Abstract

A requirement has been variously defined as a purpose, a need, a goal, a function(ality), a constraint, a behavior, a service, a condition, or a capability. Limited effort has been put into making explicit the assumptions and choices behind the various available definitions. In contrast to the available definitions, we propose one given in a restricted vocabulary of a foundational ontology. Assumptions and philosophical choices are discussed along with implications. We study various closely related concepts including ‘environment’, ‘system’, ‘machine’, ‘environment assumption’, along with the common taxonomic categories of the requirement concept. We argue for the introduction of new notions, namely those of ‘preference’ and ‘priority’. The various interwoven definitions build up to an ontology of core concepts in requirements engineering. The resulting, so-called CORE ontology, calls for a revision of the accepted understanding of the requirements problem. We outline the revised requirements problem and contrast it to the current variant.

Status: Started on May 30, 2007; initial version finished in June 2007; first revision in July 2007; second revision in August 2007. This is the second complete draft.
1 Introduction

1.1 Motivation

A decade ago Zave and Jackson [77] observed that the field of requirements engineering (RE) left behind simplistic approaches to understanding what a system will do in favor of novel and varied terminology, methods, languages, tools, and issues considered to be critical. Despite continued progress, one particular area of RE remains weak at present—namely, there is limited consensus as to the precise meaning of the basic terminology used in RE. It does not require considerable effort to see that the field richly and variously defines the requirement concept. For instance, Ross and Schoman’s [57] early definition of RE points out that requirements describe the purpose of a system:

“Requirements definition must say why a system is needed, based on current or foreseen conditions, which may be internal operations or an external market. It must say what system features will serve and satisfy this context. And it must say how the system is to be constructed.”

In a survey of research efforts in RE, Zave [76] elaborates:

“Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.”

Letier and van Lamsweerde [39] concur while placing emphasis on the critical role of system goals:

“Requirements engineering is concerned with the identification of the goals to be achieved by the system-to-be, the operationalization of such goals into specifications of services and constraints, and the assignment of responsibilities for such services and constraints among human, physical, and software components forming the system.”
A different terminology is used in the IEEE Guide to the Software Engineering Body of Knowledge [24] and Kotonya and Sommerville’s textbook on RE [36]:

“[Requirements engineering] is concerned with the elicitation, analysis, specification, and validation of software requirements. [...] Software requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem.”

The above agrees with Goguen and Linde [18] that requirements express needs:

“A basic question in Requirements Engineering is how to find out what users really need.”

To go beyond needs, one might consider the IEEE 610 standard on Software Engineering Terms [23] and read:

“[A requirement is]: (1) a condition or capability needed by a user to solve a problem or achieve an objective; (2) a condition or capability that must be met or possessed by a system component to satisfy a contract, standard, specification or other formally imposed documents; (3) a documented representation of a condition or capability as in (1) or (2).”

The requirement concept has evidently been variously defined as a purpose, a need, a goal, a function(ality), a constraint, a behavior, a service, a condition, or a capability. Surveying more definitions would give same or other interpretations, but the point remains the same: the requirement concept is richly interpreted and variously used. The problem here lies not in the need for rich conceptual foundations, but in the fact that there actually are no common conceptual foundations on which the various efforts can be compared and benefits of their accumulation and combination reaped. Definitions above are hardly synonymous.

It is undoubtedly of clear interest to aim for as precise as feasible delimitation and definition of the conceptual fundamentals in RE. Experience shows that this is favorable to rigor in both research and practice. One example is goal-oriented RE where accepting that a requirement is a goal led to the accumulation of various compatible approaches to analyzing requirements (see, e.g., [48, 10, 73, 2, 74, 56, 6, 38, 5, 17, 39] and [69, 70] for overviews).

1.2 Response

Any discussion and redefinition of a concept so widely and variously used as that of ‘requirement’ meets scepticism. As the above cited definitions illustrate, one concept is defined by mapping to other concepts. Broadly speaking, these other concepts are hopefully more familiar and precise in the reader’s vocabulary. It is then hoped that the more these other concepts fit the reader’s intuition, the more relevant in her view the proposed definition. This common approach is avoided here for it is much too open for unstructured criticism: meanings of these other concepts remain in most cases open and the abstract choices behind them untold.

More precision can nevertheless be achieved. The approach adopted herein consists of using a restricted vocabulary in which the above mentioned ‘other’ concepts are given and their meaning already sharply delimited. In contrast to the available definitions, we thus propose one given in a restricted vocabulary of a foundational ontology. Our worldview is open to debate, but at least is explicit as are therefore the critical assumptions that underlie the definitions. In other words, we define the fundamentals of RE by mapping them to other fundamentals, the latter being at a higher level of abstraction and all resting in well-delimited categories. Assumptions and philosophical choices can therefore be discussed along with their implications.

1.3 Summary of Contributions

All definitions in CORE are mappings to concepts and relations in a foundational ontology obtained by combining four components:

1. We first require an established foundational ontology which tells us at a sufficiently fine grained level what kind of entities there are in reality. The main purposes of any foundational ontology are to act as a starting point for building new ontologies, as a reference point for easy and rigorous comparisons
among ontological approaches, and as a foundational framework for analyzing, harmonizing, and integrating existing ontologies. Among the various available foundational ontologies (e.g., [65, 58, 9, 25]; for a comparison, see [44, 7]), we choose DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [44], which rests on intuitively attractive ontological choices for the present discussion. In particular, DOLCE is descriptive in that it aims to capture the ontological categories underlying natural language and human common sense. Categories in the ontology are therefore conceived as cognitive artifacts ultimately depending on human perception, cultural imprints, and social conventions.

2. Because RE involves the manipulation of representations of reality, we rely on Smith and colleagues’ reference terminology for ontology research and development [66] to provide foundations for the distinction between reality, cognitive representations thereof, and concretizations of cognitive representations. Smith and colleagues’ contribution also lets us distinguish particulars from universals.

3. As we know that requirements may express desires, intentions, beliefs, and attitudes. We therefore need to accommodate these categories within the foundational ontology.

4. Finally, desires, intentions, beliefs, and attitudes are usually expressed in natural language so that a classification of natural language expressions according to their pragmatics is needed; we employ Searle’s [59] illocutionary acts to this aim.

We discuss the integration and relevance of the above components throughout the paper. Principal original contributions from the RE perspective alone are the following:

- We identify ontological categories in which key RE concepts are grounded. We therefore restrict the meaning of RE concepts more precisely than usual natural language definitions do. In other words, we define CORE by mapping its concepts to concepts and relations in a higher-level, foundational ontology. We thus define key RE concepts using a restricted vocabulary, wherein terms have already been discussed and their meanings grounded in explicit philosophical choices. For instance, when we define a requirement in terms of DOLCE ontological categories, we use terms whose meaning has already been discussed and made explicit, along with its limitations elsewhere. Any discussion of the proposed definitions is thus facilitated, since many choices leading to the ontology are explicit.

- In addition to the requirement concept, we study various closely related concepts including ‘environment’, ‘system’, ‘machine’, ‘environment assumption’, along with the common taxonomic categories of the requirement concept. We argue for the introduction of new notions, namely those of ‘preference’ and ‘priority’.

- We discuss the relationship between desires, intentions, beliefs, and attitudes and the basic RE concepts. We further consider the relevance of illocutionary points as means for expressing desires, intentions, beliefs, and attitudes. To the best of our knowledge, this is the first time such a link is treated in RE. We thereby relate mental states and pragmatics of natural language with RE concepts—roughly speaking, we establish how the way in which something is expressed influences whether it is to be understood as one or another kind of requirement, or domain assumption, or otherwise. This is of particular interest for future research on requirements elicitation from speech. For instance, this contribution gives sound reasons to reuse available results in natural language processing for requirements elicitation.

- The various interwoven definitions build up to an ontology of core concepts in requirements engineering. The resulting, so-called CORE ontology, calls for a revision of the accepted understanding of the requirements problem. We outline the revised requirements problem and contrast it to the current variant. In this respect, the most apparent departure from established ideas is that the aim of RE is not only to satisfy but also to optimize the degree to which the requirements are satisfied. The accepted requirements problem is, given domain assumptions \( D \), to find a specification \( S \) such that requirements \( R \) are satisfied, i.e., \( D, S \models R \) [77]. With CORE in place, and given a classification of requirements to hard requirements \( HR \) (which must be satisfied) and soft requirements \( SR \) (which are desired, but not compulsory), the requirements problem is to find a specification which optimally satisfies all hard and either none, some, or all soft requirements (i.e., satisfies to the
1.4 Organization

We start with conceptual foundations (§2) on which we then build the definitions of ‘environment’, ‘system’, ‘machine’ (§3), and ‘information source’ (§4) concepts. The requirement concept is defined (§5) and a classification suggested and discussed (§6). The requirements problem is revisited in light of the obtained CORE ontology (§7). The concept of specification is defined (§8). We finally relate the definitions in CORE to the goal concept and its variants commonly used in RE (§9). Related work is discussed throughout the various sections. Conclusions and pointers for future work close the text. Ad hoc examples from the travel booking domain prove sufficient to illustrate our arguments and conclusions throughout the paper.

Some simple notational rules are observed in the remainder. When, e.g., we write environment (no emphasis whatsoever) our intention is to refer to whatever the reader understands as environment (i.e., we refer to the reader’s intuition) given the context of the present paper. When we write environment, we refer to the concept (or relation) which has been either been defined in the text or is a primitive (i.e., left undefined because clear enough). In domain-specific examples, we emphasize domain-specific concepts and relations: for a flight booking system, we say a flight ticket (no emphasis) to refer to what the reader conceives to be a flight ticket, whereas we write flight ticket to refer to how a flight ticket is understood within the flight booking system (it may refer to, e.g., a class in a particular data schema for the given system). Same applies for domain-specific relations.

2 Foundations

This section motivates the need for, and introduces rather uncontroversial notions from research in ontology design. In the RE phase of any development project, we can reasonably assume that there are requirements engineers and information sources, and both are located in an environment. Requirements engineers obtain information relevant for the engineering of requirements from information sources. Information sources usually include whatever information the stakeholders express, documentation, and, e.g., specifications of legacy systems that are part of the environment. Information that sources provide is relevant if it is about the environment. We are fully in line with Zave and Jackson [77]:

“A portion of the real world becomes the ‘environment’ of a development project because its current behavior is unsatisfactory in some way. The developers propose to build a computer-based machine and connect it to the existing environment in such a way that the behavior of the environment becomes satisfactory.”

Zave and Jackson distinguish machine from system as follows:

“[W]e use ‘system’ only to refer to a general artifact that might have both manual and automatic components, such as an ‘airline reservation system’. Whenever we are referring to the computer-based artifact that is the target of software development, we use the more precise term ‘machine’,”

They then clarify how information relevant during RE relates to the environment:

“[A]ll statements made in the course of requirements engineering are statements about the environment.”

By the virtue of not being openly debated, all of the above is uncontroversial in RE. Intended interpretations for the given concepts can be more precisely delimited. Namely, it would be relevant to understand when we speak of reality and when of a cognitive representation of reality or of a concretization of a cognitive representation. It appears for instance that a statement is about the environment, and it is likely a concretization of a cognitive representation of the environment. To clarify this within a restricted vocabulary and without digressing to semiotics, we borrow from Smith and colleagues’ reference terminology for ontology research and development [66], which they introduce to discuss best practices in ontology design in the (bio)medical domain; their proposal is, however, generic.
2.1 Entities, Representations thereof, and Concretizations of Representations

Following Smith and colleagues, an entity is anything which exists (e.g., flight ticket, aeroplane, pilot, etc. if we are interested in flight booking). A representation is anything which refers to or is about or is intended to refer to or be about, some entity external to the representation (e.g., someone’s understanding of a text1 describing the lifecycle of flight ticket is a representation). Composite representation is a representation built of its constituent representations, whereby the smallest constituent representations are called representational units (e.g., names used to represent individual lifecycle phases in the text describing the lifecycle of flight ticket for some flight booking system). A cognitive representation is a representation whose representational units are ideas, thoughts, or beliefs in the mind of some cognitive subject (e.g., the idea of a lifecycle of flight ticket had by a requirements engineer performing RE for an automated flight booking system). Finally, a representational artifact is a representation that is fixed in some medium in such a way that it can be used to make cognitive representations of separate cognitive subjects accessible in some enduring way (e.g., text in which the said requirements engineer expresses his idea of what is a lifecycle of flight ticket).

