Jurnal Teknologi

Full paper

Self-Explaining Agents

Johannes Fähndrich[a*], Sebastian Ahrndt[a], Sahin Albayrak[a]

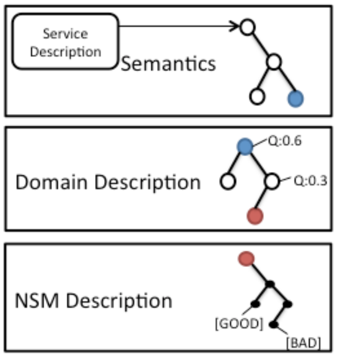
[a]*Faculty of Electrical Engineering and Computer Science, DAI-Labor, Technische Universtität Berlin, Ernst-Reuter-Platz 7, 10587 Berlin, Germany*

*Corresponding author: johannes.faehndrich@dai-labor.de

Article history

Received :11 September 2012
Received in revised form :
21 February 2013
Accepted :15 April 2013

Graphical abstract



Abstract

This work advocate self-explanation as one foundation of self-* properties. Arguing that for system component to become more self-explanatory the underlining foundation is an awareness of themselves and their environment. In the research area of adaptive software, self-* properties have shifted into focus caused by the tendency to push ever more design decisions to the applications runtime. Thus fostering new paradigms for system development like intelligent and learning agents. This work surveys the state-of-the-art methods of self-explanation in software systems and distills a definition of self-explanation. Additionally, we introduce a measure to compare explanations and propose an approach for the first steps towards extending descriptions to become more explanatory. The conclusion shows that explanation is a special kind of description. The kind of description that provides additional information about a subject of interest and is understandable for the audience of the explanation. Further the explanation is dependent on the context it is used in, which brings about that one explanation can transport different information in different contexts. The proposed measure reflects those requirements.

Keywords: Self-*; self-explanation; agent-capability descriptions; self-explanatory descriptions; NSM; pragmatic

■1.0 INTRODUCTION

In nowadays computing environments where different parties at different times are allowed to make use of different technologies it seems to be necessary to move evermore details from the application design time to the application runtime. This trend, which is a consequence of the arising complexity crisis [1], can be supported with applications possessing a set of self-* properties, where the initial set is known as self-CHOP (configuration, healing, optimization, protection). As these are the "big four", several researchers have begun to investigate the requirements and in consequence introduced more self-* properties refining the initial set [2]. One of these basic properties is *Self-Explanation*, which can be seen as a prerequisite for self-configuration [3]. Self-explanation is the capability of a system to provide information about itself and its functionalities. Of course, the term system comprises not only the whole system but also its components. Yet, providing information about functionalities is only the first step towards self-explaining systems as there must be the ability to consume and interpret that information as well.

The goal of this work is to foster the understanding of the self-explanation property, with a special focus on multi-agent systems. Here, self-explanation is the ability of an agent to describe its capabilities to other agents in order to enable them to autonomous reach the system goal, i.e. using planning techniques to do so. Typically, planning agents have the ability to solve problems autonomously by creating a plan (a sequence of actions) that reaches a desired goal state. In such a plan, agents can include capabilities of other agents. To elude a brute-force approach on trying every combination of available capabilities or in other words to reduce the branching factor of the search space, answers to the following questions are of interest:

- Which functionalities does an agent provide?
- How and under which conditions can another agent use these functionalities?
- What is the expected outcome of the provided capability?

An agent has to be able to reason upon the information given to it, to decide if a given action is helpful in regard of achieving an active goal. In this work we reduce the agent to its reasoning capability, since the execution, the plan creation, communication aspects and other details of agent systems are out of scope of this work. Henceforth we will refer to these agents as *reasoner*, where the term reasoner will be explained in more detail in the following section, which also provides an overview about the research field (See Section 2.0).

The rest of the work is organized as follows. In addition and as an extension to a prior work [3] we introduce a formal definition of self-explanation and a measure enabling to decide which description is more self-explanatory (See Section 2.3). Subsequently, as self-explanation requires to formulate an abstract description of the system components Section 3.0 introduce the term self-explanatory descriptions. Afterwards, the work proceeds with an approach on creating self-explanatory descriptions that, in contrast to classical descriptions, provide additional semantically and contextual information in a structured and computer readable manner as proposed by Oaks *et al*. [4] (See Section 4.0). This approach investigates the applicability of the Natural Semantic Metalanguage [5] (NSM), a theory from linguistic, which introduces a bag of sematic primes able to represent all expressions producible in a natural language. Finally, Section 5.0 concludes how all these pieces fit together and elaborates on future work.

■2.0 SELF-EXPLANATION

There is a German saying that translates to: “to understand something, you have to be able to explain it”. Here, one might notice that the activity of explaining something comprises the understanding of a subject of interest (SoI) as well as the ability to describe how this SoI works (or at least to convey all the information about a SoI available). We can distinguish two entities involved into the activity of explaining something: The explaining entity, which is the producer or the provider of the explanation and the audience of the explanation, which is the reasoner or consumer of the explanation. One interesting fact is that the former and the latter could be the same entity. In this context, self-explanation is defined as “activity of explaining to oneself in an attempt to make sense of new information, either presented in a text or in some other medium” [6]. Commonly, explaining events, intentions and ideas is a well-known way of communicating information in everyday life. On the one hand, the explaining entity (the producer) is able to impart knowledge to some audience. On the other hand, the audience (the reasoner/consumer) is able to understand and comprehend the explainer's intentions and they may even understand the explainer's course of actions.

