Fundamentals of Quality on the Web

B van Gils and H.A. (Erik) Proper and P. van Bommel and Th.P. van der Weide

June 9, 2006

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; search engines assess the quality of resources for players.

In this paper we present a formal model for the notion of quality on the Web. We base our model on a thorough literature study of how the quality notion is used in different fields. Even more, we show how the quality of resources is affected by software manipulations (transformations).

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 Figure 1. In this paradigm the main challenges are:

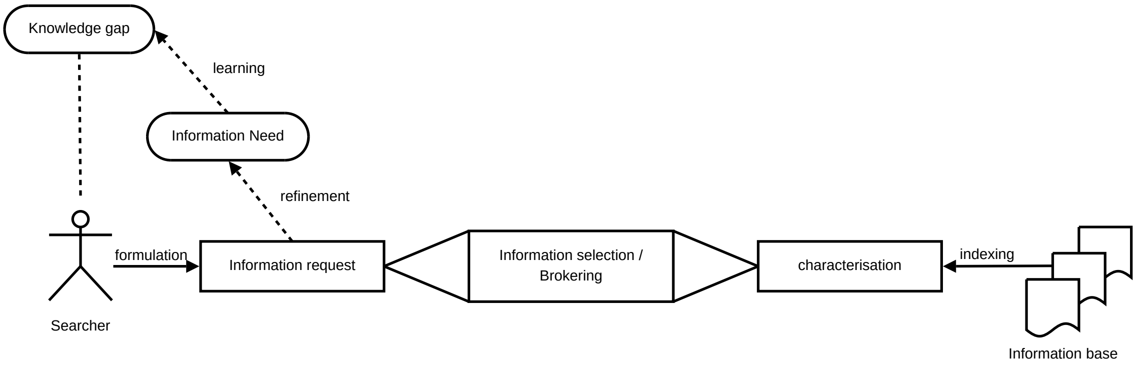


Figure 1: The information retrieval paradigm

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 [Cle91].

Characterizing supply – Good characterization of information resources is imperative for effective information discovery, as poor characterizations inevitably lead 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 [BL94, WGMD95].

Matching demand & 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: [BW90, Bru90, BW92, HPW96, PPY01], characterizing supply: [SFG$^+$00, Gro00, Gro01], matching demand & supply: [ATK97, ATKW98, AWKB00]).

The notion of *quality* seems to be particularly important in this area. 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?

Recognizing the fact that the quality-notion is important on the Web is one thing, but defining exactly what quality is, how it can be used or what that would imply in practice seems a daunting task. Literature (see Section 2 for an extensive survey of literature on quality) seems to suggest that there are two aspects of quality: qualities in the sense of attributes that artifacts may have and quality in the sense of desirability. 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 different (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. Observer, however, that we (humans) may associate a judgment (reasonable quality, poor quality) to this comparison.

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

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, our research goal is:

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.

We start out by presenting an extensive survey of the available literature about quality in different fields, ranging from philosophy to software engineering, in Section 2. We will use the insights gained from this survey to present a formal model for quality in Section 3. This model aims at combining 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

As was stated in the previous section, the notion of quality can only be considered in context. In this section we will present an overview of how quality is used in different fields. Furthermore, we analyze these definitions in the last subsection.

2.1 Dictionary

The *Webster's third new international dictionary, unabridged* (1981) has an extensive entry detailing quality. 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

The *Concise Oxford Dictionary* is, as its name implies, more concise. It states that quality is

> (1) the standard of something as measured against other things of a similar kind. (2) a distinguishing characteristic or characteristic.

The *Wikipedia*[1] relates the notion of quality to different fields:

> The term quality is used to refer to the desirability of properties or characteristics of a person, object, or process. In the case of a person this is considered in a particular context, such as worker, student, sports person, etcetera. The term is often used in opposition to quantity. In science, the work of Aristotle focused on measuring quality; whereas, the work of Galileo resulted in a shift towards the study of quantity.

It also describes that in manufacturing, the notion of quality relates to making a product fit for a purpose with the fewest possible defects (see also the ISO 9000 standard which specifies requirements for a Quality Management System overseeing the production of a product or service). Finally, quality can historically have four different interpretations: conformance to specifications, fitness for use, must-be/attractive quality and value to some person.

2.2 Philosophy

The notion of quality has a long history. For example, in his work on *the Philosophy of Nature* Aristotle used the notion of quality (e.g., [IEP06]):

[1] `http://en.wikipedia.org/wiki/Quality`

> Aristotle sees the universe as a scale lying between the two extremes: form without matter is on one end, and matter without form is on the other end. The passage of matter into form must be shown in its various stages in the world of nature. To do this is the object of Aristotle's physics, or philosophy of nature. It is important to keep in mind that the passage from form to matter within nature is a movement towards ends or purposes. Everything in nature has its end and function, and nothing is without its purpose. Everywhere we find evidences of design and rational plan. No doctrine of physics can ignore the fundamental notions of motion, space, and time. Motion is the passage of matter into form, and it is of four kinds: (1) motion which affects the substance of a thing, particularly its beginning and its ending; (2) motion which brings about changes in quality; (3) motion which brings about changes in quantity, by increasing it and decreasing it; and (4) motion which brings about locomotion, or change of place. Of these the last is the most fundamental and important.

In other words, quality is the category according to which objects are said to be like or unlike. Other great philosophers such as Descartes, Bacon, Newton, and Galileo oppose to Aristotle's view on (the quality of) matter (see e.g., [Eus]):

> It is interesting to note that Descartes' great contemporary, Galileo, had a very similar approach to matter. Like Descartes and Bacon he was strongly opposed to the scholastic or Aristotelian tradition. Like Descartes, he distinguished between the real and the apparent qualities of matter - a distinction which goes back to the pre-Socratic atomists. A century later Newton was also to make the same point in his optics. For Galileo the real or objective qualities of matter are extension in space, figure, number and motion wherever color, taste, smell, bitter or sweet 'are no more than mere names so far as the object in which we place them is concerned'. (Letter to Virginia Cesarini). Galileo could not begin his investigations into the mechanics of matter without some kind of definition of matter, an operational definition of matter. He needed to decide which properties of matter were essential and which could be neglected.

In other words, Descartes makes a distinction between the objective qualities of matter and its largely subjective qualities.

2.3 E-commerce

In [TLKC99] 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.

In [LASG02] the notion of quality of information is related to E-Commerce. The (lack of) quality of information about assets which can be either products or services can pose a risk for web-consumers. For example: a financial risk, a performance risk, risk for loss of time/convenience. In the description of a real-life situation (selling insurance via the Web), it is stated that:

> Controlling the information quality dimension is more challenging as quality can be addressed through either process or outcome measures. Process measures relate to the process used to complete an insurance engagement and the established standards for conducting such engagement. Outcome measures related to the amount of increased confidence (value) a consumer places on an insurance engagement by simply receiving the report.

In other words, a distinction can be made between process quality and the quality of the actual outcome too.

2.4 Operations Management

Quality is an important notion in operations management. In [Har96] an entire chapter is devoted to the topic of managing quality, involving concepts such as total quality management, quality of service etcetera. The key dimensions of quality are defined to be:

- product attributes
- product performance
- service characteristics
- warranty
- service availability
- total price

In the context of operations management, the question 'How can operations contribute to delivering a quality product?' must be answered with: operations is concerned with deciding on the most suitable production process through job design, production planning and control, obtaining resources for production and with quality control in the sense of ensuring that products leaving the workplace conform to specifications. *Conformance to specification* is the central theme in operations research, especially when Total Quality Management (TQM) is considered. TQM emphasizes, at every link in the production chain, the need to arrive at agreement on performance requirements, supplier capability, timing, cost, and the monitoring of changing needs. To put it in the words of [LL96]: TQM is a concept that makes quality the responsibility of all people within an organization.

The conformance to specification approach is criticized in [LL96] for its sole focus is the supplier perspective. The consumer perspective is, according to the authors, more concerned with *value for the dollar* (i.e., getting your money's worth). This includes both characteristics of the product/service that is bought and psychological aspects such as how knowledgeable the support staff is, courtesy of the staff etcetera.