We thus have the following relations:2

- refer from representation to entity (or referred-by if inverse);
- symbolize from representational artifact to representation (or symbolized-by if inverse);
- represent from representational artifact to entity (or represented-by if inverse);

These first three relations above are not unlike those of Ogden and Richards’ semiotic triangle [51]; our entity can be read as their ‘referent’, representation as ‘though or reference’, and representational artifact as ‘symbol’. Such similarity is not unexpected—as will be seen in the remainder, requirements are statements about the environment, and in this respect these statements are representational artifacts, and obey the above given relations. We can also accommodate the empirical problem of agreeing, or equivalently, achieving a shared understanding of representational artifacts such as requirements models. Finally, we explicitly allow for divergence in what representational artifacts symbolize what cognitive representations, and represent which entities; in other words, two requirements engineers will not produce the same models (whereby ‘model’ we mean, e.g., a specific Tropos goal model, or KAOS goal model, or particular UML class diagram) for the same environment. Moreover, if we take represents relation to be mediated and not basic primitive relation, we are in line with intuitions accepted in RE and introduced in research on viewpoints ([50] and subsequent). A basic primitive relation [44] is a domain-independent relation that holds on its relata as soon as the relata are given; a mediated relation instead holds only if its mediating entity is given along with relata. Each stakeholder thus has a possibly overlapping but also potentially divergent cognitive representation of the system from other stakeholders, so that they may not share representational artifacts. We can argue for this the other way round: because there is a need for agreeing on representational artifacts, there is mediation through cognitive representations.3

2.2 Universals and Particulars

We can further distinguish particulars and universals, whereby a universal is something that is shared by all those particulars which are its instances. Smith and colleagues then introduce collection of particulars as a particular which includes other particulars which are its members. General term represents collection of particulars (i.e., General term is collection of particulars). Note that a general term may be such as to give a collection whose members are only those particulars which are instances of some universal (see, [66] for details). There are relations between particulars and (between) universals; those relevant here are:

- instance-of from particular to its corresponding universal (e.g., this particular flight ticket instance-of flight ticket, whereby flight ticket is taken as a universal);

---
1The text itself is not a representation but a concretization of the representation, as explained later on.
2These are preliminary; we return to time-index them later on.
3We have proposed elsewhere [29] an approach to organize the activity of arriving at such an agreement through argumentation and justification.
• member-of from particular to a collection of particulars (e.g., this particular flight ticket member-of cheap flight tickets);

• p-part-of from particular to particular is the common mereological parthood relation between particulars (e.g., straightforward from the above);

• u-part-of from universal\textsubscript{1} to universal\textsubscript{2} if and only if \[64\]: (i) for any instance \(x\) of universal\textsubscript{1} existing at some time \(t\), there is some simultaneously existing instance \(y\) of universal\textsubscript{2} such that \(x\) p-part-of \(y\), and (ii) vice versa for any instance \(y\) of universal\textsubscript{2} (e.g., terms of use u-part-of flight ticket, whereby flight ticket is taken as a universal);

• is-a from universal\textsubscript{1} to universal\textsubscript{2} if and only if \([64]\) for all times \(t\), if anything instantiates universal\textsubscript{1} at \(t\) then the same thing must instantiate also universal\textsubscript{2} at \(t\).

2.3 Change over Time, and a Foundational Ontology of Particulars

To move to definitions of the environment and other, properly speaking RE concepts, we need to consider RE as a domain, and thus descend to a level of abstraction below that in which we stated the existence of entities, particulars, universals, and others, and of their mentioned relations. Note that in terms of precision, we choose to make the common notions in RE more precise by grounding them in a restricted vocabulary ontologies, and in this way delimit their intended meaning. With this in mind, we may be tempted to suggest the following definition for the concept of environment.

Environment is-a entity, p-part-of which are (i) entities referred-by cognitive representations that stakeholders have, and/or (ii) entities represented-by information sources.

An alternative to the above is not to have entities part-of environment, but particulars part-of environment. This is inappropriate since when we speak of, e.g., flight ticket to state what is booked in a flight booking system, we may conceive of situations in which flight ticket is some particular flight ticket we are interested in, and in which we are interested in all that bears inherent characteristics to what we understand to be a flight ticket.

While acceptable at first sight, the proposed definition fails to highlight one essential characteristic of the environment. Namely, the environment is never fully known, so that the collection of entities to which stakeholders’ cognitive representations refer is not the same over time. To see why, take the opposite claim: at the start of the development project, stakeholders precisely know all there is to know of the environment. The environment here bears the characteristics of any decision environment, that is, one in which a problem is to be resolved by evaluating alternative courses of action, then choosing the optimal one (or a feasible best). Our stakeholders would then bear strong resemblance to the perfectly rational economic man; we borrow a description of such a decision maker from Simon [62]:

“This [perfectly rational] man is assumed to have knowledge of the relevant aspects of his environment, which, if not absolutely complete, is at least impressively clear and voluminous. He is assumed also to have a well-organized and stable system of preferences, and a skill in computation that enables him to calculate, for the alternative courses of action that are available to him, which of these will permit him to reach the highest attainable point on his preference scale.”

Simon was among the first in economics and decision sciences to suggest an alternative view, that of a more realistic decision maker who has limited grasp of the decision problem, and therefore cannot arrive at the optimal solution. The requirements engineer is assumed to be a boundedly (instead of perfectly) rational decision maker in Mylopoulos and colleagues’ contributions on the modeling of and reasoning about nonfunctional requirements [48, 6], in which they draw on some of Simon’s work in AI [63]. The engineer ‘satisfices’ when designing a system, instead of optimizing. In other words, the optimal system design cannot be found because of bounded rationality: only a system design can be found which satisfies requirements to a more desirable (but not optimal) extent relatively to other known designs. The said perspective is mostly implicit elsewhere in RE; it is present because dictated by practical experience. The reasonable alternative then is to take that stakeholders change what they consider to be the environment over the course of RE. One way to understand how this changes is to assume that stakeholders learn
over the course of RE, initially having only a limited knowledge of how they expect the environment to change through the introduction of the future system, and expanding on this knowledge with help of RE methodologies and domain expertise. Returning to the problem of defining the environment, we need to make explicit that the environment is not fixed at any particular point in time, but that its identity is time-dependent, which would reflect the intuition that the environment at some time (e.g., start of the development project) is different than at some subsequent time. One apparent way in which it changes is that it includes the system and thereby the machine after the latter is deployed. But it is not the only way: in aiming to automate flight booking, stakeholders may initially only be concerned with the ‘flight booking environment’ then expand it to include the booking of rental cars to benefit from a partnership with a rental organization; or they may learn at some point of the project that notifications of delays can be communicated in some novel ways (e.g., through mobile messaging) and expect that it be enabled by the system.

To incorporate time in our definitions, we must have it explicit in the upper-level ontology in which we ground CORE. To do this, we use a foundational ontology. A foundational ontology is a theory about the abstract domain-independent categories in the real world. Its main purposes are to act as a starting point for building new ontologies, as a reference point for easy and rigorous comparisons among ontological approaches, and as a foundational framework for analyzing, harmonizing, and integrating existing ontologies. Among the various available foundational ontologies (e.g., [65, 58, 9, 25]; for a comparison, see [37, 7]), we choose DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [37], which rests on intuitively attractive ontological choices for the present discussion. The concepts and relations we introduced earlier can be understood as a foundational ontology, though one abstract enough to be of rather limited use. DOLCE makes the following ontological choices:

- DOLCE is descriptive in that it aims to capture the ontological categories underlying natural language and human common sense. Categories in the ontology are therefore conceived as cognitive artifacts ultimately depending on human perception, cultural imprints, and social conventions. Descriptive contrasts to revisionary; choosing the latter commits to the aim of capturing the intrinsic nature of the world, that is, answering what exists, as opposed to what is perceived to exist.

- It is multiplicative, allowing different entities to be co-localized in the same space-time. Broadly speaking, in a multiplicative ontology, we can say “the paper ticket is constituted of an amount of paper”, whereas in a reductionist ontology, we would say “the paper ticket is an amount of paper”. Consequently, if we speak of the departure date, we say in the former case “the paper ticket has a departure date” and not “the amount of paper has a departure date” as it would be more appropriate in the latter case.

- DOLCE takes a possibilist view: entities are allowed independently of their actual existence. At first sight, possibilism seems well adapted to RE for we are concerned with optative statements, which speak of what is presently not but is desired to be. DOLCE seems to adopt possibilism as a consequence of adopting modal logic for its formal characterization in documentation [44]. The actualism/possibilism discussion is considerably more elaborate, so that we may still properly deal with optative statements in (some variants of) actualism (for an introduction to the debate, see [46]).

- It distinguishes between and allows both enduring and perduring entities assuming thus that entities have both temporal and spatial parts. The ontological choice at play here is that between endurantism and perdurantism, which has consequences on the questions of how identity and change are understood. DOLCE’s cognitive bias leads to allowing both endurants and perdurants, whereby they differ in that something is an endurant iff (i) it exists at more than one moment and (ii) statements about what parts it has must be made relative to some time or other. This is interesting in the present discussion mostly in that the cognitive bias is accounted for, so that statements about perdurants and endurants are equally accepted.

- DOLCE is an ontology of particulars—DOLCE takes the domain of discourse not to contain, e.g., the instantiation relation, so that no entity in the domain has instances. The consequence is that universals are not subject of being organized and characterized within, but are left outside DOLCE.

The last choice in DOLCE may appear in contrast to the presence of universals herein. There is, however, no difficulty because [44]:

8
We thus have the $u$-part-of-at-t and represents-at-t.

We take the ontology of universals as formally separated from that of particulars. Of course, universals do appear in an ontology of particulars, insofar they are used to organize and characterize them: simply, since they are not in the domain of discourse, they are not themselves subject to being organized and characterized.”

The above leads us to organize the notions introduced above with DOLCE as shown in Figures 1 and 2, where relations parthood, membership, representation, reference, and symbolization relations are time-indexed; atemporal variants are maintained for parthood (see, [44] for a discussion) and membership (it makes sense to say without need to reference time, e.g., ‘january is member of the collection of months’). The is-a relation is atemporal. We assume that cognitive representations for an entity can vary over time (e.g., learning about the history of the Egyptian pyramids may lead to change the cognitive representations that refer to an Egyptian pyramid\(^4\)) so that the reference relation is time-indexed: same kind of reasoning justifies time-indexing of representation and reference relations.\(^5\) We thus have the following time-indexed relations, of which only represent-at-t is not a basic primitive relation:

- **refer-to-at-t** from representation to entity at time \(t\) (or referred-by-at-t if inverse);
- **symbolize-at-t** from representational artifact to representation at time \(t\) (or symbolized-by-at-t if inverse);
- **represent-at-t** from representational artifact to entity at time \(t\) (or represented-by-at-t if inverse);
- **member-of-at-t** from particular to a collection of particulars at time \(t\);
- **p-part-of-at-t** from particular to particular at time \(t\) is the time-indexed mereological parthood relation between particulars;
- **u-part-of-at-t** from universal to universal at time \(t\) is the u-part-of relation indexed with time.

Above, \(t\) is a temporal region in DOLCE.

\(^4\)We can say the same, from the perspective of a domain-independent requirements engineer, *mutatis mutandis* for, e.g., flight ticket, booking process, cancellation process, and others relevant in flight booking. Grounding an opposing position to the above by arguing that there are engineers specialized for the domain is vacuous, for each particular project carries (more or less) specificities that are learned over its course and affect subsequent decision making.

\(^5\)We have no examples of entities, cognitive representations, and and representational artifacts among which there are atemporal representation, reference, and symbolization relations; hence the committment to time-indexing of these relations.
We return to the notion of environment and information source, and propose the following definitions.