The ability of the consumer to learn from a given explanation can be seen as a major part of our adaptability as humans. It is only natural that we want our technology to be able to do the same. The research area of Artificial Intelligence (AI) studies amongst others this ability, which is the ability of machines to learn. This work can be seen as part of this research focusing on the explanations of capabilities of artificial agents for an audience of other artificial agents.

In the following we want to carve out the term self-explanation giving an overview of the research field, a definition of the term self-explanation and a formal model for a measure enabling to decide which description is more self-explanatory. This measure differs from the existing ones in terms of the point of view an explanation is rated and in consequence follows the idea that the currently available description must be enriched with semantically and contextual information. Finally we will give an overview of currently available measure to carve out the difference to ours in more detail.

2.1 Formal Explanations

Several definitions of explanations have been proposed. Each one specialized for the needs of some domain. We will look at some of them to see how they can help defining the term.

J.A. Overton [7] presents a philosophical approach to explanations, which can be described in a computable manner. Different classes of explanations are presented:

- Design/causation
- Syllogism/instantiation
- Modeling
- Argumentation/justification

Explanations are defined in a working definition as answer to Why-questions. J.A. Overton demonstrates an explanation via a type system implemented in Haskell[1]. A working definition of an explanation is given as well:

> *“An explanation is the pair of an explanans A and an explanandum B, such that there exist a why/how-question Q with B as its presupposition, and A explains B.”* [7, page 44]

Furthermore, the work explains what a scientific explanation might be and how it can be formulated but lacks to introduce or discuss a structure for such machine readable descriptions.

In AI expert systems a definition of explanations is a topic of research as well. Moore and Swartout [8] introduce an expert system that is able to engage a dialog while explaining some system state. By using static hand written explanation they are far from having semantics or any other understanding of the explanation from the machine, i.e. the provided explanations are not able to give information about the current state of the system, which is required for the superior goal of self-adaptive systems [9].

Heckerman *et al.* [10] describe explanations as a Bayesian believe network. A variable, which is the subject of interest, is explained by its predecessor in a Bayesian believe network. Each predecessor then influences the variable to a certain extend. An explanation then can be seen as an evidence weight, representing the logarithmic likelihood ratio of the influence of an observation on a variable in a Bayesian believe network. An example is shown in Figure 1.

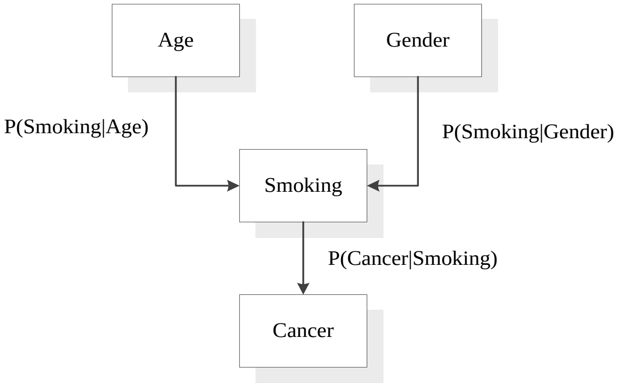


Figure 1 Example of a Bayesian believe network explaining the influence of different factors through getting cancer

[1] The Haskell Programming Language – For more information visit: http://www.haskell.org/haskellwiki/Haskell

The example, which is a formal representation for an explanation, can be seen as an explanation for getting cancer.

In AI research this formalism is used to describe probabilistic models and here we can inference that the event smoking and so forth influence the probability of the event having cancer. Thus this model can represent multiple layers of explanations in a probabilistic manner. W.G. Cole [11] uses such a graphical representation of Bayesian belief networks, to create a mental model for the Bayesian belief update. Heckerman *et al*. [10] describe the reasoning on probabilistic explanation from a decision-theoretical point of view. Here expert systems are combined to reach a decision using Bayesian beliefs updates by using an "odds–likelihood updating scheme" like the one described above.

M.J. Druzdzel [12] separates explanations in two categories: *Explanation of assumptions* focusing on the communication of the domain model of the system and *explanation of reasoning* focusing on how conclusions are made from those assumptions. In this separation the author describes on the one hand explanations transmitting assumptions about the world, like a domain model to have a common language. This model is a diagram in which nodes represent assumptions. On the other hand, the explanation of reasoning described by the edges of the diagram, explain the inference on how the a-posteriori probability is changed by an observation. It might be worthwhile to transfer these categories to self-explanation since the meaning of concepts used might differ depending the exogenous or endogenous origin of the fact explained. Therefore the reasoner has to distinguish between the explanations of the system itself and how it can be interpreted related to the current context.

