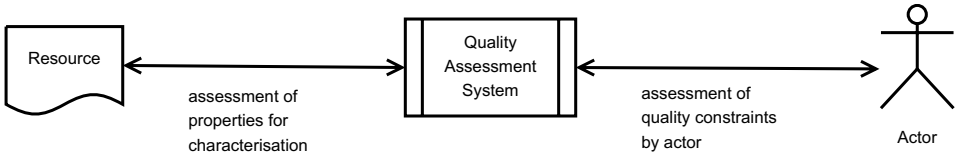

Fig. 8. Uncertainty in quality assessment.

the quality assessment system has to somehow combine the 'hard' (often numerical) measurements made with the 'soft' (and linguistic) classifications made by actors.

It turns out that the concept of a *linguistic variable* provides an elegant way to model the fuzzy assessments made by actors. In Section 5.1 we will firstly introduce the concept of a variable and in Section 5.2 we will introduce the concept of a linguistic variable based on [50–54]. In our discussion of linguistic variables we will adopt the same notation as used in Zadeh's papers. In Section 5.3 we will present our view on quality which uses the fuzzy concept of a linguistic variable. We will illustrate how this can be used to come to an actual measurement for the quality of a resource to a searcher by means of an extensive example.

5.1. Variable

In this section we will present the concept of a linguistic variable based on the work of Zadeh. Wikipedia[4] defines a variable as follows:

> In computer science and mathematics, a variable is a symbol denoting a quantity or symbolic representation. In mathematics, a variable often represents an unknown quantity; in computer science, it represents a place where a quantity can be stored. Variables are often contrasted with constants, which are known and unchanging.

In [50] the following formal definition of a variable is presented: A variable is characterized by a triple $\langle X, U, R(X;u)\rangle$, in which X is the name of the variable; U is the universe of discourse (finite or infinite set); u is a generic name for the elements of U; and $R(X;u)$ is a subset of U which represents a restriction on the values of u imposed by X. For convenience we shall abbreviate $R(X;u)$ to $R(X)$ and will refer to $R(X)$ simply as the restriction on u. In addition a variable is associated with an assignment equation $x = u{:}R(X)$ which represents the assignment of value u to x subject to $R(X)$.

The above can be extended which leads to the introduction of joint variables $X = (X_1, \ldots, X_n)$ with universe of discourse $U = U_1 \times \ldots, \times U_n$ and restriction $R(X_1, \ldots, X_n)$ a relation in U. This relation is characterized by its membership function: $\mu_R{:}U \to \{0, 1\}$ where:

$$\begin{aligned} \mu_R(u) &= 1 \quad \text{if } u \in R(X) \\ &= 0 \quad \text{otherwise} \end{aligned}$$

An example of the joint case would be the situation in which X_1 represents the age of a father and X_2 the age of his son with $U_1 = U_2 = \{1, 2, \ldots, 100\}$. Assuming that fathers are at least 20 years older than their sons leads to the following definition of $R(X_1, X_2)$:

$$\begin{aligned} \mu_R(u_1, u_2) &= 1 \quad \text{for } 21 \leqslant u_1 \leqslant 100, u_1 \geqslant u_2 + 20 \\ &= 0 \quad \text{otherwise} \end{aligned}$$

In case of joint variables the concept of *marginal restriction* plays an important role in the theory described by Zadeh. Since we do not need this concept for our theory we will now shift the focus to fuzzy variables.

5.2. Linguistic variable

The main distinction between fuzzy variables and non-fuzzy variables lies in the membership function. In case of non-fuzzy variables, an assignment of a value to value either conforms to the restriction or not. In case of a fuzzy variable this is not the case. A fuzzy variable is characterized by a triple $\langle X, U, R(X;u)\rangle$ where X is the name of the variable; U is the universe of discourse; u is a generic name for the elements of U; and $R(X;u)$ is a fuzzy subset of U which represents a fuzzy restriction on the values of u imposed by X. This fuzzy restriction is characterized by a membership function $\mu_R{:}U \to [0, 1]$ which represents the grade of membership with respect to the fuzzy restriction.

[4] http://www.wikipedia.org.

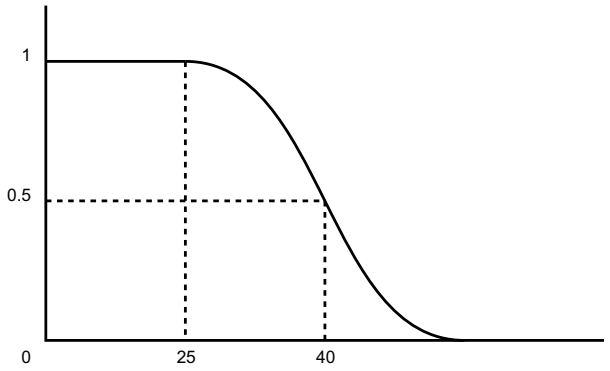

Fig. 9. Membership function for *young*.

Fig. 9 illustrates the membership function for the fuzzy variable *young* (denoted with y). The universe of discourse U, on the horizontal axis, is that of age in years. In the given example $\mu_y(40) = 0.5$.

Finally, we can turn our attention to the concept of linguistic variables which differ from normal, numerical, variables in that its values are not numbers but words, or sentences in some language. This makes the concept of a linguistic variable of a higher order than a fuzzy variable, in the sense that a linguistic variable takes fuzzy variables as its values. For example, the linguistic variable *age* might take *young*, *not young*, *old* or *not very old* as its values.

More formally, a linguistic is characterize by a quintuple $\langle \mathscr{X}, T(\mathscr{X}), U, G, M \rangle$ in which $\mathscr{X}$ is the name of the variable; $T(\mathscr{X})$ (or simply T) denotes the term-set of $\mathscr{X}$, that is, the set of names of *linguistic values* with each value being a fuzzy variable (denoted generically by X) ranging over U; G is a syntactic rule (which usually has the form of a grammar) for generating the names X of values of $\mathscr{X}$ and M is a semantic rule for associating with each X its meaning $M(X)$.