**Definition 1.** Given temporal region $t$, environment is a general term which represents-at-$t$ a collection of particulars, whose members (i.e., related to the collection by the member-of-at-$t$ relation) are, for any
stakeholder: (i) any entity referred-by-at-t cognitive representation that stakeholder has-at-t; and (ii) for any information source, any entity represented-by-at-t the information source that stakeholder communicates-at-t. Any entity member-of-at-t environment is represented-by-at-t relevant. Relevant is-a general term.

Definition 2. Given temporal region t, information source is-a representational artifact communicated-by-at-t stakeholders. Stakeholder communicates-at-t information source if information source symbolizes-at-t cognitive representation that stakeholder has-at-t.

Remarks:

- Definition 1 roughly states: over a given time period, the environment is a representational artifact which represents only those entities which are relevant and which are referred to by stakeholders’ cognitive representations and/or represented by information sources that stakeholders use over the course of the given time period.

- We subsume the requirements engineer within stakeholders in the definition. The requirement(s) engineer(s) is/are indeed an individual or group who can affect or is affected by the achievement of the system’s objectives [16].

- That environment is not a universal arises from the difficulty of arguing that an entity’s belonging to an environment is inherent in the entity; instead we allow environment to be a collection of particulars by the virtue of being a particular (therefore, e.g., ‘flight booking environment’ can be taken, depending on the situation, as a collection and thus possibly a collection of various particular environments which are instances of the ‘flight environment’, or a given particular environment; but we do not allow that ‘flight environment’ instantiates, e.g., ‘flight ticket’, which it could if it were taken to be a universal).

- We introduce the relation communicates-at-t to relate information source to stakeholder, and the relation has-at-t to relate stakeholder to cognitive representation. Because it is straightforward, it is a primitive relation and is therefore left undefined. For the communicates-at-t, we only give a necessary condition stated in Definition 2.

- Definition 2 indicates that information sources are used by stakeholders if they symbolize stakeholders’ cognitive representations during a given time period. This is only a necessary condition for some information source to be used by a stakeholder because we cannot reasonably expect a stakeholder to use all information sources that may be symbolizing her cognitive representations. For example, a body of written resources which concern the flight booking domain may contain those which are meaningful for a stakeholder (i.e., symbolize her cognitive representations), but she cannot be expected to consult all of these.

- In Definition 1 taking either only entities designated in (i) or only those in (ii) is inappropriate. Not all relevant in the environment is, broadly speaking, known by the stakeholders, just as not all relevant is to be found in information sources. There is an important nuance in Definition 1 in this respect: namely, entities designated in (i) include both those that have a counterpart in representational artifacts that the requirements engineer can ‘access’ (e.g., documentation, verbal communication with other stakeholders, and so on), and those entities that are not made explicit in representational artifacts (i.e., inaccessible because the engineer does not ask the ‘right’ questions) or is particularly difficult to make explicit. These last cognitive representations relate to Polanyi’s notion of ‘tacit’ knowledge [53]; in this respect, Nonaka’s observations [49] on knowledge creation in organizations are relevant for the requirements engineer interested in understanding the dynamics of knowledge creation, and thus of the presence or absence of representational artifacts in organizations (along with the subsequent difficulty in requirements elicitation).

- Definition of relevant is absent. Definition 1 delimits the scope of the environment merely by stating that only relevant entities are parts of the environment. Otherwise, anything referred to by stakeholders’ cognitive representations may be considered as environment, just as anything that is represented in various available information sources. While this appears unwanted, we can only reiterate the difficulty in precisely delimiting the environment of a development project. For instance, one might hope to provide criteria, such as, e.g., that what is in the scope is only what is susceptible to be influenced by the future system. Such criteria fail upon closer inspection:
chains of cause and effect are difficult to elucidate and we already explained that the boundary of the system is unknown at the outset of the development project. One criterion which appears reasonable is that of efficiency: while much can be in the environment, it is ultimately limited by the stakeholders’ and engineer(s)’ limited resources, including, e.g., time and attention span. We consider the applicability of the general term relevant to be a methodological consideration, that is, relevance will be determined and local to the RE methodology employing CORE.

- We do not define stakeholder (that is, we can only relate herein stakeholder through a chain of is-a relations to a Physical Endurant in DOLCE). As uniform precise and practical criteria for who stakeholders are, are unavailable (see, e.g., [47, 26] for discussions), we leave this to the methodology which uses CORE.

We now turn to Zave and Jackson’s ‘system’ and ‘machine’. If the overall aim of a given development project is to automate flight booking, one may then speak of the flight booking environment, flight booking system in the environment, and the system’s automated part, which Zave and Jackson would call the flight booking machine. One acceptable criterion for distinguishing among the latter two is that there are not human ‘parts’ of the machine, which is to say that the machine encompasses only automated parts. People using or otherwise depending on it are at the boundary of the machine and of the system—e.g., the travel assistant who performs passenger check-in at the flight gate can be considered ‘in’ the system while outside the machine. Consider now whether flight control is within the environment or not. Flight control is relevant for flight booking mostly in that it provides data on current arrival and departure times, delays, and cancellations (and, e.g., accidents and incidents, but this is not affect the discussion). It is relevant therefore, since the data it produces are undoubtedly used in current flight booking and are likely to be used after the booking is automated. To know whether flight control is ‘in’ the flight booking system, there are two questions to ask: first, whether there are requirements that concern (i.e., refer to) flight control; second, whether the automation of flight booking affects the flight control environment. If answers to both are positive, than it is relevant to consider what particular parts of the flight control environment will be/are affected and how. If at least one answer is not positive, flight control (or some part thereof) remains ‘in’ the flight booking environment but ‘outside’ the flight booking system. The criterion used is then the following: if some part of the environment referred to in requirements changes because of change of and/or creation of entities that are needed to satisfy requirements, that part of the environment is also part of the system. If, for instance, to automate flight booking requires developing new interfaces with the ‘machine’ used in flight control (so that, e.g., this machine publishes flight data in some standard format) those interfaces would be considered in the flight booking system. Definition 3 assumes that t encompasses the whole duration of the development project and that time is linear and non-branching; this assumption is not made in Definitions 1, 2, and 4.

Definition 3. Given temporal region t, system p-part-of-at-t environment. Any entity which is member-of-at-t environment is also member-of-at-t system iff some requirement represents entity and entity is created or changed during t.

Definition 4. Given temporal region t, machine p-part-of-at-t system. Any entity p-part-of-at-t system is also p-part-of-at-t machine if it does not hold that person is-a entity.

Above, person (understood as human, physical person) is related through a chain of is-a relations to DOLCE Physical Endurant.

4 Classification of Information Sources

We quoted Zave and Jackson [77] who suggest that any statement relevant during RE is about the environment. They then indicate:

“The primary distinction necessary for requirements engineering is captured by two grammatical moods. Statements in the ‘indicative’ mood describe the environment as it is in the absence of the machine or regardless of the actions of the machine; these statements are often called ‘assumptions’ or ‘domain knowledge’. Statements in the ‘optative’ mood describe the environment as we would like it to be and as we hope it will be when the machine is connected to the environment. Optative statements are commonly called ‘requirements’.”
This is an important observation for it accounts for the notion that information relevant for the requirements engineer is about the truths about the problematic situation in which the development project is initiated, and which will hold after the system is deployed, and about what stakeholders desire be the case after system development and deployment. We may say then that some information sources convey what is and will be (i.e., ‘assumptions’ or ‘domain knowledge’), and some what is desired (‘requirement’). Relevance of the observation is evident in light of the use of the notion of ‘goal’ as one of key representational units in RE. A ‘goal’ in RE usually denotes what is understood as an intention in AI [8], that is, an assertion which is at present false but whose truth is desired (which breaks down to Zave and Jackson’s ‘requirement’, and which is called ‘goal’ in AI) and commitment to take action to bring about the desired state.

Recall that information source is a representational artifact which is used by stakeholders. By definition, a representational artifact is used to make cognitive representations of separate cognitive subjects, that is, stakeholders, accessible in some enduring way. Zave and Jackson’s observation entails a complete partition of information sources on optative and indicative ones. We explain below why such a complete partitioning is inappropriate.

When Zave and Jackson use ‘indicative’ and ‘optative’, they are concerned with the grammatical mood of a given statement. Broadly speaking, in any given statement the dictum, or what is said is distinguished from the modus, or how it is said in terms of speaker’s attitude about what is said. Arguments in favor of such distinction are discussed in and by, among others: Frege’s Begriffsschrift (translated in, e.g., [72]), where the content of judgement is distinguished from the act of judgement of the content; Wittengstein’s Tractatus logico-philosophicus [41], where the content of the proposition corresponds to a possible state, and it distinctively asserted that the proposition describes the real state; Stenius [68] separates a sentence radical which gives the descriptive content of a sentence, from a modal element which signifies mood (of indicative, imperative, or interrogative kind). In reading a written statement, we observe syntax in which the statement is written, attribute semantics to it, while grammatical mood may not be apparent. Mood nevertheless influences the meaning intended for a given dictum. Although seldom, if ever discussed by others than Zave and Jackson, the relevance of grammatical mood on the semantic content of a statement appears to be an implicit assumption in most of RE, and certainly in all of RE which deploys the notion of ‘goal’. To see how, consider Ex.1 as an example of a statement.

(Ex.1)
A business seat is more expensive than an economy seat.

Whether the above is presently true or someone desires it to become true at some future time is not apparent from Ex.1 alone: if it is presently true, we consider it as being in indicative mood and thus as ‘domain knowledge’ according to Zave and Jackson; if it is instead desired and presently false, it is in optative mood and is considered a ‘requirement’. In the latter case, there is no difficulty in RE frameworks such as, e.g., KAOS and Tropos to formulate Ex.1 as a ‘goal’. In this Zave and Jackson, and much of RE research follow arguments according to which grammatical mood is relevant for semantic content (e.g., [19] and related). However, optative and indicative are far from being the only grammatical moods—in addition to Ex.1, statements below each correspond to a different grammatical mood.

(Ex.2)
No business seat will have a lower price than an economy seat.

(Ex.3) If there are no special offers on business seats, a business seat is more expensive than an economy seat.

(Ex.4) There may be business seats that are less expensive than economy seats.

(Ex.5) There must be no economy seat more expensive than any business seat.

(Ex.6) There have never been special offers that make a business seat less expensive than an economy seat, so an economy seat will never be more expensive than a business seat.

---

6Note that the logical formalism of choice for writing and, e.g., verifying goals in RE are variants of linear temporal logic ([10, 17] provide examples). Any such logic does not in its semantics distinguish goals from other expressions—it is the RE framework employing the logic that says whether some expression is a goal or otherwise. In this respect, RE frameworks such as KAOS and Tropos include a syntax and semantics within a single logical formalism, while adding pragmatics (see, below) on top and externally from the logical formalism of choice.
There must be a special offer in order for a business seat to be cheaper than an economy seat.

I was told that there cannot be an economy seat which is less expensive than a business seat.

If the traveller trades in the sufficient number of miles, the price of her business seat can be reduced to below the price.

A business seat that is less expensive than the most expensive economy seat can be booked.

There is probably no business seat less expensive than the most expensive economy seat.

There will be special offers that decrease the price of a business seat below that of the most expensive economy seat.