This work focuses on the explanation of assumptions, since the audience of such a description is seen as an external system component reasoning for itself. Nevertheless, in multi-agent systems this restriction implies no loss of generality. That is, since autonomous agents typically reason for them self by observing the environment with sensors and influence the environment using available actors. Further the agents might be developed by different parties, working in different domains and having different contexts in mind while developing their agent applications.

2.2 Towards a Definition

Going back to the initial set of self-* properties one can imagine that self-explanation injects momentum not only to the self-configuration but also to the other properties. Indeed, these properties cannot be considered independently. Consequently, the term self-explanation has different meanings, too.

The information that is intended to be transported with the explanation is sometimes called *explanandum*. The explanadum typically holds some information about the SoI. The explanation itself is also sometimes called *explanans*. Thus an explanation given by one entity might contain an explanans. An example explanadum could be: "Why is this room unsecure?" A fitting explanans could be: "Because the door is open."

Taking into account the different parties involved – agents (the system itself), developers and (end) users – we can distinguish between two sides of self-explanation.

To start with, we can refer to the system side with the goal to integrate agents autonomously into existing infrastructures [1] [13]. Following the idea of self-explanation this means that agents are able to learn the capabilities of each other and to comprehend in which way they are able to interact (e.g. which data format and concepts match). One can imagine this process in the way a new human introduces itself into a prior unknown group of other humans, e.g. a team to solve some work related problem, by explaining its name and capabilities. Consequently, the system-side self-explanation is concerned with explanations to be used by artificial reasoners. Thus the descriptions are optimized to being computer readable.

Furthermore, we refer to the human side as self-explanation aiming to integrate the user (a human agent) into the system consisting of artificial agents as well as other human agents. As those systems are typically goal-driven, one example of humans interaction would be that the human can set the pursuit goals, to restrict the systems resources or other parameters using constraints and to observe the results of an otherwise autonomous process [13] [14] [15].

Taking both sides together the goal is that agents are able to learn about the capabilities of each other to the extent of having enough information to make use of them. The following definition for the term self-explanation is proposed [3]:

> *"Self-explanation identifies the capability of systems and system components to describe themselves and their functionalities to other systems, components or human beings."*

2.3 A Measure for Self-Explanation

Explanation of assumptions might informally be defined as a description to reveal the identity of some subject of interest. This might for example include information about its functionality. Imagine that we want to identify different boat types for tax reasons. We might not use the appearance to identify the difference of a rowing-, sailing- and a motor boat, because there might be different appearances in each class of boats. Instead, to identify the different boat classes, we need to describe some other details like the propulsion method and the tonnage of the boat. In contrast, if somebody wants to describe the different boat types to a child the functionality might be the detail separating the identities. In AI this fact is well known, since we seek different metrics to decrease intra class scatter and increase inter class scatter [16, page 121]. Furthermore, the explanation depends not only on the context but also on the reasoner who infers about it. With this in mind, an explanation should help the audience, to identify the classes a SoI might be part of and with that better describe its identity to foster understanding of the explanation whereas the understanding determines the goodness of an explanation [17]. To rate this goodness a measure for explanations is required. Roughly speaking one will need a way to rate if a self-explanation capability is available from the point of view of the reasoner. Indeed, this point of view constitutes the difference between the measure presented in the following and the measures available in the related work presented afterwards.

2.3.1 Abstract Measure

To determine the quality of an explanation and in consequence to ease the creation of measureable properties of explanations, we will now formalize a measure. As mentioned above, we define the amount of information transferred to the audience as a measure of quality of an explanation.

First we want to define a domain as a set of information concerning this domain:

Definition 1. The information available in one domain d is defined as the set $\mathbb{D}_d$ with $\mathbb{D}_d \subset \mathbb{I}$ and $\mathbb{I}$ being the set of all information.

Here, the basic assumption we follow is, that in computer science where information is digitalized, information is a discrete entity. For example the chess move "Qxd4" (e.g. as move in the center game of a Danish Gambit) in the domain of playing chess is one piece of information $i \in \mathbb{I}$ in the domain of chess $\mathbb{D}_{chess}$ – where $\mathbb{I}$ is the amount of information available and $\mathbb{D}_d$ is the formal description of a domain as a proper subset of the information space $\mathbb{I}$. Consequently, a domain $\mathbb{D}_d$ contains those information necessary to create fully observable planning for the given domain. Here, planning as the reasoning side of acting [18] and one inherent part of artificial agents, which are typically goal driven and try to achieve their goals autonomously.

As illustrated in the boat example, the quality of explanations depends on the reasoner who infers about this explanation. As this point of view is one important part of our measure, we now need to define what a reasoner is. The following definition expresses what a reasoner is:

Definition 2. Given a set of explanations $\mathbb{E}$ and a domain d, a reasoner for d and e, $r := (I_r, \iota_r)$ is defined as an entity which integrates a new explanation $e \in \mathbb{E}$ into its knowledge-base $I_r \in \mathfrak{D}$ using the function $\iota_r\colon \mathfrak{D} \times \mathbb{E} \rightarrow \mathfrak{D}$ where $\mathfrak{D}$ is a σ-Algebra over the information $\mathbb{D}_d$ available in the domain d.