Continuing the previous example, let $\mathscr{X} = age$ be a linguistic variable with $U = [0,100]$. So we assume that people do not get older than *100* years. In this case *young* is considered to be a linguistic value of $\mathscr{X}$. More specifically, if $T(\mathscr{X}) = \{young, medium\ age, old\}$ then Fig. 10 illustrates the possible value assignments with their respective membership functions. In this example everyone below 25 years of age has membership degree 1 for the fuzzy variable *young* and everyone over 75 years of age has membership degree of 1 for the fuzzy variable *old*.

Frequently, the syntactic rule G that generates the terms in T is a context-free grammar such as, for example:

$$T \rightarrow young$$
$$T \rightarrow very\ T$$

The above example G is capable of generating terms such as *young*, *very young* but also *very ... very young*. To compute the meaning of such term one only needs the meaning of the term *young* (i.e. μ_{young}) and the meaning of the term *very*. The former is a *primary term*, that is, a term whose meaning must be specified as an membership function. The latter is a *linguistic hedge*, that is, a modifier of the meaning of its operand. These can be specified as function that operates on the membership function. The example membership function given in [51] for the variable *young* is as follows:

$$\mu_{\text{young}} = \begin{cases} 1 & \text{for } 0 \leqslant u \leqslant 25 \\ \left[1 + \left(\frac{u-25}{5}\right)\right]^{-1} & \text{otherwise} \end{cases}$$

Even more, if the interpretation of the hedge *very* is the square of the term to which it belongs then the interpretation of *very old* is the square of the above function.

Last but not least, the interpretations of the *fuzzy and*, *fuzzy or* and *fuzzy not* have to be defined. These are fairly straightforward and similar to their logical counterparts. Let $\sqcap$, $\sqcup$ and $\neg$ denote the fuzzy and, fuzzy or

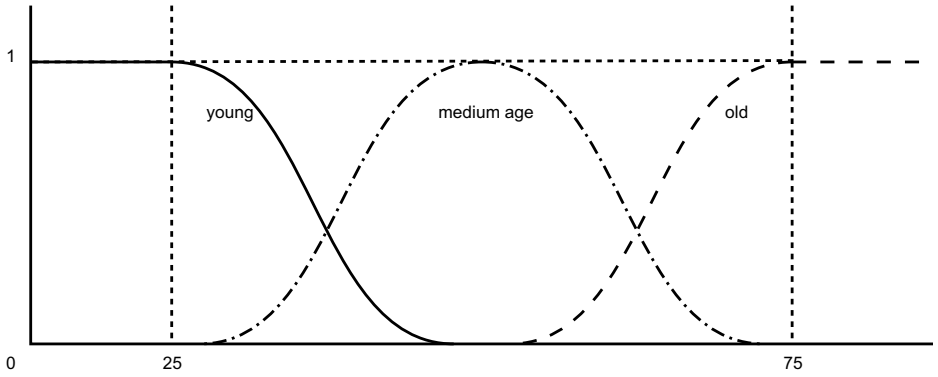

Fig. 10. The linguistic variable *Age*.

and fuzzy not. Furthermore, assume we have a linguistic variable $\mathscr{X}$ with underlying domain U and restriction R. Let X_1 and X_2 be two linguistic values of this variable (i.e. $X_1, X_2 \in T(\mathscr{X})$) such that for a given object o we have:

$$\mu_{R(X_1)}(o) = p_1 \quad \text{and} \quad \mu_{R(X_2)}(o) = p_2$$

Then for this object we have the following membership degrees:

- $X_1 \sqcap X_2 = \min(p_1, p_2)$
- $X_1 \sqcup X_2 = \min(p_1, p_2)$
- $\neg X_1 = 1 - p_1$

5.3. Fuzziness and quality

In the previous subsections we have explained the two kinds of uncertainty that play a role in quality assessments. Furthermore, we have introduced the concept of a linguistic variable. In this section we will elaborate on this discussion and present our view on quality assessment of resources on the Web from the perspective of a searcher.

Recall from Section 3 the assessment of the quality of some artifact is always done for a specific actor. More specifically, actors (unconsciously) use a set of requirements/constraints to determine the quality of an artifact. These requirements are often 'soft' in the sense that they can not be measured directly. Some examples are:

- The resource must have a high pagerank.
- The resource must be recent.

In Section 4 we have presented a language with which we are able to express 'hard' requirements. At first sight it does seems to make sense to translate the above requirements as:

- Data resource having attribution (with value ''high'' AND-ALSO of type ''pagerank'').
- Data resource having attribution (with value ''recent'' AND-ALSO of type ''modification date'').

However, under the assumption that 'high' and 'recent' are fuzzy values which are somehow mapped to their respective hard domains it does *not* make sense to simply follow this approach. This fuzziness must somehow be dealt with. A second issue that we already pointed out in previous sections is the observation that one may not be 100% certain about measurements. For example: How accurate is the measurement that a mug has a certain volume? How accurate is the measurement of the maximum speed of a car?

5.3.1. Measurements

Firstly we must define what it means if we assert that we measure some property of an artifact (to have a certain value) with some degree of certainty. An important observation in this respect is that measurements

depend on the situation in which they are done. For example, measuring the weight of an artifact depends on the location (on the moon, versus earth). Furthermore, the measuring device is another cause for concern. For example, one thermometer may be less accurate than another. To model this we introduce the set $\mathscr{SI}$ to be the set of all possible situations and $\mathscr{MD}$ to be the set of all measuring devices.

Two additional observations are relevant to our discussion here. First of all, two different kinds of measurements can be done:

1. One can attempt to measure the value of some property of an artifact.
2. One can attempt to verify whether the value associated to a property of an artifact equals some value.

This implies that a measurement always results in some value. In the first case it is the value that is measured but in the second case it would be a Boolean true/false. Let $\mathscr{MV}$ be the union of all possible value domains. A measuring device $R \in \mathscr{MD}$ can now be modeled as a function that maps object-situation combinations into values:

$$R = [\mathscr{AF} \times \mathscr{SI}] \rightarrowtail \mathscr{MV}$$

Furthermore, we can denote a specific measurement with $\mathsf{M}(a, s, d) = v$ where a denotes the artifact under consideration, s the present situation, d the measuring device and finally v the actually observed value. The following example illustrates how this may be used.