The focus in [Pij94] is on the ex-post evaluation of quality of information in organizations. The ISO-8402 definition of quality:

> The totality of features and characteristics of a product, process or service that bear on its ability to satisfy stated or implicit goals.

is used as a starting point. The author makes several observations:

- Any conceptual quality model should take account of the importance of the production process.
- The quality of a product or service has to be considered in the light of the use that is made of it.
- Quality is described in terms of a series of features and characteristics of a product, service or process.

In his quality mode, Van der Pijl proposes a dual view on quality: the *causal point of view* deals with the quality of information, seen as the result of the quality of the process in which it is produced. In the *teleological point of view* the quality of information is seen as the degree to which it satisfies stated or implicit needs, derived from the situation in which it is used.

2.5 Software Engineering

In the field of software engineering the notion of quality plays two important roles: the quality of the software itself on the one hand and quality of the software engineering process on the other.

[Som89] discusses quality in the chapter on *Quality management*. The author starts with the observation that the classical notion of quality (conformance to specification, see Section 2.4) is difficult to apply to software systems because:

- The specification should be oriented towards the characteristics of the product that the customers wants. However, the development organization may also have requirements which are not included in the specification.
- We do not know how to specify certain quality characteristics in an unambiguous way.
- It is very difficult to write complete software specifications. Therefore, although a software product may conform to its specification, users may not consider it to be a high-quality product.

It is emphasized that the quality of a software system can only be assessed in terms of *quality attributes* (such as safety, security, reliability, resilience, robustness, learnability etcetera) and that software quality management can be structured into three principle activities: quality assurance, quality planning, and quality control. Standards (product standards, process standards, documentation standards) play an important role in these activities. Last but not least, *software metrics* can be used to make quality measurable:

> Software measurement is concerned with deriving a numeric value for some attribute of a software product or a software process. By comparing these values to each other and to standards which apply across an organization, it is possible to draw conclusions about the quality of software or software processes.

Also in [DO85], where quality is defined as excellence or fitness, it is proposed to measure the quality of information systems by means of characteristics such as complete data, accurate data, relevant output, meaningful output etcetera. A highly systematic approach to measuring quality attributes of a system is needed if measurement using attributes is to succeed, for faulty measurements lead to an incorrect assessment of the quality of the system under consideration. In the classic work [BJ87] the author indeed corroborates that quality (of software products) can only be achieved with discipline and systematic software quality controls.

Even more, in [McC04] it is explained that the attempt to maximize certain aspects of quality inevitably conflicts with the attempt of others. At a certain level of abstraction this can be read as: increasing quality conflicts with increasing quality. In practice this is dealt with by prioritizing the different quality characteristics and maximize within certain bounds (i.e. a budget).

In [Gil88] the focus is on the process of software engineering where determining attribute specification is one of the difficult problems. Attributes are in two kinds:

Resources (people, time, money): are almost always limited

Qualities or benefits (performance, reliability): we always want more than we can afford.

Three principles that relate to this particular problem 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.

2.6 Quality on the Web

In [GÖSS04] a discussion on the quality of data on the Web is presented. This discussion starts off with the observation that:

> Well-founded and practical approaches to assess or even guarantee a required degree of the quality of data are still missing.

According to the authors, data quality comprises more than the format of the data (text, multimedia, streaming data); it has to do with how fit (*apt*) data is for consumers. Relevant keywords in this respect are accuracy, completeness, timeliness, and consistency. As such, data quality can not be studied in isolation. The data producers, custodians (entities that provide and manage data) as well as consumers have to be taken into account as well. The authors propose that a *quality algebra* be used for assessing/ dealing with data quality on the Web. Factors to take into account when designing such algebra are:

data quality assessment : the 'raw' and unweighted quality, independent from later usage.

data quality interpretation : uses the assessment output in order to perform reasoning on data quality using additional (user) information.

data quality dynamics : touches on areas like the history of data, data lineage, application evolution and change detection.

The above mentioned custodians are called information intermediaries in [VW99]. The authors pose that user concerns about (their perception of) the quality of information on the Web continues to be a strong incentive for "the emergence and success of information intermediaries." They can play an important role in the trust relationship between suppliers and consumers, as well as in quality/price control:

> Quality in information products is a complex and elusive phenomenon. It can be described on the basis of outcomes for their users and potential increase in the efficiency for the tasks they perform. A broader understanding of quality can comprise not only quality proper but also additional parameters or clearly qualitative nature.

The central observation in [Orr98] is that data quality is the measure of the agreement between the data view presented by an information system and some data in the real world. The authors propose the following 6 data quality rules for maintaining the quality of data in (web) information systems:

1. Unused data cannot remain correct for very long
2. Data quality in an information system is a function of its use, not its collection.
3. Data quality will be no better than its most stringent use.
4. Data quality problems tend to become worse as the information system ages.
5. The less likely some attribute (element) is to change, the more traumatic it will be when it finally does change.
6. Laws of data quality apply equally to data and meta-data.

2.7 Library Information Systems

Another field where the notion of quality plays an important role is the library and information science community. An interesting starting point in this respect is [DES05] which has a "comprehensive list of possible selection criteria" in the context of information gateways. This list can be summarized as follows;

Relating to the internal quality of resources	Relating to quality in the subject gateway context
Content criteria	Scope criteria
Form criteria	Collection criteria
Process criteria	

It is interesting to observe that the authors make a distinction between quality aspects pertaining to the resources themselves and quality aspects pertaining to the proces of locating / accessing the resources. In [HC05] the focus is on the latter, and gives an extensive list of efforts geared towards achieving a high quality of service. The authors observe that:

> Most researchers in library and informaton science have concentrated on the perspective of services qualit yas meeting and/or exceeding expectations.

The authors then elaborate on the several models for service quality such as LibQUAL and WebQUAL[2]. These approaches all have in common that they focus on quality attributes of the offered services. As an example, the LibQUAL approach lists the following *dimensions of library services quality*: reliability, affect of service, ubiquity of access, comprehensive collections, library as place and self reliance.

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.

3.1 Quality & 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. Figure 2 shows our model using the Object Role Modeling (ORM) notation[3], 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

[2] See `http://www.libqual.org` and `http://www.webqual.net` respectively

[3] See e.g. [Hal01] for an overview of this notation.