This does not mean that all elements of $\mathbb{D}_d$ are available to each reasoner *r*. This offers the advantage that reasoners are able to infer in both fully and partial observable problems. Indeed, the typical agent application is located in partial observable environments and requires capabilities enabling to achieve given goals under uncertainty [19]. To elude the problem of domain overarching knowledge, we define a domain as a σ-Algebra introducing the characteristic that all unions of information of one domain with information of the same domain are always part of the domain again. This could e.g. happen if two agents share their knowledge in one domain. Later on we will use this and other characteristics of σ-Algebras as important properties for our measure.

Now, let $\mathfrak{R}$ be a σ−Algebra of sets over all reasoners of concern R. Reasoners of concern are the reasoners which make up the audience of an explanation. Further let $e \in \mathbb{E}$ be an explanation in some domain $\mathbb{D}_d$. Then we can define how an explanation maps to information by defining how the information in an

Definition 3. Given a domain d and a set of explanations $\mathbb{E}$, an information $i \in \mathbb{D}_d$ is contained in an explanation $e \in \mathbb{E}$ for a set of reasoners (the audience) $A \in \mathfrak{R}$, iff all reasoner $r \in A$ are able to integrate i into their knowledge-base ($i \in \iota_r(I_r, e)$), written $e \underset{r}{\longrightarrow} i$

Integration of new information into a knowledge base can be seen as the union $I_r = I_r \cup i$ of the new information with the knowledge base I_r, without destroying the consistency of I_r. With this definition an explanation holds and transmits information to an audience if a reasoner of the audience can integrate new information into its knowledge-base. To avoid a philosophical discussion, we define that an explanation has to be understood by someone. Now that we have some definitions about explanations, we will look at self-explanation to determine more specifically what exactly an explanation is.

The dictionary defines *self-explanatory* as "easily understood from the information already given and not needing further explanation" [20]. This definition leads to the conclusion that the information given by self-explaining descriptions is sufficient for some reasoner in the audience to understand the subject of interest and that the explanation is given by the entity representing of encapsulating this subject. Taking this definition into account, we define a *degree of explanation* as follows:

Definition 4. Given a set of explanations $\mathbb{E}$, a set of reasoners $A \in \mathfrak{R}$ and a domain d, $\mu\colon \mathfrak{E} \rightarrow \overline{\mathbb{R}}$ is a measure to an affine extension of the real numbers $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty, -\infty\}$, where $\mathfrak{E}$ is the σ-algebra over $\mathbb{E}$. μ is defined as:

$$\mu(E) := \sup_{\substack{r=(I_r,\iota_r)\in A\\ i\in\mathbb{D}_d}} \left(\sum_{e\in E} (\delta_{i,r,e})\right)$$

with

$$\delta_{i,r,e} = \begin{cases} 1, & if\ e \underset{r}{\longrightarrow} i\ and\ I_r \cup i\ \neq I_r \\ 0, & if\ e \underset{r}{\longrightarrow} i\ and\ I_r \cup i\ = I_r \\ -1, & else \end{cases}$$

Where $E \in \mathfrak{E}$ is the set of explanations, $A \in \mathfrak{R}$ is the audience observing the explanation and $I_r \in \mathfrak{D}$ is the knowledge base of an reasoner which is used to reason upon the *explanandum* for a SoI in the given domain. We acknowledge that this is a practical measure, since the degree of explanations drops when an explanation is repeated in front of the same audience several times. Further we chose the supremum instead of an average since for a scientific "proof of concept" we need one reasoner able to reason upon the explanation. With this definition of a measure for the degree of explanation, we can conclude that a theoretical complete self-explaining explanation with $\mu(E) = |\mathbb{D}_d|$ for some explanations $E \in \mathfrak{E}$ could exist, so that no other explanation $e_i \in E$ could explain the information i better to the audience – practically it may be not possible to produce such an explanation. For example, a good practical explanation is given by the Peano-Axiomes a set of axioms that define the natural numbers [21]. Indeed, this is only a good explanation for reasoners able to interpret the formalism. For them, there might be no explanation easier to understood, as the Peano-Axiomes present the most accurate way to describe natural numbers. In contrast, if the reasoner are not able to understand the formalism, they a not able to infer what natural numbers are given the Peano-Axiomes.

However, for a specific domain $\mathbb{D}_d$ an explanation *e* might be self-explanatory if the knowledge base $I_{r_1} \ldots I_{r_n} \in \mathbb{D}_d$ of the audience $r_1 \ldots r_{n}, n \in \mathbb{N}$ of a domain $\mathbb{D}_d$, is filled in that way, that the audience might reason to extract the entire information *i* hold by *e* by observing *e*.

On the one hand, the degree of self-explanation can be interpreted as the additional information needed to create understanding. On the other hand, a measure depends on the reasoning capability of the audience and how the explanation fits to those capabilities. If no further information/capability is required for some reasoner to understand the SOI, then the degree of self-explanation rises. The more information is needed the less the degree of self-explanation becomes, where in the worst case no useful information about the SOI can be extracted from the explanation.

In a domain, the information about the domain might be limited, and with that, the possibility for a good explanation might be given.