Example 5.1. Let c be the car or a John Doe. At a certain point in time, John is driving down the highway somewhere in Europe. Let s denote his situation, i.e. his current point in the space-time continuum. John happens to be so fortunate to drive past a police officer who users a certain device d which checks the speed of cars. The observation that John is driving at a speed of 125 km/h is expressed as: $\mathsf{M}(c, s, d) = 125$ km/h.

A remaining, yet very important, issue is: what about the accuracy of measurements? In this context one must realize that (values of) measurements are expressed in a domain and that there are standards for expressing them. For example, speed can be measured in terms of kilometers per hour, weight can be measured in terms of grams, distances in terms of meters and so on. Standards bodies (department of weights and measures) govern these standards. By comparing an actual measurement to the measurement by a standards body (we dub this the standard measurement) one obtains a metric for determining the accuracy of a measurement device. To continue the above example:

Example 5.2. Let d_s be an 'approved' measuring device for speed, i.e. it measures exactly according the department of weights and measures. This means that a measurement executed with this device is always 100% correct. If $\mathsf{M}(c, s, d) = \mathsf{M}(c, s, d_s)$ then we know that John was indeed driving exactly at 125 km/h.

In many cases a (very) small deviation of measurement can be allowed when comparing an actual measurement to a standard measurement. To put it differently, when determining whether an actual measurement is equal to a standard measurement one tests if they are *sufficiently equal*. We define $\stackrel{\circ}{=}$ to be an operator that measures if a measurement is sufficiently equal to a standard measurement.[5] In other words, a measurement is accurate (sufficiently equal to a standard measurement) if $\mathsf{M}(c, s, d) \stackrel{\circ}{=} \mathsf{M}(c, s, d_s)$.

Last but not least, we can relate the above discussion to the uncertainty involved with measurements. This uncertainty is caused by two things: the accuracy (or, if you wish, the quality) of the measurement devices and the many possible situations in which they are used. The following illustrates what we mean by this. Let d be a measurement device and d_s be a standard measurement device for the same domain. This measurements of device d can be tested against d_s in many (but not necessarily all) situations $S \subseteq \mathscr{SI}$. The accuracy of d is defined to be the average deviation of that device with respect to the situations in which it is tested:

$$\mathsf{Acc}(d) = \frac{\sum_{s \in S} \mathsf{M}(c, s, d) \stackrel{\circ}{=} \mathsf{M}(c, s, d_s)}{|S|}$$

[5] In a more elaborate theory it would be interesting to parameterize the $\stackrel{\circ}{=}$ to be able to specify the allowable deviation. This is, however, beyond the score of this paper.

This accuracy is the basis for defining the measurement uncertainty. That is, if we assert that (the value of) a property can be measured with a degree of certainty n then we mean that measurements done with this device are correct in $n\%$ of the situations.

5.3.2. Interpretation

The uncertainty involved with interpreting measurements is modeled similarly and makes use of linguistic variables. Let $\langle \mathscr{X}, T(\mathscr{X}), U, G, M\rangle$ be a linguistic variable. In the running example for this section, $\mathscr{X}$ represents the variable *volume of a mug* with term set $T(\mathscr{X}) = \{\text{big}, \text{medium}, \text{small}\}$. We interpret the membership degree for these linguistic values as the degree of certainty that we have in this specific interpretation of the actual measurement. Let $\mu_t{:}U \to [0 \ldots 1]$ denote the membership degree for the terms t in the term set. To set the stage, consider the following running example:

Example 5.3. In our example, the linguistic variable $\mathscr{X}$ denotes volume with term set $\{\text{small}, \text{medium}, \text{big}\}$. The domain U represents the volume in *cc*'s. The membership function for the linguistic value 'big' is given by:

$$\mu_b(u) = \begin{cases} 0 & u \leqslant 15 \\ \frac{1}{15}u - 1 & \text{otherwise} \\ 1 & u \geqslant 30 \end{cases}$$

and is drawn in Fig. 11. For ease of computation we have chosen the membership function to be linear.

In the running example we wish to answer the following question:

> Suppose I measure the volume of a mug to be 25 cm^3. What are the odds that this mug is considered to be big?

The answer to this question depends on the (accuracy of) measurements as previously described, but also on the interpretation of the linguistic value 'big'. The trick is to interpret the membership degree as certainty of interpretation. This requires a conversion of the (graph of the) membership degree function to a probability distribution.

By examining the increase of the surface under this membership function we get a cumulative probability distribution, provided that for each linguistic value v it holds that

Axiom 12. $\int_0^\infty \mu_v(u)\mathrm{d}u = 1$.

In our example it is easy to verify that this indeed the case. The certainty for our interpretation given measured value u and linguistic value v is given by:

$$P_v^i(u) = \int_0^u \mu_v(u)\mathrm{d}u$$

In our case, $P_b^i(25) = \frac{2}{3}$ indicates that we are approximately 67% certain that the contents of the mug will be assessed as 'big' and, consequently, that the quality of the mug will be 'high'.

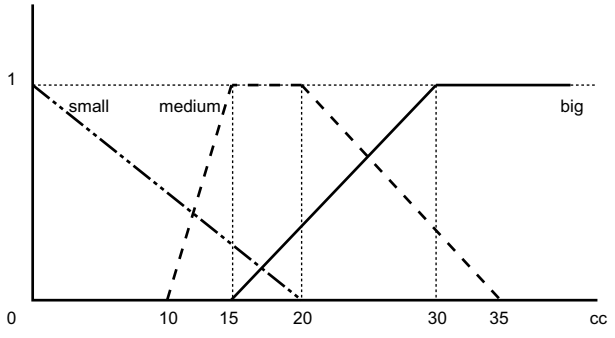

Fig. 11. Probability profile for the values of the linguistic variable 'volume'.