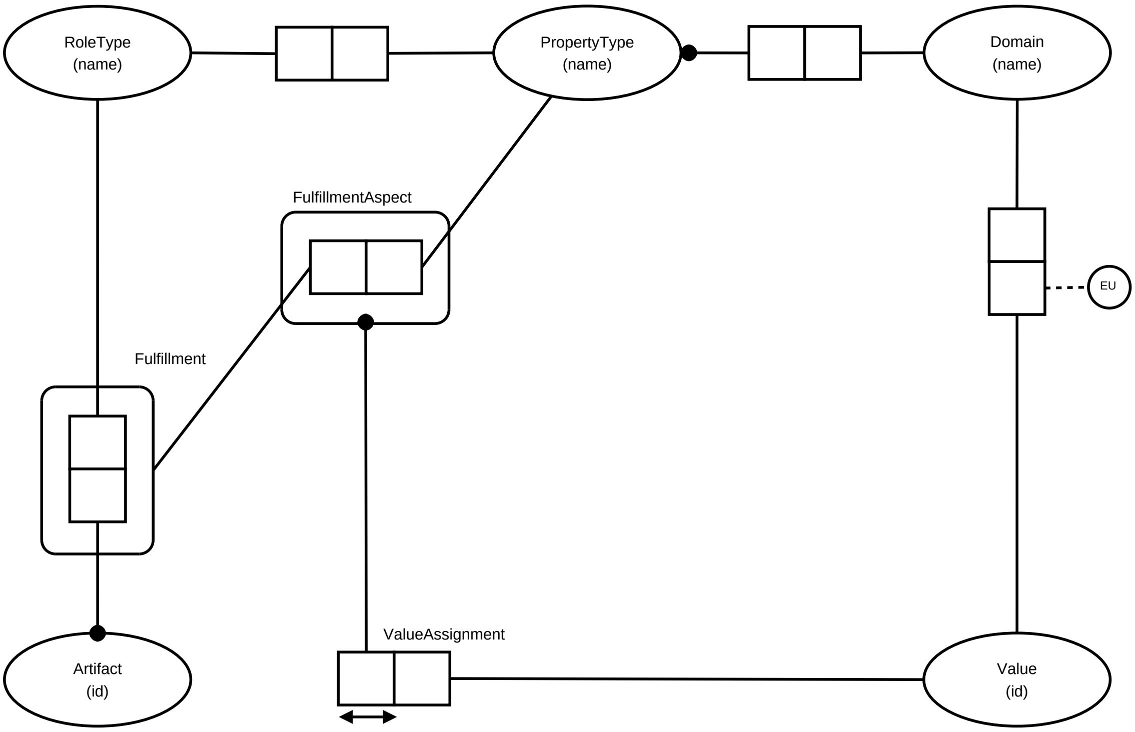


Figure 2: Properties of artifacts

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 $\mathcal{AF}$ be the set of all artifacts that may have certain qualities (properties) and let $\mathcal{RO}$ 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): $\mathcal{FL}$. The artifacts and roles that participate in a fulfillment can be found using the functions $\mathsf{Artifact} : \mathcal{FL} \rightarrow \mathcal{AF}$ and $\mathsf{Role} : \mathcal{FL} \rightarrow \mathcal{RO}$ 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)

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

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:*

$$\begin{array}{ll} \mathsf{Artifact}(e_1) = a & \mathsf{Role}(e_1) = r_1 \\ \mathsf{Artifact}(e_2) = a & \mathsf{Role}(e_2) = r_2 \end{array}$$

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, 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 $\mathcal{PT}$ be the set of property types and $\mathcal{PD}$ be the set of all property domains. The properties that can be played by a certain role type are given by the relation $\mathsf{Props} \subseteq \mathcal{RO} \times \mathcal{PT}$ and the domain in which (values of) a property can be expressed is given by the function $\mathsf{PrDom} \subseteq \mathcal{PT} \times \mathcal{PD}$. 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 $\mathcal{PD}$ and the values from this domain. The set $\mathcal{VL}$ consists of sets of values for a certain domain. In other words, an element from $\mathcal{PD}$ is the *names* of a certain domain and an element of $\mathcal{VL}$ consists of its values. In the ORM-schema (Figure 2) the extentional uniqueness constraint denotes the fact that the values uniquely determine the domain(name). The functions $\mathsf{Value} : \mathcal{PD} \to \mathcal{VL}$ and $\mathsf{VlDom} : \mathcal{VL} \to \mathcal{PD}$ 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 $20cc$. The property type of a fulfillment is denoted in our model by a *fulfillment aspect*. The set of these fulfillment aspects is denoted by $\mathcal{FA} \triangleq \mathcal{FL} \times \mathcal{PT}$ such that

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

The intended meaning is as follows:

Example 3.4 *Let $f = \langle mug, drinking\ device \rangle$ denote the fulfillment of a mug in its role as drinking device and let $color \in \mathcal{PT}$ be a property type. Then $\langle f, color \rangle$ 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 $\mathsf{ValAss} : \mathcal{FL} \to \mathcal{VL}$ 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 (#FF0000) 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: $\mathsf{ValAss}(\langle \text{mug}, \text{drinking device}\rangle) = 20cc$. To be able to make this observation, the value $20cc$ must be in $\mathcal{VL}$ 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)

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

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

The fact that devising such measuring method is a problem in itself. In [Ald02]. 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, 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 a system be, then? Note that, in essence, there are two ways of measuring systems:

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.

3.2 Quality & 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 [BGP+05]: 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., [KG03, Som89, Bev99]
- Search on the web: searchers must try to specify their information need. See e.g., [BBWW98, Gro00, HPW96]

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. [DB04]). 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 liters. 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 $\mathcal{RQ}$ be the set of all requirements and $\mathcal{CS}$ be the set of all constraint operators[4]. 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 by operators and possibly some logical connectors. Consider example, the expression $P(x) \ \wedge \ Q(x, y)$. This expression has a unary binary 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: $\mathcal{EX} \triangleq \mathcal{VL} \cup \mathsf{ValAss}$[5] denotes the set of all expressions. Let $\mathsf{Prop} : \mathcal{RQ} \to \mathcal{PT}$, $\mathsf{Constr} : \mathcal{RQ} \to \mathcal{CS}$, and $\mathsf{Expr} : \mathcal{RQ} \rightarrowtail \mathcal{EX}$. We introduce the following shorthand notation:

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

The previous examples can now be written more formally as:

Example 3.7

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

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

- 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 Poperty Type "Price" by Constraint Operator "may not exceed" is the Value of Artifact "c" with respect to Property Type "price"

Figure 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 $\mathcal{AC}$ be the set of actors and $\mathsf{Req} : \mathcal{AC} \times \mathcal{FL} \to \wp(\mathcal{RQ})$ denote the requirements of an actor with regard to a fulfillment. For example:

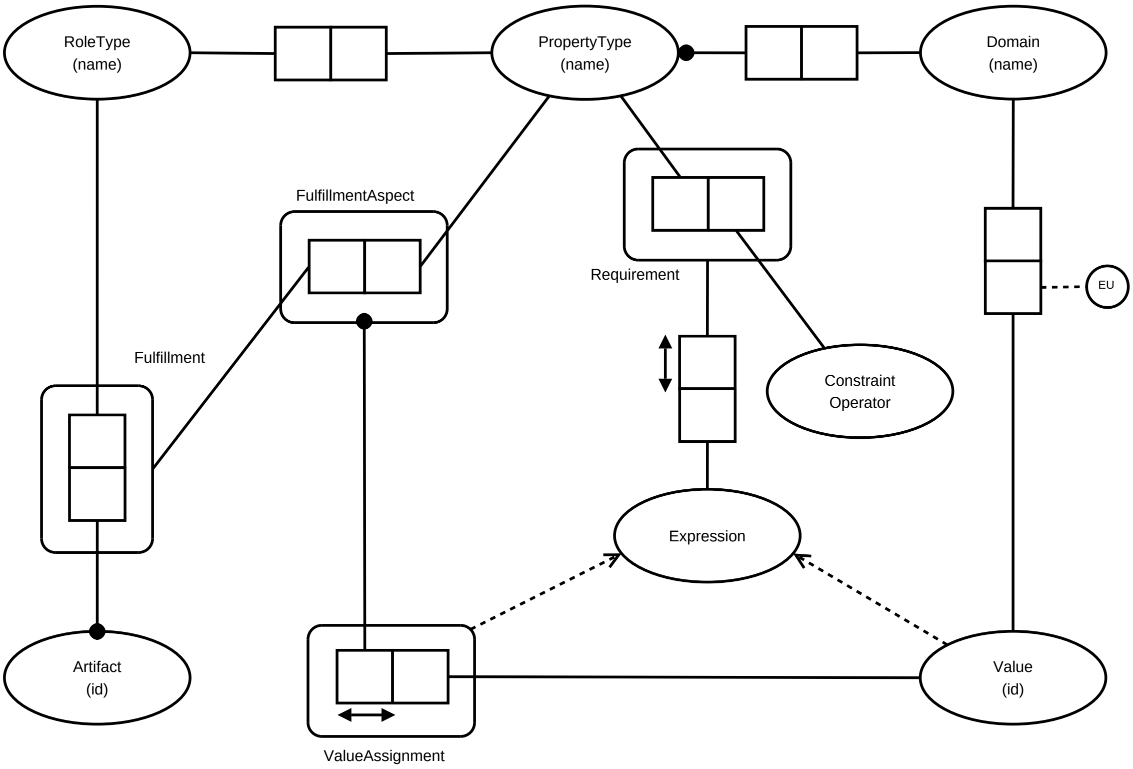


Figure 3: Requirements & constraints

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

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 assesment 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***	***price***
m_1	*red*	$20cc$	*€3*
m_2	*red*	$25cc$	*€3*
m_3	*blue*	$25cc$	*€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* $r_2 = \langle \textit{color}, =, \textit{red} \rangle$, $r_3 = \langle \textit{volume}, \geq, \textit{25cc} \rangle$ *and* $r_1 = \langle \textit{price}, \leq, e3 \rangle$. *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.*

Literature suggests numerous ways to deal with these kinds of selection/ optimization problems such as Operations Research [KA97, Tah92] and multi-objective decision making [Diw03, Bom95]. 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 (I.e., [GPB04, GPBW05]) 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 is summarized in Figure 4. We will firstly present a short formalization of our model. After that we will illustrates its use by means of a small example.