To come back to our chess example the move: "Qxd4" probably needs further explanation. First we could explain the steno-notation syntax: The first element represent a chess piece here *Q* for the queen. The second element represents an optional action, here *x* which stands for making a capture and the last element *d4* concerning the location on the chess board where the move ends. Further we could additionally explain the meaning of "queen" or "making a capture". If e.g. the audience has watched the move of the chess game, the first explanation of the move given above is $\mu(E) = 0$. Under the assumption that the audience of this explanation does not know the steno-notation, the first explanation of the move given above could is $\mu(E) = -1$. That is, because there is one explanation and it is not understood. Now the second more detailed explanation can be of higher or lower quality. Since we have added multiple sub-explanations (Q, x, d4, queen and capture), if the audience still does not understand the explanation the measure of explanation can become $\mu(E) = -6$. In this case all explanations did fail to transport information to the audience. Further this explanation does not contain information about where the move started from, thus not being completely self-explanatory, since this depends on the context of the chess game. As we argued above, such contextual information is needed in self-explanatory descriptions. As contexts can be hierarchical we see a domain as a set of contexts.

2.3.2 Practical Measures

Since we define a practical but abstract measure of explanation, the next section will survey different measures of explanations. To start with we have different types of measures: confirmation measures and coherence measures [22]. Here an explanation *E* is incrementally confirmed by evidence *d* given some background knowledge *I_r*. The observations collected are formalized in the data *d*. I_r is defined as the information that the reasoner *r* holds as background knowledge.

D.H. Glass [22] argues that the most probable explanation is not equal the "best explanation". Thus the authors work compares different measures for the quality of explanation. The measures compared are separated into three classes. The first class is based on the theorem of Bayes:

1. As baseline the maximum Bayesian a-posteriori probability (MPE) of an explanation is chosen.

$$P(E_i|d) = \frac{P(d|E_i) \times P(E_i)}{P(d)}$$

With that an explanation E_1 is better than explanation E_2 iff:

$$P(E_1|d) > P(E_2|d)$$

Here the most probable explanation regarding the observations d is chosen. Furthermore, the author argues that defining the best explanation as the most probable one makes the inference to the best explanation trivial. We argue that the evidence and the prior used in Bayes differ according to the view point or context of a reasoner – which makes the interpretation of the probability a subjective one. Other Bayesian measures are:

2. The maximum likelihood approach [23] which compares the likelihood of an event:

$$P(d|E_1) > P(d|E_2)$$

3. The Conservative Bayesian [24] combines the maximum likelihood approach with the addition that the a-priori probability of E_i is regarded as well. Thus E_1 is a better explanation then E_2 iff:

$$P(d|E_1) > P(d|E_2) \text{ and } P(E_1) > P(E_2)$$

The conservative Bayesian has the problem that in the case where the likelihood of one explanation and the a-prior of the other one is greater it fails to order the two explanations.

The second class is based on confirmation theory. Here the confirmation or disconfirmation of an explanation E by an observation d is given only if there is a positive or negative dependence between the explanation E and the observation d. Thus confirmation measures rate the increase of probabilities of an explanation E with the observation d [22].

4. The difference confirmation measure compares the difference of confirmation after an observation d:

$$P(E_1|d) - P(E_1) > P(E_2|d) - P(E_2)$$

5. The likelihood ratio measures the ratio of the confirmation of the two explanations:

$$log\left[\frac{P(d|E_1)}{P(d|\neg E_i)}\right] > log\left[\frac{P(d|E_2)}{P(d|\neg E_i)}\right]$$

where

$$P(d|\neg E_i) = \sum_{e \in \mathbb{E}\setminus E_i} P(d|e)$$

The third class of measures is based on coherence. There is a discussion if an increase in coherence is accompanied with an increase in the likelihood of truth [25]. E.J. Olsson [26] defines the coherence as:

$$Co(E_i, d) = \frac{P(E_i \wedge d)}{P(E_i \vee d)}$$

6. The overlap coherence measure (OCM) is then defined as:

$$\frac{P(E_1 \wedge d)}{P(E_1 \vee d)} > \frac{P(E_2 \wedge d)}{P(E_2 \vee d)}$$

7. The Fitelson coherence measure [27] (FCM) is then defined as:

$$C_F(E_1, d) > F(E_2, d)$$

with

$$C_F = \frac{F(E_i, d) + F(d, E_i)}{2}$$

and with

$$F(E_i, d) = \frac{P(d|E_i) - P(d|\neg E_i)}{P(d|E_i) + P(d|\neg E_i)}$$

D.H. Glass [22] compares all other measures proportional to the MPE measure. The author experiment concludes that the OCM measure is closest to the most probable explanation.

All of these measures are probabilistic measures but none of them takes the ability of the reasoner into account. In line with D.B. Leake [28] we argue that an explanation can only be evaluated with the perspective of a reasoner, thus taking its knowledge, goal and reasoning capabilities into account. Further we argue in the line of D.H. Glass [22] since our postulation differs from the MPE measure.