Ex.2 is intended to be read in optative mood, that is, its modus expresses wishes, hopes, or commands. Ex.3 is in conditional mood, stating that a realization is dependent on whether a certain condition obtains. Dubitative mood exemplified in Ex.4 indicates uncertainty about the dictum “business seats are less expensive than economy seats”. Direct command or prohibition is given in Ex.5, in imperative mood. Ex.6 is in inferential mood, in which the speaker assumes that what she says is true based on inference from other information. Ex.7 expresses a necessary or sufficient condition for “a business seat is cheaper than an economy seat” to hold, and is considered in the intentional mood. When the statement conveys something that the speaker did not personally witness, it is in indirect indicative mood (Ex.8). The permissive mood (Ex.9) expresses permission to perform some action. The potential mood (Ex.10) conveys the possibility of something to hold. Finally, predictive mood (Ex.11) “indicates that although the speaker is not certain that the event she is speaking of will occur, she thinks it is likely enough to make a prediction” [45], and promissive mood (Ex.12) conveys promises. This inventory draws on work in natural language processing [45]. It is not exhaustive as (some) moods are language-specific (for discussions, see, e.g., [27] and later). Since all of the given statements Ex.1–Ex.12 can straightforwardly be considered as representational artifacts, it is unclear why only optative and indicative are relevant for RE. Even if we interpret Zave and Jackson’s indicative to subsume all non-optative statements, it remains unknown why, e.g., imperatives are not requirements. Otherwise, it is the status of the ten other moods exemplified above that remains unclear in RE. In the face of a statement in, e.g., inferential or potential, do we treat it as ‘domain knowledge’ or ‘requirement’? If we take a statement in inferential to be ‘domain knowledge’, it may be useful to know that it has different status, in terms certainty, than domain knowledge expressing natural or social laws. Evidently, we hope to have facts for domain knowledge and optatives for requirements, but we are likely to face a whole range of dictum/modus cases in information sources. To completely partition information sources onto optatives and indicatives is to take a shortcut unacceptable when classification of information sources is performed.

4.1 Classification Grounded in Speech Acts

While mood has a grammatical flavor, the distinction dictum/modus also corresponds to Searle’s separation of, respectively the proposition from the illocutionary (or speech) act [59]. Turning to speech acts instead of grammatical mood (as discussed above, and as common in natural language processing) gives us a natural language-independent basis for classification.7 Searle distinguishes [60] the following kinds of illocutionary acts:

1. **assertives**, which assert the proposition that the speaker believes is true;

2. **directives**, which convey the proposition that the speaker wants to see become true;

---

7This is not to say that Searle’s classification of speech acts is not debated—see, e.g., [75] for a recent discussion. It is, however, widely accepted and fits the practical purpose herein.
3. **commissives** stating what the speaker intends to (do to) make a proposition true;

4. **expressives** conveying speaker’s emotion/attitude about herself or the hearer;

5. **declarations** which by the very act of being stated make a proposition true;

6. **representative declaratives** which recognize the truth of a proposition that has been made true through a declaration.

Returning to the earlier examples, we can consider our examples in the indicative (Ex.1), conditional (Ex.3), dubitative (Ex.4), inferential (Ex.6), intentional (Ex.7), indirect indicative (Ex.8), potential (Ex.10), and predictive (Ex.11) moods as assertives; optative (Ex.2) and imperative (Ex.5) as directives; promissive (Ex.12) as a commissive; permissive (Ex.9) as a declaration. Note that we do this merely for the sake of example—we cannot be prescriptive in relating grammatical moods to illocutionary acts since a precise link seems absent. Observe for instance that, e.g., we can take our example for the potential mood (Ex.10) and understand it as a declaration.

Given the above discussions, it is important to observe that from the perspective of our present aims in establishing an ontology for RE, Searle’s taxonomy of illocutionary acts is relevant, but lacking with regards to the richness of grammatical moods. The illocutionary acts taxonomy is relevant because it lets us admit Zave and Jackson’s intuitive distinction between what is desired and what is currently (believed) true, and this in terms of, respectively, directives and assertives. It also lets us have a preliminary separation between what is open to automation and what is done by the stakeholders within a system, namely by distinguishing directives from commissives. A directive carries no commitment as to who is to bring about what is desired, which is not the case for the commissive. Declarations have a straightforward intuitive interpretation in RE as statements about rules or social laws of the environment, brought about by inherent properties of the environment, but by the social conventions, history, or otherwise. What the taxonomy lack from our perspective are the notions of uncertainty and conditionality.

### 4.2 Uncertainty and Conditionality

Uncertainty is significant both in relation to what is desired and what is believed to be currently the case. As March pointed out regarding sources of difficulty in decision making [43]:

> “Rational choice involves two kinds of guesses: guesses about future consequences of current actions and guesses about future preferences for those consequences.”

When a stakeholder expresses her desires, she states desired outcomes of the development project, so that she anticipates an uncertain outcome and her preference for that outcome over anticipated alternative outcomes. There is therefore inherently more or less uncertainty in directive statements of interest for the requirements engineer. Same applies for what Zave and Jackson call domain knowledge: while some of it involves no uncertainty (at least not to the point of being relevant in the practical setting), such as, e.g., laws governing aviation and contracts between the traveler and the airline, statements such as that in Ex.10 may be considered more or less certain, except if it is a declaration given by an institutionally relevant stakeholder and is not opposing certain and fixed domain knowledge. We consequently need to distinguish, among statements those which are certain enough for all practical purposes, from others which are uncertain. This is relevant from a methodological perspective, as one does not analyze (sufficiently) certain and (sufficiently) uncertain statements in the same manner; if for instance we need to establish whether there can or cannot be business seats that are less expensive than most expensive economy seats, we are interested in being as certain as possible which of the two options is what the stakeholders assert or desire, since it may translate in a rule for checking the consistency of some database within the

---

8What the examples lack is the background over which the statements are made [61]. One way of understanding the background of a statement is by knowing the purpose of the discourse (i.e., conversation) within which the statement appears. Vanderveken [71] suggested that descriptive (in which beliefs about what is true are exchanged), deliberative (in which the speaker and hearer deliberate on what to commit), declaratory (in which declarations are made), and expressive (in which attitudes of the speaker and the hearer are conveyed) discursive goals be distinguished. If, e.g., we know that Ex.10 is stated within a declaratory discourse, we would consider it a declaration and not otherwise. Vanderveken’s distinction may be relevant from a methodological perspective for the elicitation of requirements, but we leave this discussion for a separate treatment.

9That is, a stakeholder who has the authority to make declarations which end up being followed.
automated flight booking system. In this respect, we would consider Ex.6 above, i.e., the assertive in inferential mood, as uncertain.

The interest in conditionality arises from the need to express that for beliefs (as in assertives), desires (as in directives), intentions (as in commissives), and world-altering utterances (as in declarations) to obtain depends on some other beliefs, desires, intentions, and world-altering obtaining. In the following example, we can see the condition as an assertive and the remaining proposition as a directive.

(Ex.13)  
If the payment is confirmed, send the booking confirmation.

Having shown above that the binary indicative/optative partitioning of information sources is a short-cut undesirable in the present discussion, that a rich inventory of grammatical moods is available, we return to the initial problem, namely that of providing a definition of requirement and domain knowledge/assumption notions. The definitions that we suggest below are grounded in the classification of information sources; that is, we classify information sources and determine which categories thereof can be called requirements and which domain knowledge. We base the classification of information sources on the conclusion of the above discussion: dictum/modus distinction can be accounted for in a language-independent manner using Searle’s classification of speech acts, augmented for uncertainty and conditionality. We obtain the classification shown in Figure 3.

At the first level of the classification, we distinguish free from constrained information sources. For example, the statement Ex.13 below is a free information source for our flight environment.

(Ex.14)  
Issue a ticket.

Ex.14 is called ‘free’ because there is nothing that indicates how the issuing of the ticket is to be performed, or under which conditions. We can constrain it by adding conditions and describing the ticket, as follows:

(Ex.15)  
If the payment is received and booking recorded, issue an electronic and paper ticket.

It is visible above why we go beyond conditioning in a classification of information sources. Namely, we speak of constraints, among which are conditions. Any constraint can be considered an information source in its own right. Ex.15 indeed involves several information sources, e.g., (i) “issue a ticket”; (ii) “every ticket takes electronic and paper form”; (iii) “payment is received”; (iv) “booking is recorded”.

The second level of the classification introduces the degree of certainty of the information source. The taxonomic representation in Figure 3 hides an important nuance: there is rarely complete certainty, so that both ‘uncertain’ and ‘certain’ are to be read ‘sufficiently (—) for the given practical purposes’ where (—) is replaced with either of the two. For constrained information sources, certainty is determined from the certainty of individual information sources that compose it. A safe choice is to consider a constrained information source as certain at most to the degree of its least certain component.

Finally, the third level gives the classification of the information source according to its relevant illocutionary act. For free information sources, classification is straightforward. For constrained sources, no precise rules based on compositionality are available, so that the third level classification for constrained sources is based on the illocutionary act in which the whole of the constrained information source is given. We have the following definitions.

Definition 5. Given temporal region t, free information source is-a information source such that there is no other information source, p-part-of-at-t or u-part-of-at-t information source.

Definition 6. Given temporal region t, constrained information source is-a information source such that there is at least one other information source, p-part-of-at-t or u-part-of-at-t information source.

4.3 Relating Speech Acts to Beliefs, Desires, Intentions, and Attitudes

We first consider the third level of classification and return to the certainty distinction afterwards. To define each of the concepts obtained at the third-level of classification of information sources, we need to look into how Searle conceptualizes the differences between speech acts and see how his conceptualization can be localized within the foundational ontology we use. Searle grounds the classification of speech acts using four dimensions of discrimination. First, the illocutionary point designates the purpose intended by the speaker when making an utterance. The illocutionary point can be assertive, directive, commissive, expressive, declarative, and representative declarative. In the ontology, we consider each of the four to be related upwards through is-a relations to communication, whereby communication is-a process, and process
**is-a stative** in DOLCE. We consider them as being ways of communicating. Second, the *direction of fit* distinguishes utterances which fit the actual state of affairs (termed “word-to-world”) from those which alter the actual state of affairs (“world-to-word”). This separation has no corresponding categories in the foundational ontology. It is treated in terms of fit between the the cognitive representation and the entities the former refers to at a given time: word-to-world corresponds to a fit at the current time period between the cognitive representation and the entities, whereas world-to-word to the fit between the current cognitive representation and entities at a future time period. The fit is local to the stakeholder, that is, may not be true; instead, it is taken to be true by the stakeholder who has that cognitive representation. We therefore introduce the basic primitive relation *fits-at-t* from cognitive representation to entity, which holds if the stakeholder considers the a cognitive representation she has corresponds to entity. Third, psychological state of the speaker is either that of belief, desire, or intention. Since these are endurants and are non-physical, we consider them as sub-categories of DOLCE mental object. Finally,
propositional content conveys what is being, e.g., believed, desired, or intended by the speaker, or, in other words, communicated by the stakeholder using an illocutionary point. Propositional content translates to cognitive representation in Figure 1. Recall that, as pointed out earlier, cognitive representation is equivalent to DOLCE mental object.

Following Searle’s discrimination dimensions and speech act classification, and our localization of his conceptualizations within the foundational ontology deployed herein, we can partition information sources on assertive, directive, commissive, attitudinal, declarative, and representative declarative information sources. The definitions follow Searle’s use of discriminating dimensions to classify speech acts in six types that we recalled earlier. We obtain the following definition for the assertive information source.

Definition 7. Given temporal region \( t \), assertive information source is-a information source, which stakeholder communicates-at-\( t \) by means of assertive illocutionary point. The assertive information source symbolizes-at-\( t \) cognitive representation, which stakeholder has-at-\( t \). The cognitive representation fits-at-\( t \) entity. The cognitive representation is-a belief.

The assertive information source corresponds to the assertive speech act in that it is a belief which, according to the stakeholder fits reality; the belief is expressed using the assertive illocutionary point. A directive information source differs in that the stakeholder expresses the cognitive representation through the directive illocutionary point, she does not believe but instead desires what she expresses, and the fit is appropriate for some future (as opposed to current) time period.

Definition 8. Given current temporal region \( t \), directive information source is-a information source, which stakeholder communicates-at-\( t \) by means of directive illocutionary point. The directive information source symbolizes-at-\( t \) cognitive representation, which stakeholder has-at-\( t \). Given a later time period \( t' \), the cognitive representation fits-at-\( t' \) entity. The cognitive representation is-a desire.