4.1.1 Formalization

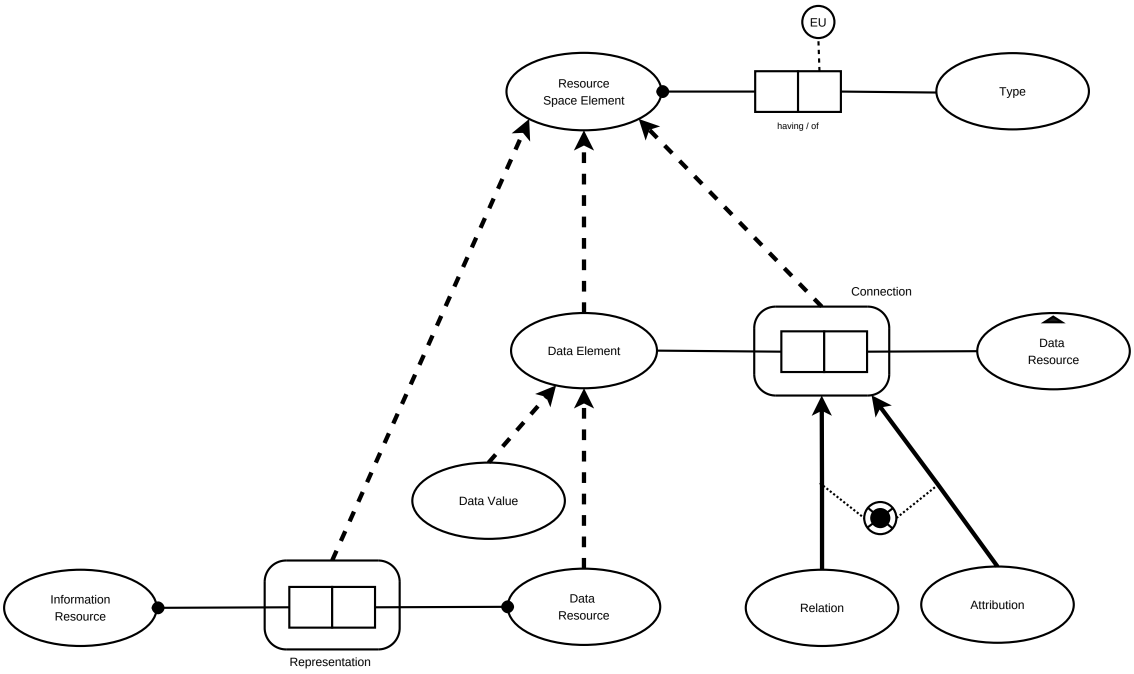


Figure 4: A reference model for information supply

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 [BL94]. Data resources can be a lot of 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. [LS99]) 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 [Bus45, Con87] 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) $\mathcal{IR}, \mathcal{DR}, \mathcal{RP}, \mathcal{DV}, \mathcal{RL}$ 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: $\mathcal{CN} \triangleq \mathcal{AT} \cup \mathcal{RL}$. 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:

$$\begin{array}{rcl} s \overset{c}{\rightsquigarrow} d & \triangleq & \mathsf{Src}(c) = s \wedge \mathsf{Dst}(c) = d \\ s \rightsquigarrow d & \triangleq & \exists_c \, [s \overset{c}{\rightsquigarrow} d] \end{array}$$

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

Axiom 4 (Relations) $r \in \mathcal{RL} \implies \mathsf{Dst}(r) \in \mathcal{DR}$

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

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 surjective 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 $\mathcal{TP}$ be the set of all types and $\mathcal{RE} \triangleq \mathcal{DE} \cup \mathcal{CN} \cup \mathcal{RP}$ be the resource space elements that form the basis for the typing mechanism; then $\mathsf{HasType} \subseteq \mathcal{RE} \times \mathcal{TP}$ denotes the relation for typing. Observe that a $t \in \mathcal{TP}$ 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:

$$\begin{array}{lcllcl} \pi(t) & \triangleq & \{e \mid e \,\mathsf{HasType}\, t\} & \tau(t) & \triangleq & \{t \mid e \,\mathsf{HasType}\, t\} \\ \pi(T) & \triangleq & \bigcup_{t \in T} \pi(t) & \tau(E) & \triangleq & \bigcup_{e \in E} \tau(e) \end{array}$$

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 $X \subseteq \mathcal{RE}$, in particular one of the basic sets such as $\mathcal{RP}$ or $\mathcal{DR}$, 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) $\tau(e) \neq \varnothing$

Axiom 9 (Existential typing) $\pi(t) \neq \varnothing$

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

Axiom 10 (Equal types) $\pi(s) = \pi(t) \implies s = t$

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

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

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 sub-tying in the example):

$$\begin{array}{lll} \mathit{davinci.html} & \mathsf{HasType} & \mathit{Html} \\ \mathit{monalisa.eps} & \mathsf{HasType} & \mathit{Eps} \end{array}$$

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

$$\begin{array}{llllll} \mathsf{IRes}(r_1) = \mathit{Leonaro\ DaVinci} & \text{and} & \mathsf{DRes}(r_1) = \mathit{davinci.html} & \text{and} & r_1 \,\mathsf{HasType}\, \mathit{Website\ about} \\ \mathsf{IRes}(r_2) = \mathit{The\ Mona\ Lisa} & \text{and} & \mathsf{DRes}(r_2) = \mathit{davinci.html} & \text{and} & r_2 \,\mathsf{HasType}\, \mathit{Website\ about} \\ \mathsf{IRes}(r_3) = \mathit{The\ Mona\ Lisa} & \text{and} & \mathsf{DRes}(r_3) = \mathit{monalisa.eps} & \text{and} & r_3 \,\mathsf{HasType}\, \mathit{Picture\ of} \end{array}$$

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

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

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

$$\begin{array}{ll} \textit{monalisa.eps} \overset{a_1}{\rightsquigarrow} \textit{1024}\times\textit{768} & a_1\ \mathsf{HasType}\ \textit{resolution} \\ & \textit{1024}\times\textit{768}\ \mathsf{HasType}\ \textit{ResolutionString} \\ \textit{monalisa.eps} \overset{a_2}{\rightsquigarrow} \textit{24-06-2003,10:12} & a_2\ \mathsf{HasType}\ \textit{creation date} \\ & \textit{24-06-2003,10:12}\ \mathsf{HasType}\ \textit{DateString} \\ \textit{davinci.html} \overset{a_3}{\rightsquigarrow} \textit{24-06-2003,16:45} & a_3\ \mathsf{HasType}\ \textit{modification date} \\ & \textit{24-06-2003,16:45}\ \mathsf{HasType}\ \textit{DateString} \end{array}$$

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:

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

Last but not least, the populations of the generalizations $\mathcal{DE}$, $\mathcal{CN}$ and $\mathcal{RE}$ 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.