■3.0 SELF-EXPLANATORY DESCRIPTIONS

Self-explanation requires formulating an abstract description of the capabilities an agent provides in order to answer the question stated in the Introduction. Indeed, different software engineering paradigms like Service Oriented Architectures (SOA) and of course Agent Oriented Software Development use such descriptions to enable system components to access and use the available capabilities nowadays. Thus descriptions that origin – for example from the SOA community – typically provide additional information to the description to enable the audience to better understand it. This additional information in what we call classical descriptions is constrained to semantic information. The problem at hand is the semi-optimal performance of AI algorithms using currently available descriptions like service matcher and planner [29].

Here we want to motivate this by the results of the Annual International Contest on Semantic Service Selection[2] (S3) [29] at which the SeMa^2 [30] has performed well in the latest (2012) edition. Figure 2 illustrates the precision and recall values of the SeMa^2 and shows the space of improvement.

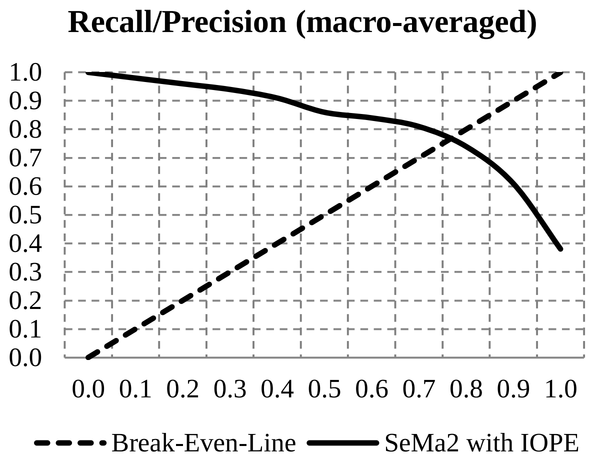


Figure 2 Recall (x-axis) and Precision (y-axis) value of the SeMa^2 for the S3 Contest sample problems

Here, we are able to identify several improvement points. For example the artificial reasoner that reasons upon the given description. Another improvement points are the used knowledge-base, the used languages and the used formalisms. Since this work copes with capability descriptions we try to improve this point. Therefore self-explanatory descriptions – in contrast to classical descriptions – provide additional semantically and contextual information in a structured and computer readable manner as proposed in Oaks *et al.* [4]. This allows us to define self-explanatory descriptions as follows:

> *"A self-explanatory description provides all information necessary for a reasoner in a given context."*

Furthermore, using the measure presented earlier we can state for a given reasoner:

> *"A description is more self-explanatory if it provides additional information about the explanandum B in order to ease the reasoning on the explanans A, such that there exist a why/how-question Q with B as its presupposition, and A explains B."*

In our example of the open door, a more self-explanatory description could be: "The room is unsecure, because the door is open, giving unauthorized personal the possibility to breach the security."

The measure defined above describes an explanation as more self-explanatory if the audience is able extract the explained information. To ease this extraction the explanation is enriched with information, which might help the artificial reasoner. This leads to the question on which information should be provided in the self-explaining description. The following section will try to answer this question by looking at state of the art description and propose an extension to make them more self-explanatory.

3.1 Contained Information

In order to enable a system to be more self-explaining the system has to provide information about its capabilities, interaction ways and current state. Nowadays this information is provided by e.g. service descriptions. This has been subject to standardization, e.g. for service description in the Web Ontology Language (OWL).[3] OWL-S[4] based descriptions present the current state-of-the-art in service description. They enrich the OWL based descriptions with semantically information and contain the following three elements to explain a service:

1. Service profile: Describes the service offered by some provider so that a service requester is able to decide if this is a service fulfilling its request.
2. Service Model: describes how the service is used. This may involve a process, which extends the basic idea of a simple function call.
3. Service Grounding: Specifies the details on how the service can be invoked.

To extend the current available descriptions to self-explanatory descriptions, we opine the idea of C. W. Morris [31] and in consequence distinguish between three different types of information:

1. *Syntax* – concerning the interpretation of signals,
2. *Semantics* – concerning the meaning and relationship between entities and
3. *Pragmatics* – concerning the interpretation of statements

Sooriamurthi and Leake [32] present in an early work their viewpoint on explanations in the Artificial Intelligence (AI) research domain. They developed a definition of explanation "as a situated, utility-based, hierarchical, goal-driven process" [32]. The authors emphasize that the context should be incorporated in the interpretation and creation process of explanations to enable systems to adapt to dynamic situations and therefore introduce the use of *pragmatics* as *context-dependent interpretation of meanings*. This is important since the explaining system might

[2] The Annual International Contest on Semantic Service Selection (S3) – For more information visit: http://www-ags.dfki.uni-sb.de/~klusch/s3/html/2012.html

[3] The Web Ontology Language (OWL) – For more information visit: http://www.w3.org/TR/owl-features/

[4] OWL-S: Semantic Markup for Web Services – For more information visit: http://www.w3.org/Submission/OWL-S/

have to cope with partial observable situations while creating an explanation. In such situations the proposed approach suggests to take former explanations to guide the search for information to create a new explanation.