The question that remains is: how can these probabilities be combined to calculate the certainty of our quality assessment? Continuing the previous example:

Example 5.4. We use measuring device d to determine the contents of mug a in situation s. The accuracy of measurement $\mathsf{Acc}(d) = 0.9$. Let P^m denote this accuracy. The observed volume of this mug is $\mathsf{M}(a,s,d) = 25$ cc. Before we can compute $P(25 = big)$ we must define the membership functions for the linguistic values 'small' and 'medium'. We presume these to be:

$$P^i_s(u) = \begin{cases} 1 - \frac{1}{20}u & u \leqslant 20 \\ 0 & \text{otherwise} \end{cases}$$

$$P^i_m(u) = \begin{cases} 0 & u \leqslant 10, u > 35 \\ \frac{1}{5}u - 2 & 10 < u \leqslant 15 \\ 1 & 15 < u \leqslant 20 \\ \frac{35}{15} - \frac{1}{15}u & 20 < u \leqslant 35 \end{cases}$$

respectively. The membership functions are illustrated in Fig. 11. It is easy to verify that:

- the certainly that measured volume is indeed interpreted as 'big': $P^i_b(25) = 0.67$,
- the certainly that measured volume is indeed interpreted as 'medium': $P^i_s(25) = 0.67$,
- the certainly that measured volume is indeed interpreted as 'small': $P^i_m(25) = 0$.

We still have to combine the uncertainty involved with measurements and uncertainty as a result of interpretations in order to compute the certainty with which we can assess that an artifact is of high quality for an actor. This is computed by multiplying the P^m with $P^i_v(u)$. For our toy example this would mean:

Example 5.5. The certainty which we can assess that our mug is of high quality is: $0.9 \times 0.67 = 0.6$ (viz, the accuracy of the device multiplied by the interpretation uncertainty).

Before we move on to the quality of transformations, we will illustrate the theory introduced so far by means of an extensive example in the next section.

5.4. Example

In this section we will illustrate the theory introduced so far by means of an example. The setting of this example is as follows. A quality assessment system (from now on: the system) is assigned the task to assess the quality of an the newsletter of an online news site. The role of this site is 'informative medium'. In terms of our formalism: $n \in \mathscr{AF}$ denote the newsletter and $r \in \mathscr{R}\mathscr{O}$ denotes the role played by this site. Furthermore, $f = \langle n, r\rangle$ is the fulfillment for this newsletter.

The assessment has to take place for a certain actor $a \in \mathscr{AC}$. We know that the actor has three requirements with regard to this artifact: $\mathsf{Req}(f) = \{r_1, r_2, r_3\}$ which are verbalized as follows:

r_1: Data resource involved in Representation having type ''newsletter''
r_2: Data resource having type ''Pdf''
r_3: Data resource having attribution (with value ''high'' AND-ALSO having type ''importance'')

These requirements translate to our formalism as follows:

$r_1 = \langle p_1, c_1, e_1\rangle$ where p_1 is the property type 'representation type', c_1 is the equality constraint and e_1 is the value expression 'newsletter

$r_2 = \langle p_2, c_2, e_2\rangle$ where p_2 is the property type 'data resource type', c_2 also refers to the quality constraint and e_2 is the value expression 'Pdf' (which is a data resource type in the model for resource space in Section 4.1)

$r_3 = \langle p_3, c_3, e_3\rangle$ where p_3 is the property type 'importance', c_3 again is the equality constraint and e_3 the value 'high'. Note that in this case the system must use a linguistic variable to represent this constraint since 'high' is a soft value. The underlying 'hard' domain for importance is chosen to be the *PageRank* metric.

To be able to make a quality assessment the system uses three measuring devices $d_1, d_2, d_3 \in \mathscr{MD}$, one for each constraint. The three measurements will be done in parallel; in other words, in one situation $s \in \mathscr{SI}$. Based on previous experiences and tests the system knows that:

d_1: is software tool that is designed with the sole purpose of determining whether a given artifact is a newsletter or not. Furthermore, $\mathsf{Acc}(d_1) = 0.95$ which means that the system is able to correctly judge whether a given artifact is actually a newsletter in 95% of the situations.

d_2: is a tool that checks the (data resource) types of artifacts. This general purpose tool has been trained extensively on all known types and therefore $\mathsf{Acc}(d_2) = 1$ means that assessments are always correct.

d_3: is a highly complex tool. It assumes that the PageRank is a good measure for importance's of artifacts but knows that this need not always be a 100% correct assumption; hence: $\mathsf{Acc}(d_3) = 0.9$.

As stated previously, the system uses a linguistic variable to express the values of the constraints. For r_1 and r_2 the membership function is straightforward; *1* if the condition is met and *0* if it isn't met. However, for r_3 the situation is a little more complex. The term set for this variable is {*low*,*average*,high} and the underlying domain $U = [0 \ldots 10]$ the domain for expressing pagerank. After careful consideration of the user-profile of a the system decides the following membership function for the linguistic value 'high':

$$\mu_{\text{high}}(u) = \begin{cases} 0 & 0 \leqslant u \leqslant 6 \\ \frac{1}{4}u - 1\frac{1}{2} & 6 < u \leqslant 10 \end{cases}$$

Finally, in situation s the system makes the following measurements:

$\mathsf{M}(n, s, d_1) = \mathit{true}$: which means that the system suggests that s is indeed a newsletter. Hence, the membership degree is 1.

$\mathsf{M}(n, s, d_2) = \mathit{Pdf}$: which means that the system suggests that s is a *Pdf* file. Hence, the membership degree is 1.

$\mathsf{M}(n, s, d_3) = 9$: which means that the observed pagerank for n is 9. The membership degree, then, is 0.75.

Last but not least we can compute the certainty with which the system can assert that n is of high quality to a:

- $P_{r_1} = 0.95 \times 1 = 0.95$
- $P_{r_2} = 1 \times 1 = 1$
- $P_{r_3} = 0.9 \times 0.75 = 0.675$

Finally the total quality is the multiplication of these three certainties which results in 0.64. This should be interpreted as: the system is able to assert with 64% certainty that newsletter n is of high quality to actor a.

The above example may seem "simple" in the sense that every step is straightforward. This is mainly due to the fact that in creating the above setting we assumed a perfect world with complete, and unambiguous information. We will conclude this example by listing several ways to extend the above example and studying the impact of these extensions.