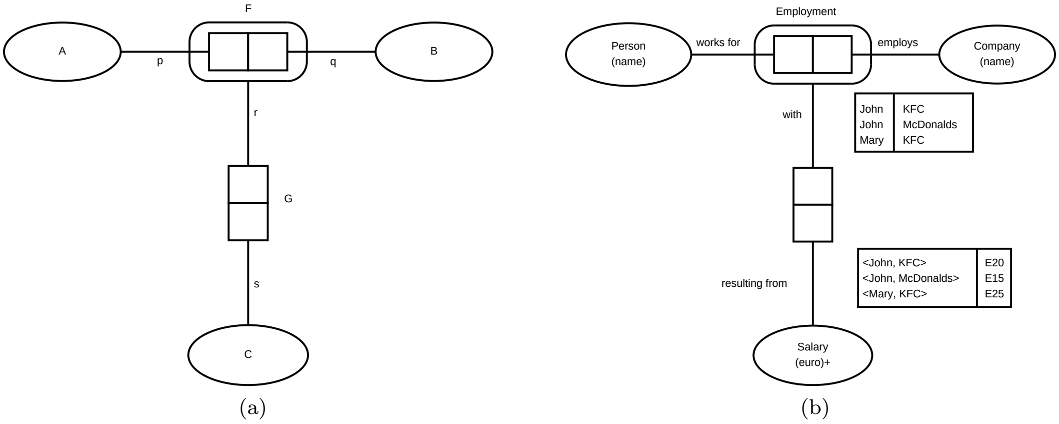
4.2 Language

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 [HW93, HPW93, PW95].

4.2.1 PSM & LISA-D

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

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



John	KFC
John	McDonalds
Mary	KFC

<John, KFC>	E20
<John, McDonalds>	E15
<Mary, KFC>	E25

Figure 5: Example: information structure and names

that the Fact relation is derivable, it is defined as follows:

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

- 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 [HW93].

An information structure such as Figure 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 $\mathcal{I}$ is the assignment of sets of instances to the object types in $\mathcal{O}$; $\mathsf{Pop} : \mathcal{O} \to \wp(\Omega)$, 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. Ofen, an orderning 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 ($\mathcal{PE}$) 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 ($\mathcal{O}$) and predicators ($\mathcal{P}$). Let $\mu : \mathcal{PE} \to \Omega$ 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:

$$\begin{aligned} N \circ M &\triangleq \lambda \langle x, y \rangle . \bigcup_{a \in X} N(x,a) \times M(a,y) \\ N^{\leftarrow} &\triangleq \lambda \langle x, y \rangle . N(y,x) \end{aligned}$$

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	$\varnothing$	$\mu [\![\varnothing]\!] = \varnothing$
a constant	c	$\mu [\![c]\!] = \{\!\{ c, c \}\!\}$
multiset	X	$\mu [\![X]\!] = \{\!\{ \langle x, x \rangle \uparrow^1 \mid x \in X \}\!\}$
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 $\not{f}\ P$ 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 $\mathcal{N}$ be the set of all names.
- Object types are referenced by a unique name: $\mathsf{ONm} : \mathcal{O} \rightarrowtail \mathcal{N}$.
- Predicators are referenced by a unique name:$\mathsf{PNm} : \mathcal{P} \rightarrowtail \mathcal{N}$.
- Role names correspond to special connections (in the form of path expressions) through (binary) fact types: $\mathsf{RNm} : \mathcal{P} \rightarrowtail \mathcal{N}$.

The actual naming is administered by the function $\mathsf{Path} : \mathcal{O} \times \mathcal{O} \times \mathcal{N} \rightarrowtail \mathcal{PE}$ that assigns, in a given context, a path expressions to a name. For optimization urposes, beginning and 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 $\mathbb{D} : \mathcal{N} \rightarrow \mathcal{PE}$ translates information descriptors to paths. The lexicon Path contains all atomic information descriptiors:

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

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-orriented (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:

$$\mathbb{D}[\![\textsf{P AND-ALSO Q}]\!] = \fint \mathbb{D}[\![P]\!] \cap \fint \mathbb{D}[\![Q]\!]$$

$$\mathbb{D}[\![\textsf{P OR-ELSE Q}]\!] = \fint \mathbb{D}[\![P]\!] \cup \fint \mathbb{D}[\![Q]\!]$$

$$\mathbb{D}[\![\textsf{P BUT-NOT Q}]\!] = \fint \mathbb{D}[\![P]\!] - \fint \mathbb{D}[\![Q]\!]$$

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

- $\textsf{Path}(A, A, \textsf{Person}) = A$
- $\textsf{Path}(A, B, \textsf{works for}) = p \circ q^{\leftarrow}$
- $\textsf{Path}(B, A, \textsf{employs}) = q \circ p^{\leftarrow}$
- $\textsf{Path}(A, F, \textsf{having}) = p$
- $\textsf{Path}(F, A, \textsf{of}) = p^{\leftarrow}$
- $\textsf{Path}(*, *, \text{``}\textsf{KFC}\text{''}) = \text{``KFC''}$

Observe that Figure 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:

$$\begin{aligned}
&\mathbb{D}[\![\textsf{Person works for ``KFC''}]\!] = \\
&\mathbb{D}[\![\textsf{Person}]\!] \circ \mathbb{D}[\![\textsf{works for}]\!] \circ \mathbb{D}[\![\textsf{``KFC''}]\!] = \\
&A \circ p \circ q^{\leftarrow} \circ \text{``KFC''}
\end{aligned}$$

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

$$\begin{aligned}
&\mu[\![A \circ p \circ q^{\leftarrow} \circ \text{``KFC''}]\!] = \\
&\mu[\![A]\!] \circ \mu[\![p]\!] \circ \mu[\![q^{\leftarrow}]\!] \circ \mu[\![\text{``KFC''}]\!] = \\
&\mu[\![p]\!] \circ \mu[\![q^{\leftarrow}]\!] \circ \mu[\![\text{``KFC''}]\!]
\end{aligned}$$

In the remainde we will use quoted names to rever 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 Figure 6:

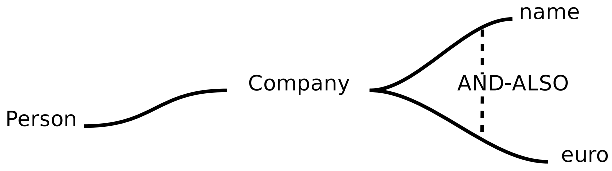


Figure 6: Example path

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

The expressions $\mu\left[\!\left[q^{\leftarrow} \circ \text{“KFC”} \right]\!\right]$ and $\mu\left[\!\left[r \circ s^{\leftarrow} \circ \text{“20”} \right]\!\right]$ 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 $\mu\left[\!\left[p \right]\!\right]$ we get the answer to the query which is:

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

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 Figure 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 Figure 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