D.B. Leake [28] underpins this finding while arguing that with changing system goals the interpretation of an explanation should change as well. Furthermore, the author motivates an explanation evaluation process, which concentrates on evaluating explanations of anomaly events. The proposed process that is illustrated in Figure 3 reaches from an observed anomaly to an evaluation decision. We want to emphasize that an explanation in this work is goal driven, too. That is, because the reasoner using an explanation infers on this explanation in a given context to reach a given goal. Thus multiple explanations can be created depending to the goal of the reasoner and in consequence of the explanation. With this evaluation of explanation, the work of D.B. Leake copes with the same problem as the one followed in this work, which is to create self-explaining agents.

The author also emphasizes that this requirement holds in different research fields like Psychology, Philosophy and AI. At the same time, D.B. Leake uses the factors *plausibility*, *relevance* and *usefulness* for explanations concerning anomalies in regard to a given goal. Coming to the conclusion that "(m)any explanations can be generated for any event, and only some of them are plausible" [28]. The three factors are defined as follows:

- Plausibility – the explanation has to provide a plausible reason on why the entity is of interest.
- Relevance – for the explanation to be relevant it has to create knowledge or belief that is conflict free and should provide information about the entity of interest explained.
- Usefulness – the explanation should contribute to the goal and the explanations purpose.

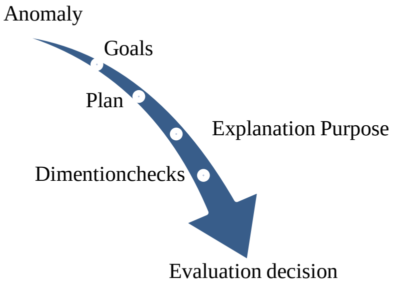


Figure 3 Goal driven explanation evaluation process as proposed by D.B. Leake [28]

Those three properties can be seen as requirements on explanations. The requirements we identify here is that a self-explanatory description must include not only regular information but also semantic information (about the meaning of the regular information) and context information for the context dependent meaning. This correlates with the overall goal of self-explanation proposed by C. Müller-Schloer [14] to enable systems to explain its current state, which seems to be impossible without providing contextual information. Current descriptions do no fulfill all of these requirements – providing only regular information and sometimes semantic information but no context information as e.g. shown in Listing 1.

Coleman *et al.* [33] notice that within communication between agents which are not context aware, the meaning cannot extend beyond the meaning explicitly carried by the message. Thus the meaning needs to be predefined or explained within the message, which forms the crux of the motivation of this work to use context dependent meaning. On the one hand meaning might be twofold: First we can define abstract domain specific meaning. Secondly during runtime the abstract values have to be filled with concrete contextual information. On the other hand, with context aware communicating agents, context dependent meaning might change over time, since the context may change. As argued by R. Laddaga [34] adaptive software requires knowledge of the context as in: "what does the application/capability do?", how to react/adapt to change and runtime support on detecting change as well as modifying parameters to adapt to that change. Thus the contextual information needs to be part of the description. A more detailed overview about the current state-of-the-art in description is given in the following.

3.2 State of the Art Description

As the approaches above are quite fundamental and thus general or theoretic we further want to list more practical approaches in the agent community.

Braubach *et al*. [35] uses the beliefs, desires and intents to formulate goals, knowledge and capabilities for a multi-agent system. Here the beliefs represent the knowledge of an agent. This knowledge base is influenced by belief updates representing the observations an agent does. Desires represent the goals an agent has and intents are the capabilities available to the agent to achieve its goals.

Sycara *et al*. [36] formulates agent and service capabilities utilizing the Input, Output, Precondition and Effect (IOPE) approach. The authors propose the description Language for Advertisement and Request of Knowledge Sharing (LARKS). Additionally a textual description of the meaning of the IOPE description can be attached. This additional description is not optimized for computer readability.

Grüninger *et al*. [37] use First-order Logic Ontology for Web Services (FLOWS) to describe the functionalities of a service. FLOWS can be seen as an extension of a process specification language enabling the description of IOPE as well as internal structures of services, composition patterns, messaging behaviors, and impact of services on the external world.

Martin *et al*. [38] uses the Web Ontology Language to structure the description of services. The proposed Ontology Web Language for Services (OWL-S) is an extension of the Web Services Description Language (WSDL) to enable the description of workflows composed of basic services.

Those approaches all explain something about the subject of interest in specific domains but all lack the ability to measure the amount of information transferred by such an explanation, making it impossible to distinguish the quality of such explanations. Furthermore, the descriptions presented here present context-independent information and with this are not able to fulfill the requirements presented above.

■4.0 APPROACH

So far we have motivated why we need descriptions and why they should be self-explanatory. Now we will introduce a formal definition of self-explanatory descriptions and present the components of such a description in more detail. Furthermore, we will introduce how we want to enrich a description with additional contextual information to describe context dependent meaning. Therefore we want to use a concept from linguistic and investigate its applicability to modern computer readable capability description. We will introduce this theory, which is called Natural Semantic Metalanguage, in the following as the theoretical foundation for our approach. We then explain how meaning is described using this metalanguage and how this meaning is contextualized.