- The first issue that we should study deals with the fact that we assumed that the system *knows* the entire set of requirements (i.e., the qualities) which form the basis for quality assessment. In practice this is often not the case, for example because searchers may find it difficult to express their information need. Several approaches have been developed to assist searchers in formulating their information need, such as *query by navigation* (see e.g., [32,5]). In terms of our example: if the quality assessment system does not know part of the requirements which have to be used to assess the aptness of resources then it can not possibly give an accurate assessment.
- The second aspect has to do with language that is used to express the requirements. Some searchers may find it difficult to use *restricted language* as defined in this paper. A "wrong" formulation of requirements results in aptness calculations which are incorrect in the sense that the true information need is not taken

into account. To remedy this situation it may be necessary to incorporate an additional factor in the aptness calculations which expresses the (experienced) proficiency of the searcher with the requirements language. Simply including this factor does not fundamentally alter the nature of the above example.

- Thirdly, the requirements as formulated may seem somewhat simple in the sense that there are not a lot of AND or OR connectors. Creating more complex requirements is definitely possible, especially when more language constructs are added to our language for resource space. Interpreting these requirements implies a little bit more computation in the sense that more path expressions have to be juggled (see Section 4.2). These computations have been studied extensively in literature (e.g. [31]).
- Also, adding *more* requirements does not alter the example significantly. Surely some additional computation is needed but the overall nature of the example does not change. The same goes for creating more complex membership functions for the linguistic variables.

In summary we propose that "this is as hard as it gets", at least conceptually.

6. Quality of transformations

So far we have focussed on (the quality of) resources on the Web. With the apparent growth of the Web, more and more of these resources are available to us online. Even more, resources can be *manipulated*. Examples of systems that manipulate resources online are translation services, bundling of resources on portals, abstract generation or file type conversions. In this section we study the quality effects that these *transformations* have on resources on the Web.

6.1. Transformations

In previous work we have presented a reference architecture for transformations on the Web (e.g., [22,24,23,25]). In this section we will briefly outline our framework for transformations so that we can study the quality of transformations in the next subsection. Transformations are defined to be systems that transform data resources (of a certain type) into other data resources. Let $\mathscr{T}\mathscr{R}$ be a set of transformations.

The semantics of a transformation specify what this transformation actually *does*. The semantics of a transformation is given by the function:

$$\mathsf{SEM} : \mathscr{T}\mathscr{R} \to (\mathscr{D}\mathscr{R} \to \mathscr{D}\mathscr{R})$$

In other words, transformations transform one data resource into another. As an abbreviation we use $\overrightarrow{T}$ to denote $\mathsf{SEM}(T)$. Any given transformation has a fixed input and output type for which it is defined, similar to the notion of mathematical functions having a domain and a range. In our formalism we model this using $\mathsf{Input}, \mathsf{Output} : \mathscr{T}\mathscr{R} \to \tau(\mathscr{D}\mathscr{R})$. As an abbreviation we introduce:

$$t_1 \xrightarrow{T} t_2 \triangleq \mathsf{Input}(T) = t_1 \wedge \mathsf{Output}(T) = t_2$$

to express that transformation T transforms data resources of type t_1 into data resources of type t_2. In our formalism, a transformation is identified by its semantics:

Axiom 13 (*Identity of transformations*). $\overrightarrow{T_1} = \overrightarrow{T_2} \Rightarrow T_1 = T_2$

Observe that transformations are defined at the typing level. We will now describe the relation with the instance level. Recall that a transformation is only defined for instances of the correct input type, and that it only produces instances of the specified output type. If a transformation is applied to a data resource which is not of its input type then this data resource will not be changed. The proper behavior of transformations at the instance level is enforced by the following axioms:

Axiom 14 (*Output of transformations*). $e \in \mathsf{Input}(T) \Rightarrow \overrightarrow{T}(e) \in \mathsf{Output}(T)$.

Axiom 15 (*Input of transformations*). $e \notin \mathsf{Input}(T) \Rightarrow \overrightarrow{T}(e) = e$.

Transformations may also be applied to sets of data resources. Let E be such a set and T a transformation, then the application of T to E results in a new set of data resources:

$$\overrightarrow{T}(E) \triangleq \{\overrightarrow{T}(e) | e \in E\}$$

This means the following. If a transformation T is applied to a set of data resources E then the transformation will transform all resources for which it is defined (Axiom 14). The instances in E that are not in its input type are left untouched (Axiom 15).

Another property of transformations is the fact that they are closed under composition. Transformations can be composed by performing one after the other. We therefore assume $\circ$ to be a binary operator on $\mathscr{TR}$ such that $\overrightarrow{T_1 \circ T_2} = \overrightarrow{T_1} \circ \overrightarrow{T_2}$ denotes transformation composition in terms of mapping composition. We can now prove the following:

Lemma 1. $\circ$ *is an associative operator for transformations.*

Proof. Since mapping composition is associative we may conclude this property from Axiom 13.

Note that we do not require transformations to have an inverse. The following example illustrates the composition of transformations. □

Example 6.1. Let $t_1 \xrightarrow{T_1} t_2$ and $t_3 \xrightarrow{T_2} t_4$ be two transformations such that $t_4 \neq t_2$. Let T denote a transformation with $\overrightarrow{T} = \overrightarrow{T_1 \circ T_2}$. If T is applied to a single instance then either one of two things can happen: (1) nothing happens; this is the case when e is not in the input types of T_1 and T_2. (2) e is actually changed; this is the case when the type of e is either the input type or T_1 or the input type of T_2. Similarly, if T is applied to a set of data resources then the above holds for each of the data resources in this set.

6.2. Measuring the quality of transformations

An interesting dichotomy is that of the internal quality of a transformation (how well does it perform its task) and the external quality of a transformation (how does the user perceive the effects of the transformations). A similar distinction is made in *recommender systems* where one distinguishes between the *internal* and *perceived* quality of recommendations.

Since we adopt a black-box approach to transformations, we are mainly interested in the external quality of transformations and the aptness metric enables to compute it as follows:

Definition 6.1 (*Quality of a transformation*). Quality of a transformation is measured by the expected increase of aptness of a data resource after this transformation has been applied to it. A positive score implies that the transformation is expected to increase the aptness of the data resource, whereas a negative score implies the inverse.

In other words, to be able to compute the (external) quality of transformations we need to know both the wishes of the searcher, the aptness of the data resource and the effects of transformations. In the remainder of this section we present a small example that illustrates the computation of the quality of transformations.