A commissive information source expresses the stakeholder’s intention, fits a future time period, and is stated with the commissive illocutionary point. The attitudinal information source corresponds to the expressive speech act\(^{10} \); the fit is unspecified, the information source expresses an attitude, and is expressed using the expressive illocutionary point. Attitude has been defined as a relatively enduring organization of beliefs, desires, and intentions which gives rise to a psychological tendency that is expressed by evaluating an entity with some degree of favor or disfavor (e.g., [13, 22]). Because its relationship with beliefs, desires, and intentions is unclear, a safe choice is to position attitude as a sub-category of DOLCE mental object.

Definition 9. Given current temporal region \( t \), commissive information source is-a information source, which stakeholder communicates-at-\( t \) by means of commissive illocutionary point. The commissive information source symbolizes-at-\( t \) cognitive representation, which stakeholder has-at-\( t \). Given a later time period \( t' \), the cognitive representation fits-at-\( t' \) entity. The cognitive representation is-a intention.

Definition 10. Given temporal region \( t \), attitudinal information source is-a information source, which stakeholder communicates-at-\( t \) by means of expressive illocutionary point. The attitudinal information source symbolizes-at-\( t \) cognitive representation, which stakeholder has-at-\( t \). The cognitive representation is-a attitude.

A declarative information source is expressed with the declarative illocutionary point and there is no explicit psychological state. Since the utterance brings about, by the act of being expressed, change in entities, the direction of fit is also of no interest. Representative declarative information source instead expresses a belief which, according to the stakeholder fits reality.

Definition 11. Given temporal region \( t \), declarative information source is-a information source, which stakeholder communicates-at-\( t \) by means of declarative illocutionary point. The declarative information source symbolizes-at-\( t \) cognitive representation, which stakeholder has-at-\( t \). The cognitive representation fits-at-\( t \) entity.

Definition 12. Given current temporal region \( t \), representative declarative information source is-a information source, which stakeholder communicates-at-\( t \) by means of representative declarative illocutionary point. The representative declarative information source symbolizes-at-\( t \) cognitive representation, which stakeholder has-at-\( t \). The cognitive representation fits-at-\( t \) entity. The cognitive representation is-a belief.

\(^{10}\) The difference in terminology is introduced to avoid association with the notion of expressiveness and to highlight the notion of attitude.
For illustration, consider Ex.16 below. If it is believed by the stakeholder communicating it, and fits with the current way of performing bookings and confirmations, and is expressed through an assertive illocutionary point, Ex.16 is an assertive information source. If the statement below were originally written in, e.g., documentation, the requirements engineer can find out whether it is believed and whether it is asserted by asking appropriate questions to stakeholders: for the former issue, the question may concern their experience of whether the current practices correspond to the statement; for the latter issue, the engineer may ask whether the statement reflects current practices or not.

(Ex.16) Booking must be confirmed.

Note that the proposed conceptualization makes apparent the impact of the way in which an information source is communicated on how well its intended interpretations are delimited. Given a written document (which is, e.g., not a detailed transcript of stakeholder meetings), the engineer is likely to have significant difficulties in properly understanding the content of the document. Ex.16 is a declarative information source if the stakeholder communicating it has the authority to set the given rule. If the rule has been in application already, Ex.16 is a representative declarative. If for instance the stakeholder communicating the statement in Ex.16 considers that not all bookings are currently being confirmed, and desires that it be so in the future, Ex.16 is then a directive information source. Ex.17 gives a formulation of the statement in Ex.16 which is more likely to be understood as a commissive information source (we added the constraints ‘always’ and ‘in the new system’ for presentational convenience).

(Ex.17) I hope/wish/desire/expect booking to be always confirmed in the new system.

To have a commissive information source, intention must be present, as in Ex.18, which is a variant of Ex.16 communicated by the stakeholder responsible for, e.g., consumer service in automated flight booking. Ex.19 expresses an attitude about how booking confirmation is to be expressed.

(Ex.18) I will ensure that booking is always confirmed.

(Ex.19) It is preferred that the booking confirmation be sent quickly after booking.

4.4 Placing Uncertainty in the Foundational Ontology

Before we suggest definitions of ‘requirement’ and ‘domain assumption’ in relation to the classification of information sources, it is necessary to return to the second level of the classification at which more or less certain information sources are distinguished. Regardless of the specific source of uncertainty, we can reasonably argue that a stakeholder can be more or less certain about her beliefs, desires, intentions, and attitudes. Uncertainty in beliefs is an accepted notion. Regarding uncertainty of desires and intentions, it is enough to observe that in both cases the stakeholder is concerned about the future, so that uncertainty is certainly present. Namely, attitudes may change as new information is acquired, which is the same for desires. Same applies for intentions, as they contain a future reference as well. In this we follow March [43], who we quoted earlier to highlight that decision makers, herein stakeholders make a guess about their future preferences over outcomes, and thereby face uncertainty. For the present purposes, it is acceptable to consider the degree of certainty as being a classification dimension over beliefs, desires, intentions, and attitudes, and thus introduce sufficiently uncertain mental object and sufficiently certain mental object as sub-categories of mental object, orthogonal to the classification of mental objects into belief, desire, intention, and attitude among other. That is, a mental object is always classified both as either sufficiently certain/uncertain, and as either belief, or intention, or otherwise. Definitions related to uncertainty are straightforward after the notion of uncertainty is introduced as a primitive in sub-categories of DOLCE categories. Herein we leave the notion of sufficiency undefined, that is, we consider it specific to the intuition of the requirements engineer who interprets information sources that stakeholders communicate. How the degree of certainty/uncertainty is elicited is not of interest at present.

Definition 13. Given temporal region t, uncertain information source is-a information source. The uncertain information source symbolizes-at-t cognitive representation, which stakeholder has-at-t. The cognitive representation is-a sufficiently uncertain mental object.

Definition 14. Given temporal region t, certain information source is-a information source. The certain
(a) Extending DOLCE to accommodate illocutionary points.

(b) Extending DOLCE mental object.

**Figure 4:** Extension is shown of the taxonomy of DOLCE basic categories in Figure 2. The extension is needed for definitional purposes, as shown in the text. Illocutionary points in (a) are understood as being kinds of communication processes, whereby we do not postulate herein the precise chain between the communication process and the illocutionary acts. The second extension shown in (b) highlights how certainty/uncertainty and belief/desire/intention/attitude distinctions are introduced as sub-categories of the DOLCE mental object. We do not postulate that beliefs, desires, intentions, and attitudes exhaust all mental objects.

**Figure 4**

*Figure 4:* Extension is shown of the taxonomy of DOLCE basic categories in Figure 2. The extension is needed for definitional purposes, as shown in the text. Illocutionary points in (a) are understood as being kinds of communication processes, whereby we do not postulate herein the precise chain between the communication process and the illocutionary acts. The second extension shown in (b) highlights how certainty/uncertainty and belief/desire/intention/attitude distinctions are introduced as sub-categories of the DOLCE mental object. We do not postulate that beliefs, desires, intentions, and attitudes exhaust all mental objects.

*Figure 4:*

**Figure 4:** Extension is shown of the taxonomy of DOLCE basic categories in Figure 2. The extension is needed for definitional purposes, as shown in the text. Illocutionary points in (a) are understood as being kinds of communication processes, whereby we do not postulate herein the precise chain between the communication process and the illocutionary acts. The second extension shown in (b) highlights how certainty/uncertainty and belief/desire/intention/attitude distinctions are introduced as sub-categories of the DOLCE mental object. We do not postulate that beliefs, desires, intentions, and attitudes exhaust all mental objects.

**information source symbolizes-at-t cognitive representation,** which **stakeholder has-at-t.** The **cognitive representation is-a sufficiently certain mental object.**

Figures 4 and 5 show, respectively the extensions of DOLCE used in definitions discussed above, and concepts and relations made explicit in the definitions.

5 Requirement

Having established a classification of information sources, we can propose a definition of the ‘requirement’ and ‘assumption’ notions grounded in kinds of information sources.

**Definition 15.** Given temporal region t, **requirement** is-a directive information source or commissive information source or attitudinal information source, which represent-at-t one or more entity member-of-at-t environment.

**Definition 16.** Given temporal region t, **environment assumption** is-a assertive information source or declarative information source, which represent-at-t one or more entity member-of-at-t environment.

A requirement states what is desired, in accordance with the intuitive ideas usually associated to what a requirement is in RE. However, it also states attitudes, which can be informally seen as pointing out the strength with which something is desired, or comparing it to what is less desired. The requirement can therefore encompass preferences. We also take intentions to be requirements since they are not realized when stated; that is, as intentions involve commitment to bring about what is desired, they involve desires, and are therefore considered as requirements. Regarding Zave and Jackson’s ‘assumption’, we call it environment assumption and consider not only assertives, but also declaratives. Declaratives are by definition not requirements, since their very communication brings about what has been desired before

20
communication. Representative declaratives are neither requirements nor environment assumptions, but merely restatements of what are environment assumptions.

The proposed definitions are convenient for several reasons. They accommodate subjectivity and therefore the multiplicity of stakeholder views. What requirements and environment assumptions are in terms of communication is now clearer, as is their relationship with what are usually understood to be beliefs, desires, intentions, and attitudes. Notions of uncertainty and constraints are introduced at a level higher than requirements, highlighting thereby the need to study both requirements and environment assumptions in detail, and in similar ways. Distinction between what is informally understood as requirement and domain knowledge (i.e., environment assumption) is more precise. Definitions are grounded on more stable conceptual foundations than previously.

6 Classification of Requirements

As shown in the taxonomy for requirement in Figure 6, requirement is categorized as:

1. either functional or nonfunctional, and
2. either hard or soft, and
3. either neutral or preferred to one or more other requirements.

Figure 5: Partial conceptual model of the concepts and relations introduced in definitions is shown. The partial model above connects to that in Figure 1 as discussed in the text. Definitions of most of the concepts and relations in the figure are grounded in DOLCE (Figure 2) and its extensions introduced in this paper (Figure 4). The taxonomy obtained from is-a relations outgoing from information source is shown in Figure 3.
Figure 6: Taxonomy of requirement. Levels are interchangeable and their order given here is for illustration only. The distinction between functional and nonfunctional requirements does not give disjoint sets—as explained in the text, any nonfunctional requirement is always a functional requirement constrained by way of gradable adjective and within an attitudinal expression.

6.1 Functional vs. Nonfunctional

The distinction between functional and nonfunctional requirements in RE has not been particularly debated. Early suggestions on how to distinguish these two categories persist. For instance, van Lam-sweerde [69] borrows Keller, Kahn, and Panara’s [33] distinction which remains implicit or explicit and in unchanged form and interpretation in subsequent literature (e.g., [48, 6, 17, 39]):

“Functional [requirements] underlie services that the system is expected to deliver whereas non-functional [requirements] refer to expected system qualities such as security, safety, performance, usability, flexibility, customizability, interoperability, and so forth.”

We then consider the following to be functional requirements:

(Ex.20) \textit{Issue a ticket.}
(Ex.21) \textit{Book a flight.}
(Ex.22) \textit{Pay for the ticket.}

Continuing with the same functional/nonfunctional conceptualization, we call nonfunctional the requirements below (we say that they are requirements by assuming that they are expressed in optative mood):

(Ex.23) \textit{Quickly issue a ticket.}
(Ex.24) \textit{Conveniently book a flight.}
(Ex.25) \textit{Securely pay for a ticket.}

It appears intuitive that nonfunctional requirements constrain how well functional requirements need to be satisfied. Moreover, there is no nonfunctional requirement without a functional requirement—that is, we do not require “convenience” independently of stating what is expected to be convenient. This perspective is applied in any RE framework which uses the softgoal modeling concept [48, 6] to represent nonfunctional requirements. Above, we would identify the softgoal “Convenient” and relate the functional requirements “Book a flight” to it. The purpose of doing so is to rate alternative ways of booking a flight according to how convenient they are. Thus, we extract from nonfunctional requirements the criteria according to which we rate alternative sets of designs that satisfy functional requirements. That one accepts this, rather uncontroversial approach does not subsume that one is content with Keller, Kahn, and Panara’s definition of functional and nonfunctional requirements. Consider the following:

(Ex.26) \textit{Issue a cheap ticket.}
(Ex.27) \textit{Book a convenient flight.}
Above, we speak of “cheap” and “convenient”, which for all practical purposes we could represent as soft-goals and deal with in many RE frameworks. But are “cheap” and “convenient” here “system qualities” as Keller, Kahn, and Panara, and subsequently van Lamsweerde understand them? As they are not expected system qualities, but expected qualities of entities created/produced/manipulated by the system, they do not seem to be “system qualities” referred to in the accepted definition. This is problematic since our intention is to completely partition all requirements onto functional and nonfunctional ones. The last two examples would not be requirements at all if we take the given functional/nonfunctional distinction. Yet requirements they are for they are optative statements about the environment (of which the system is only a part, as are its outputs).