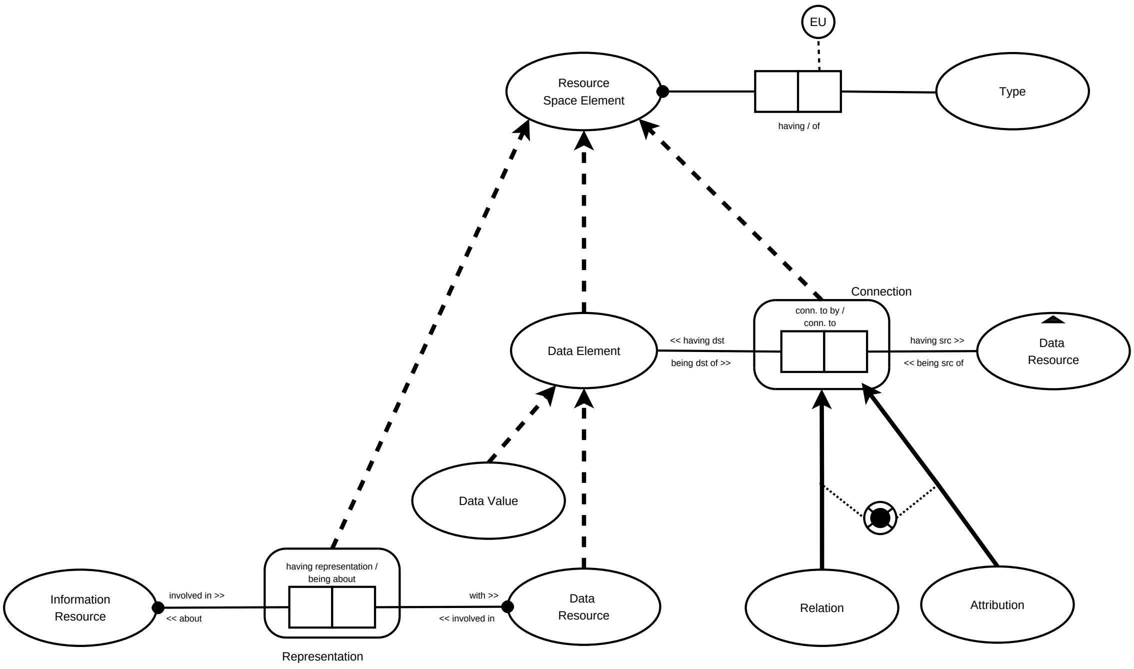


Figure 7: A reference model for information supply

- 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 $\triangleq$ being src of and with value $\triangleq$ 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 representaton ( 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 contraints 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 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 very well 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 Figure 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, the quality assessment system has to somehow combine the ‘hard’ (often numberical) 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 [Zad75a, Zad75b, Zad75c, Zad02]. 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.

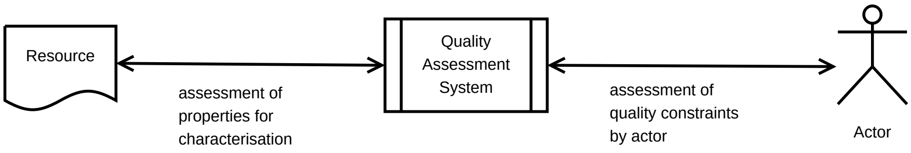

Resource
Quality Assessment System
assessment of properties for characterisation
assessment of quality constraints by actor
Actor

Figure 8: Uncertainty in quality assessment

5.1 Variable

In this section we will present the concept of a linguistic variable based on the work of Zadeh. Wikipedia[6] 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 [Zad75a] the following formal definition of a variable is presented: A variable is characterize 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:

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

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 $R(X_1, X_2)$:

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

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 characterize by a membership function $\mu_R : U \to [0, 1]$ which represents the grade of membership with respect to the fuzzy restriction. Figure 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.

[6] `http://www.wikipedia.org`

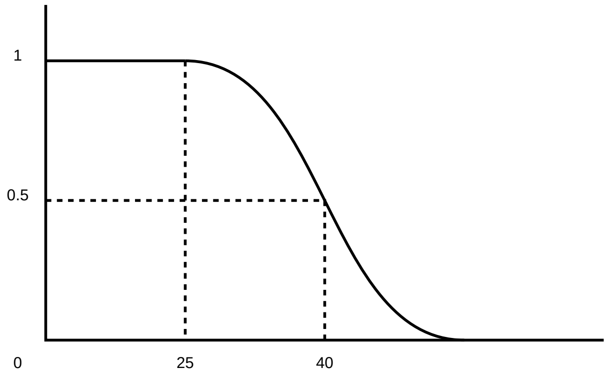


Figure 9: Membership function for *young*

More formally, a linguistic is characterize by a quintuple $\langle \mathcal{X}, T(\mathcal{X}), U, G, M \rangle$ in which $\mathcal{X}$ is the name of the variable; $T(\mathcal{X})$ (or simply T) denotes the term-set of $\mathcal{X}$, 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 $\mathcal{X}$ and M is a semantic rule for associating with each X its meaning $M(X)$.

Continuing the the previous example, let $\mathcal{X} = age$ be a linguistic variable with $U = [0, 100]$. I.e. we assume that people do not get older than 100 years. In this case *young* is considered to be a linguistic value of $\mathcal{X}$. More specifically, if $T(\mathcal{X}) = \{young, medium\ age, old\}$ then Figure 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*.

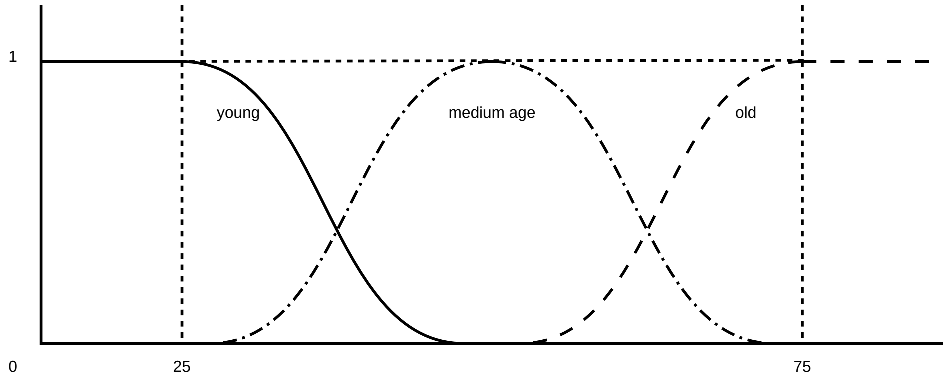


Figure 10: The linguistic variable *Age*

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

$$\begin{array}{lcl} T & \rightarrow & young \\ T & \rightarrow & very\ T \end{array}$$

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 [Zad75b] for the variable *young* is as follows:

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

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

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

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 accurrate 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 $\mathcal{SI}$ to be the set of all possible situations and $\mathcal{MD}$ 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 $\mathcal{MV}$ be the union of all possbible value domains. A measuring device $R \in \mathcal{MD}$ can now be modeled as a function that maps object-situation combinations into values:

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

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* 125km/h *is expressed as:* $\mathsf{M}(c, s, d) = 125\text{km/h}$

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 determing 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 eactly at* 125km/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 $\circeq$ to be an operator that measures if a measurement is sufficiently equal to a standard measurement[7]. In other words, a measurement is accurate (sufficiently equal to a standard measurement) if $\mathsf{M}(c, s, d) \circeq \mathsf{M}(c, s, d_s)$.

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

[7] In a more elaborate theory it would be interesting to parameterize the $\circeq$ to be able to specify the allowable deviation. This is, however, beyond the score of this paper.

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 neccesarily all) situations $S \subseteq \mathcal{SI}$. The accuracy of d is defined to be the average deviation of that device with respect to the situations in which it is tested:

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

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 $\langle \mathcal{X}, T(\mathcal{X}), U, G, M \rangle$ be a linguistic variable. In the running example for this section, $\mathcal{X}$ represents the variable *volume of a mug* with termset $T(\mathcal{X}) = \{\text{big}, \text{medium}, \text{small}\}$. We interpret the membershipdegree for these linguistic values as the degree of certainty that we have in this specific interpretation of the actual measurement. Let $\mu_t : U \rightarrow [0 \ldots 1]$ denote the membership degree for the terms t in the termset. To set the stage, consider the following running example:

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

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

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

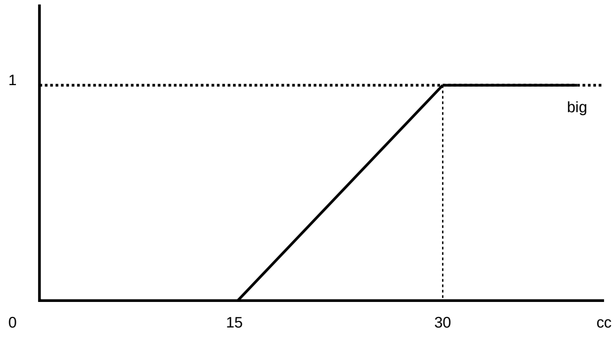


Figure 11: Probability profile for linguistic value 'big'

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

> Suppose I measure the volume of a mug to be $25cc$. 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 cummulative probability distribution, provided that for each linguistic value v it holds that

Axiom 12 $\int\limits_0^\infty \mu_v(u)du = 1$

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

$$P_v^i(u) = \int\limits_0^u \mu_v(u)du$$

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'.