Let $e \in \mathscr{DR}$ be an artifact, r a role such that $f = \langle e, r \rangle$ a fulfillment. Furthermore, the requirements of a searcher are $\mathsf{Req}(f) = \{r_1, r_2\}$ such that:

$r_1 = \langle p_1, \text{high} \rangle$ p_1 a property represented by a linguistic variable with term-set {low,medium,high} and an underlying domain of real numbers

$r_2 = \langle p_2, high \rangle$ p_2 a property represented by a linguistic variable with term-set {low,medium,high} and an underlying domain of real numbers

The membership functions for the linguistic values "high" of both variables are respectively

$$\mu_{p_1,\text{high}}(u) = \begin{cases} 0 & 0 \leqslant u < 5 \\ \frac{1}{5}u - 1 & 5 \leqslant u < 10 \\ 1 & 10 \leqslant u \end{cases}$$

$$\mu_{p_2,\text{high}}(u) = \begin{cases} \frac{1}{15}u - 1 & 0 \leqslant u < 15 \\ 1 & 10 \leqslant u \end{cases}$$

Furthermore, let d_1 and d_2 be two perfect measuring devices with $\mathsf{Acc}(d_1) = \mathsf{Acc}(d_2) = 1$ and s be the situation in which measurements take place. The measurements and aptness computations are as follows:

$$\mathsf{M}(e, p_1, d_1) = 7 \text{ such that } \mu_{p_1,\text{high}}(7) = \frac{2}{5}$$

$$\mathsf{M}(e, p_2, d_2) = 8 \text{ such that } \mu_{p_2,\text{high}}(8) = \frac{8}{15}$$

$$P_{r_1} = 1 \times \frac{2}{5} = \frac{2}{5}$$

$$P_{r_2} = 1 \times \frac{8}{15} = \frac{8}{15}$$

$$\text{Aptness} = \frac{2}{5} \times \frac{8}{15} = \frac{16}{75} \approx 0.213$$

Assume that two transformations (either singelton or composed) exist to transform this artifact: $T_1, T_2 \in \mathscr{TR}$. For the first transformation:

$$\mathsf{M}(\overrightarrow{T_1}(e), p_1, d_1) = 10 \text{ such that } \mu_{p_1,\text{high}}(10) = 1$$

$$\mathsf{M}(\overrightarrow{T_1}(e), p_2, d_2) = 2 \text{ such that } \mu_{p_2,\text{high}}(2) = \frac{2}{15}$$

$$P_{r_1} = 1 \times 1 = 1$$

$$P_{r_2} = 1 \times \frac{2}{15} = \frac{2}{15}$$

$$\text{Aptness} = \frac{2}{15} \approx 0.133$$

Even though this transformation drastically improves the situation with respect to requirement r_1, it also seriously hampers the situation with respect to requirement r_2 which results in a lower aptness score. The quality of this transformation can now be computed as the relative increase in aptness score which equals $-\frac{3}{8}$. This negative score implies that this transformation is rejected since it only lowers the aptness score. For the second transformation we have:

$$\mathsf{M}(\overrightarrow{T_2}(e), p_1, d_1) = 8 \text{ such that } \mu_{p_1,\text{high}}(8) = \frac{3}{5}$$

$$\mathsf{M}(\overrightarrow{T_2}(e), p_2, d_2) = 10 \text{ such that } \mu_{p_2,\text{high}}(10) = \frac{2}{3}$$

$$P_{r_1} = 1 \times \frac{3}{5} = \frac{3}{5}$$

$$P_{r_2} = 1 \times \frac{2}{3} = \frac{2}{3}$$

$$\text{Aptness} = \frac{3}{5} \times \frac{2}{3} = \frac{2}{5} = 0.4$$

In this case the transformation improves upon the original data resources with respect to both requirement r_1 and r_2. In this case the quality of the transformation is $\frac{7}{8}$. The fact that this magnitude is positive implies that the transformation does increase the aptness of the original data resource.

7. Conclusion

In this paper we have studied *quality on the Web*. More specifically, our goal was to study the notion of quality in order to define (1) what it is and (2) explain how it can be used in practice as an aptness metric. More specifically:

> The goal of this article is to explore the notion of quality in the context of the Web; to explain what it is and how it can be used in practice.

In answering this question we have adopted a modeling approach, where our models are inspired by a thorough study of the literature on quality. From this study we have learned that there are two main aspects to quality. Firstly the word quality is used in the sense of *attributes*. For example, the qualities (attributes) of physical artifacts can be measured. Secondly, the word quality is used in the sense of *desirability*. The latter aspect of quality is somewhat comparable to the notion of *value* as used in e.g., micro economics; it expresses how "good" a certain artifact is for an actor with certain goals. The relation between these two aspects/interpretations of quality seems fairly obvious; if the qualities of an artifact are 'just right" for a certain actor then this actor will judge the artifact to be of high quality. This idea can also be applied to resources on the information market which leads to the notion of aptness.

We have developed a model for qualities (the first aspect of quality). The basis for this model is the observation that artifacts can play different roles for different users. The support for properties of these artifacts must thus be considered in the context of these roles. In case of the Web, the artifacts are called *data resources* and we can use our model for information supply (Section 4.2) for expressing properties. We extended this model to cater for the second interpretation of the quality concept. This interpretation boils down to estimating "how good" an artifact is, based on the property support of this artifact as well as the requirements of the actor with respect to this property support. In case of resources on the Web this implies that, in order to estimate the quality/aptness of resources, we must find out (1) the requirements of the searcher and (2) the actual property support.

With respect to the former, the query tends to be a good indicator, albeit far from complete. In our view, user models and similar approaches may be beneficial. In our approach, however, we assumed that the query covers all the requirements of the searcher that are used to determine the quality of resources. We observed that these requirements tend to be vague, or fuzzy. For example, consider the constraint "the resolution must be high". It is unclear *when* the resolution can be considered to be high. This may even be personal or dependent on a specific search. To deal with this form of interpretation uncertainty we modeled fuzzy requirements using the concept of a linguistic variable from fuzzy logic.