Having established that it is more appropriate to state that nonfunctional requirements refer to expected environment (as opposed to only system) qualities, we are interested in how they refer to qualities. To elucidate this relation, we need to understand what is meant by qualities. This in turn will help in giving a more precise definition of functional and nonfunctional requirement because the latter’s reference to qualities seems to be a trait that distinguishes it from the former.

The notion of system qualities has been around since Boehm and colleagues’ early work on software quality [3] (in which “qualities” include, e.g., utility, maintainability, and portability) and persist in recent software quality research (see, e.g., [12, 35, 52]). When speaking of such qualities, one usually employs expressions such as those in Ex.23–Ex.27, which have two salient characteristics. First, they express what we called here attitudes. Ex.23–Ex.25 are clearly explicit about attitudes: for instance, Ex.23 states that quicker than slower ticket issuing is preferred. Second, expressions of nonfunctional requirements always involve gradable adjectives, such as quick, convenient, secure, or useful, efficient, accurate, and so on. When performing semantic analysis of gradable adjectives, linguists usually make two assumptions (we borrow here from Kennedy [34], emphasis in the original):

1. Gradable adjectives map their arguments onto abstract representations of measurement, or degrees.

2. A set of degrees totally ordered with respect to some dimension (height, cost, etc.) constitutes a scale.

Linguists thus assume an ontology in which the analysis of gradable adjectives is performed. To localize the discussion in DOLCE and the extensions deployed herein, we observe that dimension corresponds to DOLCE quality, scale and degree to sub-categories of DOLCE abstract. The mentioned DOLCE sub-categories must obey the two conditions given above. A gradable adjective describes that a quality obtains to some degree, whereby the degree is at least as great as some standard of comparison, the standard itself being external to the adjective and determined by the context in which the adjective is used. If we take Ex.23, whether tickets are issued quickly involves a standard of comparison which is often local to each individual stakeholder, so that the same actual time for ticket issuing may be called quick by some stakeholder and not quick enough by some other. We can therefore consider that there is gradable quality within DOLCE quality, whereby gradable quality submits to the relations with regards to DOLCE abstracts stated in the two conditions above. Hence the following definitions.

**Definition 17.** Given temporal region t, nonfunctional requirement is-a attitudinal information source which represents-at-t gradable quality of one or more entity member-of-at-t environment.

**Definition 18.** Functional requirement is-a requirement which is not nonfunctional requirement. Observe that the notion of softgoal does not correspond to nonfunctional requirement, but to gradable quality. This corresponds to the typical ways in which it is used, as mentioned above.

### 6.2 Hard vs. Soft

This distinction is considerably simpler than the functional vs. nonfunctional one. Herein, the aim is to distinguish among requirements whose satisfaction is considered compulsory by the stakeholders, as opposed to optional. Hard requirements are therefore those that *must* be satisfied (otherwise the system is, *ceteris paribus* unacceptable to the stakeholders), whereas a soft requirement can remain unsatisfied by the system—the system will still be acceptable. We use the notion of general term ‘compulsory’ in the definitions below to characterize hard requirements as compulsory and soft as non-compulsory.
Definition 19. Hard requirement is-a requirement which is represented-by-at-t compulsory. Compulsory is-a general term.

Definition 20. Soft requirement is-a requirement which is not hard requirement.

Whether some requirement is compulsory or not depends on the information elicited from the stakeholders. We apply here the same reasoning as for the general term relevant used to define environment.

6.3 Neutral vs. Preferred to

A neutral requirement is one not compared to in attitudinal terms to some other requirement. Ex.28 below is not neutral since it is comparing two forms of flight tickets; it is attitudinal for it expresses which of the two is preferred. Earlier examples Ex.20–Ex.22 are neutral.

(Ex.28) Electronic flight tickets are preferred to paper flight tickets.

To accommodate this distinction, we introduce the basic primitive relation preferred-to-at-t, which corresponds to the intuitive notion of preference. In other words, requirement1 preferred-to-at-t requirement2 is understood as “satisfying requirement1 is preferred to satisfying requirement2 at the time period t”. Instead of considering such comparative information sources as requirements, we instead introduce the notion of requirement preference which is an information source that relates requirements with the preferred-to-at-t relation. Whether some requirement is then neutral or preferred to some other requirement is straightforward from requirement preferences. Recall that any information source can be conditioned (which is one way of constraining it), so that preferences conditioning is supported.

Definition 21. Given temporal region t, requirement preference is-a attitudinal information source which relates two distinct requirement with the preferred-to-at-t relation.

As preferences may be such that not all can be simultaneously satisfied, trade-offs appear. They are dealt with through priorities which intuitively correspond to preferences over requirement preferences. A preference priority is thus introduced as follows, using the basic primitive more-important-than-at-t relation. The relation obtains from requirement preference1 to requirement preference2 if there is at least one stakeholder who considers that satisfying requirement preference1 is more important than satisfying requirement preference2.

Definition 22. Given temporal region t, preference priority is-a attitudinal information source which relates two distinct requirement preference with the more-important-than-at-t relation.

6.4 From Information Sources to Requirements

Figure 7 shows a conceptual model representation of notions introduced in Definitions 15–22. Recalling that a requirement is an information source and that there are three classification dimensions of information sources and three classification dimensions for requirements, we observe that there is a considerable number of different kinds of requirement. In particular, observe that a requirement is either free or constrained, and either certain or uncertain. In this perspective, Figure 8 shows a decision tree which is suggested for determining, given an information source, whether it is a requirement or an environment assumption. If it is a requirement, we can use the decision tree to establish what kind of requirement it is. Given an information source, the first decision to make is to determine the kind of illocutionary point with which the communication is performed. This is sufficient to determine whether the information source is an environment assumption or a requirement. If it is an environment assumption we are concerned on whether it is free or constrained, and finally whether it is more or less certain. Regarding a requirement, we test for a nonfunctional requirement only in presence of attitudinal information source. To test, we use the indications in Definition 17. We then establish if it is neutral or preferred to—if it is the latter, we know that it is constrained and not free since a preferred to requirement is related to another requirement through a requirement preference. If neutral, we need to determine whether the requirement is free or constrained, and then its degree of certainty.

7 Revisiting the Requirements Problem

The usual understanding of the so-called requirements problem, that is, the problem that any RE effort intends to resolve is grounded in the relationship between domain/environment assumptions, require-
Figure 7: Partial conceptual model of CORE is shown. The requirement concept appears along the concepts relevant for the first and second levels of the taxonomy for the requirement concept. The above model connects to that shown in Figure 4, as explained in the text. The neutral/preferred to level of classification in the requirement taxonomy (Figure 6) is not ported to the conceptual model since it is derived from requirement preference and preference priority concepts.

ments, and the specification. We therefore first need to clarify what is the accepted meaning of the specification concept. Zave and Jackson [77] define the specification concept as follows:

“A specification is an optative property, intended to be directly implementable and to support satisfaction of the requirements.”

Recall how they define the requirement concept:

“A requirement is an optative property, intended to express the desires of the customer concerning the software development project.”

In this respect, both requirement and specification are optative, so that two distinguishing points remain: (i) a specification is implementable, and (ii) a specification, once combined with domain knowledge (our environment assumptions) satisfies requirements. This leads Zave and Jackson to suggest that, given domain knowledge $D$ and requirements $R$, the aim during RE effort is to find a specification $S$ which is implementable in principle and together with domain knowledge satisfies requirements, that is:

$$D, S \models R$$

They then go on to argue that:

“The point is that requirements, specifications, and domain knowledge always have the same relationship to each other.”

Requirements describe what is desired, while environment assumptions make explicit what currently is and will be the case after the system is introduced in the environment. In such a context, the purpose of a specification is to make explicit what is to be performed in the environment in order to obtain the

11They consider a specification to be implementable in principle. It may not be implementable in practice because it requires unavailable computing resources.
future environment, one in which whatever is stated in requirements holds and in which environment assumptions are not violated. Verifiability is therefore a key characteristic of any specification. With this claim we merely reiterate what is a widely understood and accepted characteristic of any specification. For instance, Spivey suggests that [67]:

“Formal specifications use mathematical notation to describe in a precise way the properties which an information system must have, without unduly constraining the way in which these properties are achieved.”

When combined with adequate operational semantics such as, e.g., Petri nets or process algebras, a specification written in a mathematical notation can be used for verification. Zave and Jackson’s suggestion that \( D, S \models R \) translates this accepted position. To verify that indeed \( D, S \models R \), \( R \) need to be stated in a sufficiently precise manner. While apparent, this observation entails that we cannot
consider gradable adjectives without a standard of comparison as Zave and Jackson’s requirements. In other words, information sources such as, e.g., Ex.23–Ex.27 are excluded from $R$: they are not precise enough for nothing therein states when exactly the issuing of a ticket becomes quick (Ex.23), when a booking becomes convenient (Ex.24), and so on. We see that $R$ excludes any nonfunctional requirement. Furthermore, if $D, S \models R$ verifies then all information sources in $R$ obtain. Consequently, all in $R$ is compulsory so that $R$ are by the very writing of $D, S \models R$ only cases of hard requirement.

We have shown that there are not only hard and not only functional requirements. Adopting CORE leads to reject the claim that all requirements, specifications, and domain knowledge always have the same relationship to each other, one given by $D, S \models HR$, whereby $S$ ensures that hard requirements $HR$ obtain if the system fully adheres to the specification. In response to optional, soft requirements $SR$, $S$ usually only partly ensures that these obtain so that the $\models$ relation does not apply. That is, given some limited (project) resources, $S$ should satisfy $SR$ as much as resources allow, but is unlikely to achieve optimal satisfaction. This becomes obvious when one recalls that nonfunctional requirements can be idealistic.

Since we cannot relate $D$ and $S$ to $SR$ with $\models$, we introduce the relation of optimizing, $\models_{opt}$. To optimize herein is to seek the optimum (i.e., ideal satisfaction of all soft requirements) given limited resources, while acknowledging that the optimum is probably not achievable and rejecting any satisfactory solution in favor of a better solution which fits the given resources. The intended interpretation for $\models_{opt}$ is that, given the resources available for the RE phase of system development, $D, S \models_{opt} SR$ verifies if and only if there is no $S'$ which, given the same resources, satisfies $SR$ to a more desirable extent than $S$. The second relevant relation to account is therefore $D, S \models_{opt} SR$.

Recall that requirement preference and preference priority are concepts derived from, respectively the relationship between distinct requirement and between distinct requirement preference. Moreover, preferences can be given both over pairs of hard and pairs of soft requirement. $D, S \models HR$ and $D, S \models_{opt} SR$ are incomplete with regards to preferences and priorities, in the sense that both hard and soft requirements can be preference ordered and priorities may be defined over preferences due to tradeoffs. If $Pf$ includes all requirement preference and $Pt$ all preference priority, then it is relevant to suggest that $D, S \models_{opt} HR, Pf, Pr$ and $D, S \models_{opt} SR, Pf, Pr$. A specification must therefore be adequate with regards to environment assumptions and stated preferences and priorities. But there is a further nuance here: all or some or none of $SR$ must be optimally satisfied, while all of $HR$ must be satisfied and this ideally optimally. This difference (which arises from $HR$ being compulsory and $SR$ optional) also must be evident. We therefore write $D, S \models_{opt, all} HR, Pf, Pr$ and $D, S \models_{opt} SR, Pf, Pr$. Knowing that $HR, SR \not\models$, we are led to conclude that the environment assumptions $D$, the preferences $Pf$, the priorities $Pr$, the specification $S$, and the requirements $R$ (which completely partitions onto hard $HR$ and soft $SR$ requirements) stand in the following relations to each other:

$$HR, SR \not\models$$

(1)

$$D, S \models_{opt, all} HR, Pf, Pr$$

(2)

$$D, S \models_{opt} SR, Pf, Pr$$

(3)

The above brings us to the following conclusion:

Given environment assumptions $D$, the aim of RE is to elicit hard requirements $HR$, soft $SR$ requirements, preferences $Pf$, and priorities $Pr$, and produce a specification $S$ such that, given the available resources:

1. $D, S \models_{opt, all} HR, Pf, Pr$, that is, $S$ optimally satisfies all hard requirements according to preferences and priorities;
2. $D, S \models_{opt} SR, Pf, Pr$, that is, $S$ optimally satisfies soft requirements according to preferences and priorities.