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

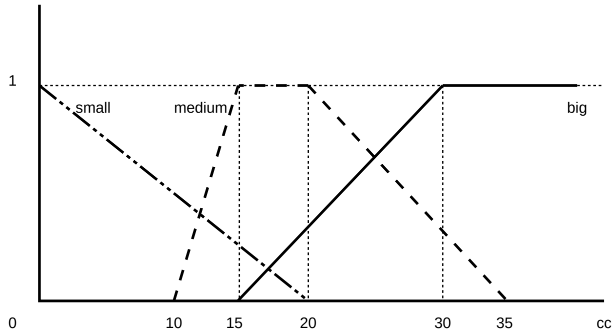


Figure 12: Probability profile for the values of the linguistic variable 'volume'

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:*

$$P_s^i(u) = \begin{cases} 1 - \frac{1}{20}u & u \leq 20 \\ 0 & otherwise \end{cases}$$

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

respectively. The membership functions are illustrated in Figure 12. It is easy to verify that:

- *the certaintly that measured volume is indeed interpreted as 'big': $P_b^i(25) = 0.67$,*
- *the certaintly that measured volume is indeed interpreted as 'medium': $P_s^i(25) = 0.67$*

- *the certaintly 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:* $0.9 \times 0.67 = 0.6$.

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 assined 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: $n \in \mathcal{AF}$ denote the newsletter and $r \in \mathcal{RO}$ 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 $a \in \mathcal{AC}$. 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 resouce 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 $d_1, d_2, d_3 \in \mathcal{MD}$, one for each constraint. The three measurements will be done in parallel; in other words, in one situation $s \in \mathcal{SI}$. Based on previous experiences and tests the system knowst hat:

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 importances of artitfacts 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 comples. The termset for this variable is $\{low, average, high\}$ and the underlying domain $U = [0 \dots 10]$ 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':

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

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.

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, bundeling 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 archtecture for transformations on the Web (e.g., [GPB04, GPBW04, GPBV05, GPBW05]. 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 systms that transform data resources (of a certain type) into other data resources. Let $\mathcal{TR}$ 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:

$$\mathsf{SEM} : \mathcal{TR} \rightarrow (\mathcal{DR} \rightarrow \mathcal{DR})$$

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 $\mathsf{Input}, \mathsf{Output} : \mathcal{TR} \rightarrow \tau(\mathcal{DR})$. 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) $\overrightarrow{T_1} = \overrightarrow{T_2} \implies T_1 = T_2$

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) $e \in \mathsf{Input}(T) \implies \overrightarrow{T}(e) \in \mathsf{Output}(T)$

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

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:

$$\overrightarrow{T}(E) \;\triangleq\; \left\{ \overrightarrow{T}(e) \;\middle|\; e \in E \right\}$$

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 $\mathcal{TR}$ 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:

As 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 $e \in \mathcal{DR}$ 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:

$r_1 = \langle p_1, high \rangle$ 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

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

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

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:

$$\mathsf{M}(e, p_1, d_1) = 7 \text{ such that } \mu_{p_1,high}(7) = \tfrac{2}{5}$$
$$\mathsf{M}(e, p_2, d_2) = 8 \text{ such that } \mu_{p_2,high}(8) = \tfrac{8}{15}$$
$$P_{r_1} = 1 \times \tfrac{2}{5} = \tfrac{2}{5}$$
$$P_{r_2} = 1 \times \tfrac{8}{15} = \tfrac{8}{15}$$
$$\text{Aptness } = \tfrac{2}{5} \times \tfrac{8}{15} = \tfrac{16}{75} \approx 0.213$$

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

$$\mathsf{M}(\overrightarrow{T_1}(e), p_1, d_1) = 10 \text{ such that } \mu_{p_1,high}(10) = 1$$
$$\mathsf{M}(\overrightarrow{T_1}(e), p_2, d_2) = 2 \text{ such that } \mu_{p_2,high}(2) = \tfrac{2}{15}$$
$$P_{r_1} = 1 \times 1 = 1$$
$$P_{r_2} = 1 \times \tfrac{2}{15} = \tfrac{2}{15}$$
$$\text{Aptness } = \tfrac{2}{15} \approx 0.133$$

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:

$$\begin{aligned}
&\mathsf{M}(\overrightarrow{T_2}(e), p_1, d_1) = 8 \;\text{ such that }\; \mu_{p_1,high}(8) = \tfrac{3}{5}\\
&\mathsf{M}(\overrightarrow{T_2}(e), p_2, d_2) = 10 \;\text{ such that }\; \mu_{p_2,high}(10) = \tfrac{2}{3}\\
&P_{r_1} = 1 \times \tfrac{3}{5} = \tfrac{3}{5}\\
&P_{r_2} = 1 \times \tfrac{2}{3} = \tfrac{2}{3}\\
&\text{Aptness } = \tfrac{3}{5} \times \tfrac{2}{3} = \tfrac{2}{5} = 0.4
\end{aligned}$$

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, our research goal was:

> 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 (attriutes) 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 aftifact 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 'jjust 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 uzing 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 aftifacts (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 somehwat 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 likelyhood that the property support of the aftifact is conform the desires of the actor, taking measurement and interpretation uncertainty into account.

References

[Ald02] K. Alder. *The measure of all things: The Seven-Year Odyssey and Hidden Error That Transformed the World.* Free Press, New York, NY, USA, 2002. ISBN 074321675X

[ATK97] A.T. Arampatzis, T. Tsoris, and C.H.A. Koster. Irena: Information retrieval engine based on natural language analysis. In *Proceedings of the RIAO'97 Conference*, pages 159–175, 1997.

[ATKW98] A.T. Arampatzis, T. Tsoris, C.H.A. Koster, and Th.P. van der Weide. Phrase-based information retrieval. *Information Processing & Management*, 34(6):693–707, December 1998.

[AWKB00] A.T. Arampatzis, Th.P. van der Weide, C.H.A. Koster, and P. van Bommel. *Linguistically-motivated Information Retrieval*, volume 69, pages 201–222. Marcel Dekker, New York, NY, USA, 2000.

[BBWW98] F.J.M. Bosman, P.D. Bruza, Th.P. van der Weide, and L.V.M. Weusten. Documentation, cataloging and query by navigation: A practical and sound approach. In C. Nikolaou and C. Stephanidis, editors, *Research and Advanced Technology for Digital Libraries, 2nd European Conference on Digital Libraries '98, ECDL '98*, volume 1513 of *Lecture Notes in Computer Science*, pages 459–478, Berlin, Germany, EU, September 1998. Springer.

[Bev99] Nigel Bevan. Quality in use: meeting user needs for quality. *Journal of System and Software*, 49(1):89–96, 1999.

[BGP$^+$05] P. van Bommel, B. van Gils, H.A. (Erik) Proper, M. van Vliet, and 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, editors, *Web Information Systems Engineering (WISE'05)*, volume 3806 of *Lecture Notes in Computer Science*, pages 577–583, Berlin, Germany, EU, November 2005. Springer-Verlag. ISBN 3540300171
doi:10.1007/11581062_50

[BJ87] F.P. Brooks Jr. No silver bullet: essence and accidents of software engineering. *IEEE Computer*, 20(4):10–19, April 1987.

[BL94] T. Berners-Lee. Universal resource identifiers in www. Technical Report RFC1630, IETF Network Working Group, June 1994.

[Bom95] P. van Bommel. *Database Optimization: An Evolutionary Approach.* PhD thesis, University of Nijmegen, The Netherlands, EU, 1995. ISBN 9090082441

[Bru90] P.D. Bruza. Hyperindices: A novel aid for searching in hypermedia. In A. Rizk, N. Streitz, and J. Andre, editors, *Hypertext: Concepts, Systems and Applications; Proceedings of the European Conference on Hypertext (ECHT '90)*, number 5 in Cambridge Series on Electronic Publishing, pages 109–122, 1990. ISBN 0521405173

[Bus45] V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, Jul 1945.