A second form of uncertainty is related to the latter, determining the actual property support of artifacts (i.e., resources on the Web): how accurately are the measuring devices that are used to assess the property support for artifacts? We know from physics that measurements may be somewhat inaccurate, and that the accuracy may even depend on the specific situation in which the measurement is done. We have extended our model to also include uncertainty (in our case: a percentage) which represents the accuracy of measuring devices.

The two forms of accuracy, together with the user requirements as well as the actual property support is the basis for quality/aptness computations. In our model, quality of an artifact (resource) for a certain actor can thus be computed by estimating the likelihood that the property support of the artifact is conform the desires of the actor, taking measurement and interpretation uncertainty into account.

When considering the implications of our work, it is important to realize that quality plays a key role to support transactions via the Internet. Note that this is relevant for the Internet in general, and for e-commerce in particular. Especially transactions in heterogeneous environments such as the Internet need a thorough foundation of quality. Several practical applications of the quality metric are possible. For example, when searching for specific resources on the Internet, the quality of resources found so far is relevant as well as the quality of possible transformations, each transforming a given resource into another format.

References

[1] K. Alder, The measure of all things: The Seven-Year Odyssey and Hidden Error That Transformed the World, Free Press, New York, USA, 2002, ISBN 074321675X.

[2] A.T. Arampatzis, T. Tsoris, C.H.A. Koster, Irena: Information Retrieval Engine based on Natural language Analysis, in: Proceedings of the RIAO'97 Conference, 1997, pp. 159–175.
[3] A.T. Arampatzis, T. Tsoris, C.H.A. Koster, Th.P. van der Weide, Phrase-based information retrieval, Information Processing & Management 34 (6) (1998) 693–707, December.
[4] A.T. Arampatzis, Th.P. van der Weide, C.H.A. Koster, P. van Bommel, Linguistically-motivated information retrieval, vol. 69, Marcel Dekker, New York, USA, 2000, pp. 201–222.
[5] F.J.M. Bosman, P.D. Bruza, Th.P. van der Weide, L.V.M. Weusten, Documentation, cataloging and query by navigation: a practical and sound approach, in: C. Nikolaou, C. Stephanidis (Eds.), Research and Advanced Technology for Digital Libraries, 2nd European Conference on Digital Libraries 98, ECDL 98, Heraklion, Crete, Greece, EU, volume 1513 of Lecture Notes in Computer Science, Berlin, Germany, EU, September 1998 Springer, pp. 459–478.
[6] T. Berners-Lee, Universal Resource Identifiers in WWW. Technical Report RFC1630, IETF Network Working Group, June 1994.
[7] Nigel Bevan, Quality in use: meeting user needs for quality, Journal of System and Software 49 (1) (1999) 89–96.
[8] P. van Bommel, B. van Gils, H.A. (Erik) Proper, M. van Vliet, Th.P. van der Weide, The information market: its basic concepts and its challenges, In: A.H.H. Ngu, M. Kitsuregawa, E.J. Neuhold, J.-Y. Chung, and Q.Z. Sheng, (Eds.), Web Information Systems Engineering – WISE 2005, New York, New York, USA, volume 3806 of Lecture Notes in Computer Science, pp. 577–583, Springer–Verlag, November 2005. ISBN 3540300171.
[9] P. van Bommel, Database Optimization: An Evolutionary Approach. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, EU, 1995, ISBN 9090082441.
[10] P.D. Bruza. Hyperindices: A novel aid for searching in hypermedia, in: A. Rizk, N. Streitz, J. Andre (Eds.), Hypertext: Concepts, Systems and Applications; Proceedings of the European Conference on Hypertext – ECHT 90, number 5 in Cambridge Series on Electronic Publishing, pp. 109–122. Cambridge University Press, Cambridge, United Kingdom, EU, 1990. ISBN 0521405173.
[11] V. Bush, As we may think, The Atlantic Monthly 176 (1) (1945) 101–108, July.
[12] P.D. Bruza, Th.P. van der Weide, Two Level Hypermedia – An Improved Architecture for Hypertext, in: A.M. Tjoa and R.R. Wagner, (Eds.), Proceedings of the Data Base and Expert System Applications Conference (DEXA 90), Springer, Vienna, Austria, EU, Berlin, Germany, EU, Berlin, Germany, EU, 1990, ISBN 3211822348, pp. 76–83.
[13] P.D. Bruza, Th.P. van der Weide, Stratified hypermedia structures for information disclosure, The Computer Journal 35 (3) (1992) 208–220.
[14] C.W. Cleverdon, The Significance of the Cranfield Tests on Index Languages, in: A. Bookstein, Y. Chiarmarella, G.E. Salton, V.V. Raghavan (Eds.), Proceedings of the 14th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'1991), Chicago, IL, USA, pp. 3–12, New York, New York, USA, October 1991. ACM. ISBN 0897914481.
[15] J. Conklin, Hypertext: an introduction and survey, IEEE Computer 20 (9) (1987) 17–41, September.
[16] P. Donzelli, B. Bresciani, Improving requirements engineering by quality modelling – a quality–based requirements engineering framework, Journal of Research and Practice in Information Technology 36 (4) (2004), November.
[17] U. Diwekar. Introduction to Applied Optimization, volume 80 of Applied Optimization. Springer, Berlin, Germany, EU, 2003. 1402074565.
[18] G.B. Davis, M.H. Olson, Management Information Systems: Conceptual Foundations, Structure and Development, McGraw–Hill, New York, USA, 1985.
[19] J. Eustace, Descartes Definition of Matter, First published in The Journal of the Limerick Phi (accessed 02.02.2006).
[20] T. Gilb, Principles of Software Engineering Management, Addison Wesley, Reading, MA, USA, 1988.
[21] Michael Gertz, M. Tamer Özsu, Gunter Saake, Kai-Uwe Sattler, Report on the Dagstuhl Seminar: 'data quality on the Web'. SIGMOD Rec. 33(1):127–132, 2004. ISSN 01635808.
[22] B. van Gils, H.A. (Erik) Proper, P. van Bommel, A conceptual model for information supply, Data & Knowledge Engineering 51 (2004) 189–222.
[23] B. van Gils, H.A. (Erik) Proper, P. van Bommel, and Paul de Vrieze, Transformation selection for aptness–based web retrieval, in: H.E. Williams, G. Dobbie (Eds.), Proceedings of the Sixteenth Australasian Database Conference (ADC2005), Newcastle, New South Wales, Australia, volume 39 of Conferences in Research and Practice in Information Technology Series, pp. 115–124, Sydney, New South Wales, Australia, January 2005. Australian Computer Society. ISBN 192068221X.
[24] B. van Gils, H.A. (Erik) Proper, P. van Bommel, Th.P. van der Weide, Transformations in Information Supply, in: J. Grundspenkis, M. Kirikova (Eds.), Proceedings of the Workshop on Web Information Systems Modelling (WISM'04), held in conjunction with the 16th Conference on Advanced Information Systems 2004 (CAiSE 2004), Riga, Latvia, EU, volume 3, pp. 60–78, Riga, Latvia, EU, June 2004. Faculty of Computer Science and Information Technology. ISBN 9984976718.
[25] B. van Gils, H.A. (Erik) Proper, P. van Bommel, and Th.P. van der Weide, Typing and transformational effects in complex information supply, Technical Report ICIS–R05018, Radboud University Nijmegen, Institute for Computing and Information Sciences, 2005.
[26] B. van Gils, H.A. (Erik) Proper, P. van Bommel, Th.P. van der Weide, Quality makes the information market, in: R. Meersman, Z. Tari (Eds.), Lecture Notes in Computer Science 4275, Springer, Berlin, Germany, EU,, 2006, pp. 345–359, October/November.
[27] F.A. Grootjen, Employing semantical issues in syntactical navigation, in: Proceedings of the 22nd BCS–IRSG Colloquium on IR Research, pp. 22–33, 2000.
[28] F.A. Grootjen, Indexing using a grammerless parser, in: 2001 IEEE International Conference on Systems, Man & Cybernetics (SMC2001), 2001. ISBN 0780370899.
[29] T.A. Halpin, Information Modeling and Relational Databases, From Conceptual Analysis to Logical Design, Morgan Kaufmann, San Mateo, CA, USA, 2001, ISBN 1558606726.