A question that arises is what it precisely means to say that “$S$ optimally satisfies soft requirements according to preferences and priorities”, that is, to write $D, S \models_{opt} SR, Pf, Pr$. For a simple illustration, take a first-order logic $L$ in which $\models$ is defined as usual. A simple way to accommodate preferences in $L$ is to add new kinds of formulas to the language of $L$: new formulas have the form $\phi \succ_{Pf} \psi$, where $\phi, \psi$ are always formulas that do not contain the operator $\succ_{Pf}$. One uncontroversial way to interpret the formulas
of the form $\phi \succ_{Pf} \psi$ is to add a function $v$ to the model in $L$. We take $v$ to map a real number (informally understood as a utility value) to formulas that do not contain $\succ_{Pf}$. We then evaluate $v$ in the following way: if $\phi \succ_{Pf} \psi$ then $v(\phi) > v(\psi)$, which means that $\phi$ is preferred to $\psi$. Priorities can be treated in the same way: we can straightforwardly introduce another function, say $p$ to the model and interpret over real numbers, as for preferences, whereby a higher real value would be informally understood as a higher priority. In this formal setting, $D, S \models_{opt} SR, Pf, Pr$ means that the optimal values of $v$ and $p$ are sought when writing the specification; that is, if both $S$ and $S'$ stay within the prescribed resources (both are realistic) and if $S$ gives higher aggregate values for $v$ and $p$ (and therefore “better” satisfies $SR$), then $S$ is chosen over $S'$.

8 Specification

We have noted above that a specification is implementable and satisfies requirements. In the present terminology, a specification is evidently a concretization of a cognitive representation. Since there can be no satisfaction without commitment, a specification must be a concretization of a cognitive representation which is an intention.

That a specification is implementable is usually understood to mean that there is very limited uncertainty as to (i) whether there are procedures whose execution satisfies requirements, and (i) whether these procedures can be successfully performed by the actors who commit to perform them. Both points highlight that there must be sufficient certainty in relation to a specification. This leads us to suggest the following definition for the specification concept in CORE:

Definition 23. Given temporal region $t$, specification is a representational artifact, which symbolize-at-$t$ intention. The intention is a sufficiently certain mental object. The intention refer-to-at-$t$ one or more entity member-of-at-$t$ environment.

The definition commits to no specification paradigm, leaving this choice to depend on the characteristics of the future system and relevant methodological considerations. It only highlights that the specification concretizes (some) certainty and commitment. The relationship between specification and requirement (§7) remains outside the definition.

9 Requirements and Goals

This section discusses the relationship between the important notion of goal in RE and the conceptualization of requirement in CORE. The question of how a goal fits within CORE is answered. We have proposed elsewhere [31] precise definitions for principal specializations of the goal concept, and reuse these results herein.

9.1 Goal Concept in RE Research

System development frameworks include, since the 1970s some form of analysis involving goals [69], among them context analysis, definition study, and participative analysis. Goals have become an essential part of any system’s documentation through standards such as e.g., the IEEE-Std-830/1993. There is no established definition for the goal concept: consider Table 1, which lists informal definitions of the goal concept appearing in various goal-oriented RE frameworks. KAOS highlights the nonoperational nature of goals, pointing to the need for taking action to make goals precise by refinement (see, e.g., [10]). Broadly speaking, the KAOS definition is in line with those of Tropos, $i^*$, GDC, and Lightswitch: a goal designates desirable conditions on the system and/or its environment. Such conditions restrict the set of alternative system and environment states, so that it is appropriate to say that a goal describes desired states. A different conceptualization appears in NFR, where goals are employed for representing nonfunctional requirements, in addition to design decisions, and arguments for or against other goals. We can interpret “design decisions” as restricting potential desired system and environment states. Notions of argument and justification appear in NFR and GBRAM. We have discussed elsewhere [29] the relevance of argumentation and justification for goal-oriented RE, arguing and illustrating that it is more appropriate to maintain the notion of argument separate from the goal concept.
Table 1: Informal definitions of the goal concept in goal-oriented RE.

<table>
<thead>
<tr>
<th>Framework</th>
<th>Informal definition of the goal (and derived) concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>KAOS</td>
<td>&quot;A goal is a nonoperational objective to be achieved by the composite system. Nonoperational means that the objective is not formulated in terms of objects and actions available to some agent in the system; in other words, a goal as it is formulated cannot be established through appropriate state transitions under control of one of the agents.&quot; [10]</td>
</tr>
<tr>
<td>Tropos and i∗</td>
<td>&quot;A goal is a condition or state of affairs in the world that the stakeholders would like to achieve.&quot; [73, 74, 5]</td>
</tr>
<tr>
<td>NFR</td>
<td>&quot;Goals [represent] non-functional requirements, design decisions and arguments in support or against other goals.&quot; [48, 6]</td>
</tr>
<tr>
<td>REF</td>
<td>&quot;According to the nature of a goal, a distinction is made between hard goals and soft goals. A goal is classified as hard when its achievement criterion is sharply defined [...]. For a soft goal, instead, it is up to the goal originator, or to an agreement between the involved agents, to decide when the goal is considered to have been achieved [...]. In comparison to hard goals, soft goals can be highly subjective and strictly related to a particular context; they enable the analysts to highlight quality issues [...] from the outset [...]&quot; [11]</td>
</tr>
<tr>
<td>GDC</td>
<td>&quot;An enterprise goal is a desired state of affairs that needs to be attained.&quot; [32]</td>
</tr>
<tr>
<td>GBRAM</td>
<td>&quot;Goals are high level objectives of the business, organization or system. They capture the reasons why a system is needed and guide decisions at various levels within the enterprise.&quot; [2]</td>
</tr>
<tr>
<td>Lightswitch</td>
<td>&quot;[A] maintenance goal is said to represent a condition that remains constant. [...] [An] achievement goal has definite pre and post-conditions. The pre-condition represents the interpretation that the state of affairs has drifted (or will drift) outside of the threshold associated with the norm [i.e., a variable of the system whose state the system attempts to maintain unchanged as defined by an observer]. The post condition is an interpretation that is within this threshold.&quot; [55]</td>
</tr>
</tbody>
</table>

Regarding the use of goals for modeling nonfunctional requirements, two relevant goal taxonomies have been introduced since the seminal contributions in the NFR framework (see, e.g., [69, 30] for discussions). Functional goals are distinguished from nonfunctional ones, and softgoals from hard goals. Functional goals have been used to represent services that the software is expected to deliver (i.e., what the software does), whereas nonfunctional goals refer to quality requirements that the software needs to satisfy while delivering the services (i.e., how the software provides services; e.g., securely, safely, rapidly, etc.). While it is common to equate nonfunctional goals and softgoals (e.g., [48]), it has been subsequently argued that softgoals belong to another taxonomy, in which they are distinguished from hardgoals [69].

According to the traditional definition, “a softgoal is similar to a (hard) goal except that the criteria for whether a softgoal is achieved are not clear-cut and a priori.” [40] The definition used in the REF framework [11] adds details, as shown in Table 1.

While softgoal satisfaction cannot be established in a clear-cut sense [48], the satisfaction of a hardgoal is objective in that it can be established using (formal) verification techniques [10]. In this respect, a hardgoal is said to be achievable, whereas a softgoal is satisficeable [48, 6, 39]. The concept of satisficing originates in H. Simon’s work [62] in economics: to satisfice is to set a threshold, and accept any achievement above the threshold. In addition to involving satisficing, a softgoal has a subjective component, in that various stakeholders of the system will have different thresholds. We have worked on a more expressive definition of softgoals elsewhere [30], but we did not provide a mathematical definition.

The KAOS framework provides the most precise hardgoal conceptualization: a hardgoal is defined in terms of predicate patterns in a discrete linear temporal first-order logic (see, e.g., [42]). A hardgoal

---

12 This paragraph follows our previous discussions on the subject [30].

13 In publications on KAOS, what we call hardgoal here is called simply “goal”. Note, however, that this conceptualization
is any one of the following [38]: an achieve hardgoal (pattern: $\phi \Rightarrow \psi$), a cease hardgoal ($\phi \Rightarrow \neg \psi$), a maintain hardgoal ($\phi \Rightarrow \Box \neg \psi$), an avoid hardgoal ($\phi \Rightarrow \Box \neg \psi$). The same conceptualization is adopted in Formal Tropos [17], where patterns are used in the same way to define hardgoals. An informal interpretation of the said conceptualization is that a hardgoal is a constraint over system histories (i.e., behavior over time).

It is clear that the goal concept is intended to be rich in meaning. Instead then of seeking an all encompassing definition, we study derived concepts, obtained by crossing the hardgoal/softgoal and functional/nonfunctional taxonomies; we thus have: (i) functional hardgoals, which are hardgoals about what the system should do (e.g., in an email application, such a goal can be: “whenever an e-mail marked as important arrives, the user is informed with a pop-up window and a sound”); (ii) nonfunctional hardgoals which describe verifiable criteria for how the system should operate (e.g., “the user should be informed about important e-mail arrival within 1 second of arrival”); (iii) functional softgoals describe a subjective requirement of a stakeholder about what the system should do (e.g., “whenever an e-mail marked as important arrives”); and (iv) nonfunctional softgoals which indicate in a subjective and nonverifiable manner how the system should operate (e.g., “the user should be informed rapidly about the arrival of an e-mail marked as important”).

In summary, there is no unique definition of goal. One reason for this is that the goal concept is intended to be rich in meaning. Variations in definitions are also due to slightly different uses of the concept in each framework. Whether a goal conceptualization is appropriate depends on how useful is the framework in which it is used. A prescriptive general definition thus seems excluded. It remains, however, of interest to seek a conceptual framework in which the derived concepts mentioned above can be used together, so that the benefits of these complementary concepts can be combined when representing and reasoning about goals. A precise definition is already available for the hardgoal concept. We can now suggest a common ground for the cited derived concepts.

### 9.2 A Common Framework for the Goal Concept

Consider a toy system that has only two properties, $p_1$ and $p_2$. All possible combinations of allowed values for $p_1$ and $p_2$ define all possible states of the system. Let $S_1$, $S_2$, and $S_3$ be arbitrary system states, as shown in the bottom part of Figure 9. Assume that measurements are performed on the system in order to evaluate its quality. To perform measurement, we define two metrics $d_1$ and $d_2$. To relate what we observe in the system and the values of the metrics, we define mappings $M_1$, $M_2$, and $M_3$ between system states and value combinations of the two metrics. Since some minimum level of quality is expected, we define thresholds on metrics: in Figure 9, $t_1 \equiv d_1 \geq d^*_1$ and $t_2 \equiv d_2 \geq d^*_2$, so that the quality is above the minimal level only when the system is in state $S_2$ and not in the other two.

Taking the state-based conceptualization of the hardgoal concept, we define a hardgoal $hg_1 \equiv (\top \Rightarrow \Box (p_2 = p^*_2))$ as a value $p^*_2$ of the property $p_2$. $hg_1$ is a functional hardgoal, since it says precisely what the system is expected to do (i.e., set property $p_2$ to value $p^*_2$). Returning to the informal understanding of the nonfunctional hardgoal given earlier (§9.1), we see that it cannot be defined over system properties, but on metrics defined for the system. We can thus define two nonfunctional hardgoals in Figure 9: $\tilde{hg}_1 \equiv (d_1 \geq d^*_1)$ and $\tilde{hg}_2 \equiv (d_2 \geq d^*_2)$.