[BW90] P.D. Bruza and Th.P. van der Weide. Two level hypermedia - an improved architecture for hypertext. In A.M. Tjoa and R.R. Wagner, editors, *Proceedings of the Data Base and Expert System Applications Conference (DEXA 90)*, pages 76–83, Berlin, Germany, EU, 1990. Springer. ISBN 3211822348

[BW92] P.D. Bruza and Th.P. van der Weide. Stratified hypermedia structures for information disclosure. *The Computer Journal*, 35(3):208–220, 1992.

[Cle91] C.W. Cleverdon. The significance of the cranfield tests on index languages. In A. Bookstein, Y. Chiarmarella, G.E Salton, and V.V. Raghavan, editors, *Proceedings of the 14th Annual ACM Conference of Research and Development in Information Retrieval (SIGIR'1991)*, pages 3–12, New York, NY, USA, October 1991. ACM Press. ISBN 0897914481

[Con87] J. Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20(9):17–41, September 1987.

[DB04] P Donzelli and B. Bresciani. Improving requirements engineering by quality modelling – a quality-based requirements engineering framework. *Journal of Research and Practice in Information Technology*, 36(4), November 2004.

[DES05] *Quality Selection Criteria for Subject Gateways*, 2005. Last checked: 27-Oct-2005. `http://www.sosig.ac.uk/desire/qindex.html`

[Diw03] U. Diwekar. *Introduction to Applied Optimization*, volume 80 of *Applied Optimization.* Springer, Berlin, Germany, EU, 2003. ISBN 1402074565

[DO85] G.B. Davis and M.H. Olson. *Management Information Systems: Conceptual Foundations, Structure and Development.* McGraw–Hill, New York, New York, USA, 1985.

[Eus] J. Eustace. Descartes' definition of matter. First published in The Journal of the Limerick Philosophical Society in 1987. last checked: 02-Feb-2006.
`http://www.ul.ie/~philos/vol1/eustac1.html`

[Gil88] T. Gilb. *Principles of software engineering management.* Addison Wesley, Reading, Massachusetts, USA, 1988.

[GÖSS04] Michael Gertz, M. Tamer Özsu, Gunter Saake, and Kai-Uwe Sattler. Report on the dagstuhl seminar: data quality on the web. *SIGMOD Rec.*, 33(1):127–132, 2004. ISSN 0163-5808

[GPB04] B. van Gils, H.A. (Erik) Proper, and P. van Bommel. A conceptual model of information supply. *Data & Knowledge Engineering*, 51:189–222, 2004.

[GPBV05] B. van Gils, H.A. (Erik) Proper, P. van Bommel, and P. de Vrieze. Transformation selection for aptness–based web retrieval. In H.E. Williams and G. Dobbie, editors, *Proceedings of the Sixteenth Australasian Database Conference (ADC2005)*, volume 39, pages 115–124, Sydney, New South Wales, Australia, January 2005. Australian Computer Society. ISBN 192068221X

[GPBW04] B. van Gils, H.A. (Erik) Proper, P. van Bommel, and Th.P. van der Weide. Transformations in information supply. In J. Grundspenkis and M. Kirikova, editors, *Proceedings of the Workshop on Web Information Systems Modelling (WISM'04), held in conjunctiun with the 16th Conference on Advanced Information Systems Engineering*, volume 3, pages 60–78, June 2004. ISBN 9984976718

[GPBW05] 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.

[Gro00] F.A. Grootjen. Employing semantical issues in syntactical navigation. In *Proceedings of the 22nd BCS–IRSG Colloquium on IR Research*, pages 22–33, 2000.

[Gro01] F.A. Grootjen. Indexing using a grammerless parser. In *2001 IEEE International Conference on Systems, Man & Cybernetics (SMC2001)*, 2001. ISBN 0780370899

[Hal01] T.A. Halpin. *Information Modeling and Relational Databases, From Conceptual Analysis to Logical Design.* Morgan Kaufmann, San Mateo, California, USA, 2001. ISBN 1558606726

[Har96] M. Harrison. *Principles of operations management.* Pitman, London, UK, EU, 1996. ISBN 0273614509

[HC05] P. Hernon and P. Calvert. E-service quality in libraries: Exploring its features and dimensions. *Library & Information Science Research*, 27(3):377–404, 2005.

[HPW93] A.H.M. ter Hofstede, H.A. (Erik) Proper, and Th.P. van der Weide. Formal definition of a conceptual language for the description and manipulation of information models. *Information Systems*, 18(7):489–523, October 1993.

[HPW96] A.H.M. ter Hofstede, H.A. (Erik) Proper, and Th.P. van der Weide. Query formulation as an information retrieval problem. *The Computer Journal*, 39(4):255–274, September 1996.

[HW93] A.H.M. ter Hofstede and Th.P. van der Weide. Expressiveness in conceptual data modelling. *Data & Knowledge Engineering*, 10(1):65–100, February 1993.

[IEP06] Aristotle (384-322 bce): General introduction. The Internet Encyclopedia of Philosophy, 2006. last checked: 02-Feb-2006.
`http://www.utm.edu/research/iep/a/aristotl.htm`

[KA97] Soung-Hie Kim and Byeong-Seok Ahn. Group decision making procedure considering preference strenght under incomplete information. *Computers & Operations Research*, 24(12):1101–1112, 1997.

[KG03] D. Kulak and E. Guiney. *Use Cases: Requirements in Context.* Addison Wesley, Reading, Massachusetts, USA, 2nd edition, 2003. ASIN 0201657678

[LASG02] V. Lala, A. Arnold, S.G. Sutten, and 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):237–253, December 2002.

[LL96] K. C. Laudon and J. P. Laudon. *Management Information Systems, International Edition.* Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1996. ISBN 0132328852

[LS99] Ora Lassila and Ralph R. Swick. Resource description framework (rdf) model and syntax specification. Technical report, W3C, February 1999.
`http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`

[McC04] Steve McConnell. *Code complete, a practical handbook of software construction.* Microsoft Press, Redmond, Washington, USA, 2 edition, 2004.

[Orr98] K. Orr. Data quality and systems theory. *Communications of the ACM*, 41(2):66–71, 1998. ISSN 00010782

[Pij94] 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*, pages 165–172, New York, NY, USA, 1994. ACM Press. ISBN 0897916522

[PPY01] M.P. Papazoglou, H.A. (Erik) Proper, and J. Yang. Landscaping the information space of large multi-database networks. *Data & Knowledge Engineering*, 36(3):251–281, 2001.

[PW95] H.A. (Erik) Proper and Th.P. van der Weide. Information disclosure in evolving information systems: Taking a shot at a moving target. *Data & Knowledge Engineering*, 15:135–168, 1995.

[SFG$^+$00] J.J. Sarbo, J.I. Farkas, F.A. Grootjen, P. van Bommel, and Th.P. van der Weide. Meaning extraction from a peircean perspective. *International Journal of Computing Anticipatory Systems*, 6:209–227, 2000.

[Som89] I. Sommerville. *Software Engineering.* Addison Wesley, Reading, Massachusetts, USA, 1989.

[Tah92] H.A. Taha. *Operations Research, an introduction.* Prentice–Hall, Englewood Cliffs, New Jersey, USA, 4 edition, 1992. ISBN 0131876597

[TLKC99] E. Turban, J. Lee, D. King, and H.M. Chung. *Electronic Commerce, a managerial perspective.* Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1999. ISBN 0139752854

[VW99] C. Vishik and A.B. Whinston. Knowledge sharing, quality, and intermediation. In *Proceedings of the international joint conference on Work activities coordination and collaboration*, pages 157–166, New York, NY, USA, 1999. ACM Press. ISBN 1581130708

[WGMD95] S. Weibel, J. Godby, E. Miller, and R. Daniel. Metadata workshop report. Dublin, Ohio, USA, March 1995.

[Zad75a] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning – i. *Information Science*, 8:199–249, 1975.

[Zad75b] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning – ii. *Information Science*, 8:301–357, 1975.

[Zad75c] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning – iii. *Information Science*, 9:301–357, 1975.

[Zad02] L. 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*, 12:307–324, 2002.