[30] M. Harrison, Principles of Operations Management, Pitman, United Kingdom, EU, London, 1996, ISBN 0273614509.
[31] A.H.M. ter Hofstede, H.A. (Erik) Proper, Th.P. van der Weide, Formal definition of a conceptual language for the description and manipulation of information models, Information Systems 18 (7) (1993) 489–523, October.
[32] A.H.M. ter Hofstede, H.A. (Erik) Proper, Th.P. van der Weide, Query formulation as an information retrieval problem, The Computer Journal 39 (4) (1996) 255–274, September.
[33] A.H.M. ter Hofstede, Th.P. van der Weide, Expressiveness in conceptual data modelling, Data & Knowledge Engineering 10 (1) (1993) 65–100, February.
[34] Aristotle (384–322 BCE). General Introduction. The Internet Encyclopedia of Philosophy, 2006 (accessed 02.02.2006).
[35] Soung-Hie Kim, Byeong-Seok Ahn, Group decision making procedure considering preference strength under incomplete information, Computers & Operations Research 24 (12) (1997) 1101–1112.
[36] D. Kulak, E. Guiney, Use Cases: Requirements in Context, second ed. Addison Wesley, Reading, MA, USA, 2003, ASIN 0201657678.
[37] V. Lala, A. Arnold, S.G. Sutten, L. Guan, The impact of relative information quality of e-commerce assurance seals on Internet purchasing behavior, International Journal of Accounting Information Systems 3 (4) (2002) 237–253, December.
[38] K.C. Laudon, J.P. Laudon, Management Information Systems, International Edition, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996, ISBN 0132328852.
[39] Ora Lassila, Ralph R. Swick, Resource Description Framework (RDF) Model and Syntax Specification. Technical report, W3C, February 1999.
[40] Ken Orr, Data quality and systems theory, Commun. ACM, 00010782 41 (2) (1998) 66–71.
[41] H.A. (Erik) Proper, P.D. Bruza, What is Information Discovery About? Journal of the American Society for Information Science 50 (9) (1999) 737–750, July.
[42] G. John van der Pijl, Quality of information and the goals and targets of the organization, in: SIGCPR '94: Proceedings of the 1994 computer personnel research conference on Reinventing IS: managing information technology in changing organizations, Alexandria, VA, USA, pp. 165–172, New York, NY, USA, 1994. ACM. ISBN 0897916522.
[43] M.P. Papazoglou, H.A. (Erik) Proper, J. Yang, Landscaping the information space of large multi–database networks, Data & Knowledge Engineering 36 (3) (2001) 251–281.
[44] H.A. (Erik) Proper, Th.P. van der Weide, Information disclosure in evolving information systems: taking a shot at a moving target, Data & Knowledge Engineering 15 (1995) 135–168.
[45] J.J. Sarbo, J.I. Farkas, F.A. Grootjen, P. van Bommel, Th.P. van der Weide, Meaning extraction from a Peircean perspective, International Journal of Computing Anticipatory Systems 6 (2000) 209–227.
[46] I. Sommerville, Software Engineering, Addison Wesley, Reading, MA, USA, 1989.
[47] H.A. Taha, Operations Research, An Introduction, fourth ed., Prentice-Hall, Englewood Cliffs, NJ, USA, 1992, 0131876597.
[48] E. Turban, J. Lee, D. King, H.M. Chung, Electronic Commerce, A Managerial Perspective, Prentice–Hall, Englewood Cliffs, NJ, USA, 1999, ISBN 0139752854.
[49] S. Weibel, J. Godby, E. Miller, R. Daniel, Metadata Workshop Report. Dublin, OH, USA, March 1995.
[50] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – I, Information Sciences 8 (1975) 199–249.
[51] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – II, Information Sciences 8 (1975) 301–357.
[52] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – III, Information Sciences 9 (1975) 301–357.
[53] Lofti A. Zadeh, From computing with numbers to computing with words – From manipulation of measurements to manipulation of perceptions, International Journal of Applied Mathematics and Computer Science 12 (2002) 307–324.
[54] L.A. Zadeh, Toward a generalized theory of uncertainty (GTU) – an outline, Information Sciences 172 (2005) 1–40.