---

Figure 9: Arbitrary functional and nonfunctional hardgoals in a toy system.
Are there any softgoals in Figure 9? We know that a softgoal is used to model requirements at the earliest stages of an RE process (e.g., [48, 69, 5, 39, 30]). Usually, initial requirements, and consequently softgoals are imprecise, subjective, idealistic, and context-specific [30], meaning that we cannot have a softgoal in Figure 9—the figure is already too precise. Consequently, and in line with contributions in the RE field, a softgoal is here understood as an initial, early form of requirements about what the system should do and how “well” it should do it, from which one or more functional and/or nonfunctional hardgoals are extracted during the requirements process. This is appropriate, since we also know that one cannot manage (i.e., assure, control, or improve) what one cannot measure (e.g., [15, 35, 4, 14, 20]): if quality-related information contained in softgoals is to be used in decision-making during an RE process, we need to make a softgoal precise, agreed upon by various stakeholders, and realistic—that is, we need to convert it into nonfunctional hardgoals. Same applies for functional softgoals: we require precise, agreed upon, and realistic requirements about what services the system should deliver in order to be able to implement them in later stages of the system development process.

Having established that softgoals appear earlier in an RE process than hardgoals, recall that softgoals are associated to the concept of satisficing. Satisficing is the reason we specified our nonfunctional hardgoals as thresholds only, instead of, e.g., more elaborately stating the desired values of \( d_1 \) and \( d_2 \). Indeed, nonfunctional hardgoals derived from nonfunctional softgoals serve in RE as criteria for comparing alternative system structures (e.g., [48, 6, 69, 5]). A system structure is chosen over an alternative one if the former dominates the latter over a set of nonfunctional softgoals or the derived nonfunctional hardgoals. In our toy system, we would choose a system structure that is associated to higher values of the two metrics, over one associated with lower values; we would discard structures that are not above both thresholds. Satisficing, while clearly useful and reflecting the inability to identify the optimal system structure, does not cover requirements in which continuous improvement is sought. Indeed, satisficing does not go as far as to say what values above a threshold are preferred over other values, also above the threshold. That is, all values are equally desired, provided that they are above the threshold. In many actual cases, we do need to set thresholds, but we need not equally prefer all above-threshold structures. This is the case in particular for adaptable systems based on the agent or services paradigms. We encountered this need in an actual setting: we proposed elsewhere [28] an adaptable system in which above-threshold structures are learned. Therein, a “system structure” corresponds to a composition of web services that allows a service request (coming from a system user and specified in terms of requirements) to be filled. To form compositions, a composer web service observes other web services during execution and subsequently selects (for participation in a composition) only those that allow it to obtain more desired values over a given set of metrics. Compositions are revised, so that the quality to which same service requests are fulfilled improves over time. When specifying requirements on such a system, we are clearly not interested only in satisficing—if we did rely on satisficing only, we would not exploit the ability of the system to improve the fulfillment of service requests. We would not exploit the system’s ability to adapt. Instead, we need to express that the system both needs to satisfice (so that below-threshold compositions be discarded) and to always improve compositions. We clearly cannot use the notion of satisficing to express requirements on continuous improvement: instead, we use the notion of excelling to do so. The limitation of satisficing that we highlight here is not novel: recently, J. L. Pollock proposed the concept of locally global planning, in which “any plan with a positive expected utility is defeasibly accepted, but only defeasibly. If a better plan is discovered, it should supplant the original.” [54] This discussion brings us to the following position: to use the concept of softgoal grounded in satisficing to express requirements that are associated to the concept of excelling is to extend the softgoal concept too far. We thus propose the concept of \textit{disposition}. A disposition is a preference order defined over goals of the same type; we thus have the following taxonomy for the disposition concept: (i) hard-functional disposition, defined over functional hardgoals; (ii) hard-nonfunctional disposition, over nonfunctional hardgoals; (iii) soft-functional disposition, over functional softgoals; and (iv) soft-nonfunctional disposition, over nonfunctional softgoals. An example of a hard-nonfunctional disposition expressed informally is: “the user should be informed about important e-mail arrival within the least time possible” generalizes a preference order in which it is clear that the nonfunctional hardgoal “the user should be informed about important e-mail arrival within 0.5 second of arrival” is preferred to “the user should be informed about important e-mail arrival within 1 second of arrival”. Note the following:

- Do not mistake excelling for optimization: the latter applies if the email system is designed so that it always gives the optimal time (i.e., 0 seconds). This is clearly idealistic. Excelling is in a
sense optimization over time and given resource boundedness; that is, the email system excels if it reduces time for informing the user at each email arrival compared to the time it needed on the last occasion an email arrived. Excelling applies even if the system does not continually improve (expecting this may be idealistic); what is important for excelling to apply is that, even if, in our example email system, notification time increases, it restablishes and goes down at some later and observable point (i.e., not indefinitely in the future).

- Not always can be a disposition so summarily expressed as in our notification time example for the email application. It may for instance happen that disjoint subsets of metric values are preferred, so that a disposition does not reduce to a formulation of desired direction for metric values. We explore in the remainder a simple notion of disposition mainly because we are introducing the concept here—extensions to its expressivity are of interest in current and future effort.

To express our various types of goals, we start from the multi-sorted first-order version of MITL [1, 21], a continuous real-time linear temporal logic. Some predefined sorts (e.g., real numbers) have a fixed interpretation that will be used to express metrics. Our logic starts from $\Phi$, a first-order vocabulary, consisting of predicate and function symbols $p,f$. They can be declared as flexible (time-dependent) or rigid. As usual, constant symbols are viewed as 0-ary rigid function symbols. Starting with atomic connectives (i.e., $\land, \lor, \neg$, and $\rightarrow$)\textsuperscript{14}, temporal operators (i.e., next $\bigcirc$, eventually $\diamond$, always $\Box$, until $U$ and unless $\mathcal{W}$) that can be indexed by a non-singular real-time constraint, existential ($\exists$) and universal ($\forall$) quantification. We denote the resulting language $\mathcal{L}$. We interpret formulas of $\mathcal{L}$ over the structure $T \equiv (D_\mathcal{S}, S, \pi, \mathcal{H})$, where $D$ gives a domain to interpret each sort, $S$ is a set of states of the system, $\pi$ is an interpretation assigning each predicate symbol and function symbol in $\Phi$ a predicate or function of the right arity over $D$, if the symbol is declared rigid. If the symbol is declared flexible, it depends furthermore on the current state. $\mathcal{H}$ is a set of timed state sequences, $S_0, I_0, S_1, I_1 \ldots$ i.e. an infinite sequence of states and their associated interval of time, representing all possible executions of the system. These intervals $I_i$ must partition the positive reals. To interpret first-order variables, we use a valuation function $\sigma$, which, given a variable of sort $s$ returns its value, an element of $D_s$. Given a structure $T$, an history $h$, a time $t$ and a valuation $\sigma$, we can associate with every formula of $\mathcal{L}$ a truth value in the usual way. A formula holds in a structure if it yields true for all histories, valuations and times. We call the obtained logic $\mathcal{L}_\mathcal{L}$. We can now give a precise definition of functional hardgoal.

Definition 24. A functional hardgoal is a formula in $\mathcal{L}_\mathcal{L}$ that restricts the possible histories of a given system only to those desired by system stakeholders.

To express the nonfunctional hardgoal concept, we use metrics. A metric is a rigid function symbol, which will return values in a sort equipped with an order.

Definition 25. A nonfunctional hardgoal is a formula in $\mathcal{L}_\mathcal{L}$ that restricts the values of metrics to those desired by system stakeholders.

To relate the metrics and the behaviour of the system (recall Figure 9), we also need mappings between functional and nonfunctional hardgoals.

Definition 26. A hardgoal mapping is a formula in $\mathcal{L}_\mathcal{L}$ over one or more functional hardgoals and one or nonfunctional hardgoals.

Taking the email application example, the following is a functional hardgoal, and is followed by an equivalent nonfunctional hardgoal:

\[ hg \equiv [\forall m : Email (arrived(m) \land important(m)) \Rightarrow \bigcirc_{\leq 1sec} \exists w : PopupWindow, s : NotifSound (display(w) \land play(s))] \]

\[ hg \equiv [timeToNotification \leq 1sec] \]

We can then define a hardgoal mapping for the above as follows: $hg \equiv \tilde{hg}$.

In contrast, we understand softgoals as expressing dispositions, i.e. preference over goals of the same type. To accommodate dispositions, we extend $\mathcal{L}_\mathcal{L}$ in the following way. First, we add a new kind of formulas, disposition formulas, such that if $\phi$ and $\psi$ are formulas of $\mathcal{L}$ then $\phi \geq_d \psi$ is a disposition formula.

\textsuperscript{14}Usual abbreviations apply, e.g., $(\phi \Rightarrow \psi) \equiv \Box(\phi \rightarrow \psi)$. 

32
We take here the simplest case, in which we do not allow the operator \( \geq_d \) to appear in, e.g., temporal formulas of the language. \( \mathcal{L} \) extended with disposition formulas is denoted \( \mathcal{L}^{\geq_d} \). Second, we extend our structures \( T \) to interpret defeasible formulas: similarly to our earlier discussion (§7), we add a function \( v \) which maps a real number (informally understood as a utility value) to non-disposition formulas. \( v \) is then evaluated as follows: if \( \phi \geq_d \psi \), then \( v(\phi) \geq v(\psi) \), which means that \( \phi \) is preferred to \( \psi \).

Following the earlier example, we may have the following hard-nonfunctional disposition:

\[
[\text{timeToNotification} \leq 0.5 \text{sec}] \geq_d [\text{timeToNotification} \leq 1 \text{sec}]
\]

We define a disposition in terms of hard and soft disposition as follows:

*Definition 27.* A hard disposition is a disposition formula in \( \mathcal{L}^{\geq_d} \) between hardgoals.

*Definition 28.* A soft disposition is a preference either between only functional softgoals or between only nonfunctional softgoals.

### 9.3 Goals in CORE

We arrived above to definitions of the concepts derived from the goal concept, namely functional and nonfunctional goal, hardgoal, and softgoal. The concept of satisficing, associated to softgoals has been revisited. We established that a softgoal is satisficed when thresholds of some precise criteria are reached. Satisficing does not cover situations in which continual improvement of thresholds is expected. The notion of *excelling* has been suggested to cover such cases, along with the concept of *disposition* to represent and reason about Excelling.

Returning to Definitions 24 and 25, we observe that desirability is a key characteristic of a hardgoal. We can therefore perceive a hardgoal as a concretization of a cognitive representation which is a desire, so that a hardgoal can be conceptualized in CORE as a specialization of *requirement*. That is, a hardgoal is a directive information source. We cannot speak of it as being a commissive since there is no explicit idea of intention in a hardgoal. If the reader considers that an RE goal is always an intention, then she will take a hardgoal to be a commissive information source and thereby still a specialization of *requirement* in CORE. Dispositions evidently correspond to *requirement preference* in CORE. Given our discussion of the softgoal concept above and the conclusion that it is used to express early, usually imprecise information, and given its common use in RE, we can reasonably argue that it involves attitudes. We therefore take the nonfunctional softgoal to be a specialization of *requirement* whereby it is a concretization of an attitude.

### 10 Conclusion

We have shown at the outset of this text that the requirement concept can be variously interpreted. We have subsequently argued that some precision is needed. We built conceptual foundations for increasing the precision of the concept and its closely associated concepts commonly used when engineering requirements. Conceptual foundations rely on established results in ontology research and development, philosophy, and linguistics. Definitions of the requirement and associated notions were then presented and their merits and limitations, as well as implications discussed. Together, the various definitions constitute the proposed Core Ontology for Requirements Engineering (CORE). Finally, we discussed the purpose of the RE effort, refined the current perspective thereon, and related the commonly encountered notion of goal in RE to CORE.

### References