4.1 The Natural Semantic Metalanguage

The Natural Semantic Metalanguage (NSM) is a linguistic theory originated in the early 1970s [39]. It stated that each expression created in a natural language can be represented using a set of atomic terms – so-called universal semantic primes. These primes have an indefinable word-meaning and can be identified in all natural languages [5]. In conjunction with associated grammatical properties NSM presents a decomposition system able to describe all expressions build in the appropriated language. Here, an expression is decomposed into less complex expressions, where the process ends if the expression is decomposed to the atomic level of semantic primes, which cannot be analyzed further. One can imagine that the decomposition builds a tree, where all leafs are semantic primes. Consequently for each natural language a metalanguage exist which consist of the semantic primes in the specific syntax and their appropriated grammatical properties. In this work, those semantic primes are predefined by the applications domain. However, about 63 semantic primes exist which can divided into 16 categories whereby Table 1 lists the semantic primes for the English language.

4.2 Formal Model of Explanation

To use NSM as a metalanguage for self-explanatory descriptions we first need to transform the words used as semantic primes in NSM into concepts that can be used to explicitly describe the meaning of concepts used in descriptions. Therefore, we need a conceptual model as theoretical device for describing the main elements of a self-explanatory description. This model can depart significantly from real world algorithms solving the problem we cope with. However, it will be very useful to explain the basic concepts we want to follow and to clarify the assumptions and the requirements for self-explaining agents. To describe an explanation with context dependent meaning we formalized a self-explanatory description here called explanation, as a five-tuple $e \in E_{\Delta_{SP}}$ with:

$$e = (\Delta_{SP}, \circ^I, \circ^P, \Gamma_d, \Xi^{\,t}_{d})$$

- Δ_{SP} is the set of atomic semantic primes
- $\circ^I$ is the interpretation function of the semantic primes
- $\circ^P$ is the decomposition function of concepts
- Γ_d is the conceptual contest
- $\Xi^{\,t}_{d}$ is the tangible context

4.2.1 Semantic Primes

The first element Δ_{SP} is the set of atomic primes given by the language used for the description. It holds all primes needed for the application. For example, for an explanation e written in the English language the set Δ_{SP} would contain all primes listed in Table 1. In the following we will use these primes for our examples. One might notice that we will not use all primes for the examples through simplicity reasons and that in consequence the set Δ_{SP} used within this work would not contain all primes listed in Table 1.

Table 1 A list of all sematic primes of the English language according to the Natural Semantic Metalanguage theory [5]

Category	Primes
Substantive	I, YOU, SOMEONE, SOMETHING/THING, PEOPLE, BODY
Relational substantives	KIND, PART
Determiners	THIS, THE SAME, OTHER/ELSE
Quantifiers	ONE, TWO, MUCH/MANY, SOME, ALL
Evaluators	GOOD, BAD
Descriptors	BIG, SMALL
Mental predicates	THINK, KNOW, WANT, FEEL, SEE, HEAR
Speech	SAY, WORDS, TRUE
Actions, events, movement, contact	DO, HAPPEN, MOVE, TOUCH
Location, existence, possession, specification	BE (SOMEWHERE), THERE IS, HAVE, BE (SOMEONE/SOMETHING)
Life and death	LIVE, DIE
Time	WHEN/TIME, NOW, BEFORE, AFTER, A LONG TIME, A SHORT TIME, FOR SOME TIME, MOMENT
Space	WHERE/PLACE, HERE, ABOVE, BELOW, FAR, NEAR, SIDE, IN- SIDE
Logical concepts	NOT, MAYBE, CAN, BECAUSE, IF
Intensifier, augmenter	VERY, MORE
Similarity	LIKE

4.2.2 Semantic Primes and their Interpretation

The second element of our theoretical model is $\circ^I: \Delta_{SP} \rightarrow \mathbb{D}_d$ the interpretation function of the semantic primes. For the formalism the set of semantic primes is held in Δ_{SP} and the interpretations are subsumed in the interpretation function $\circ^I$.

In contrast to the original idea of NSM we need to provide an interpretation or more specifically the domain-specific meaning for each semantic prime. As the interpretation for each semantic prime is an inherent part in natural languages this interpretation must be provided for computers and for the domain addressed by the application. Consequently the definition of such an interpretation requires domain knowledge.

An example could be the concept 'I' in an IP-network security domain. We could specify that if some entity has to identify itself it must return its IP-address. As mentioned above the domain-specific meaning of 'I' has to be filled by the developer for further use in descriptions. An example interpretation of 'I' and therefore of the IP could be that the structure of IPs encodes locations in subnets. Thus giving meaning to the otherwise random numbers.

4.2.3 Decomposition

As mentioned above NSM presents a decomposition system able to describe all expressions build in a natural language using the semantic primes.[5] In order to enable agents to decompose

[5] Expression is a term used in the domain of linguistic, where concept is a term used in the domain of description. We will further use the term concept.

concepts to the atomic level of semantic primes we introduce the third element of our model $\circ^P : \Delta_{SP} \times \mathbb{D}_d \rightarrow \wp(\Delta_{SP})$ the decomposition function of concepts.

After having filled all used semantic primes with meaning using the interpretation function, we are able to describe the meaning of all other concepts used in descriptions utilizing the decomposition function.

An example here could be the composed statement 'I CAN DO SOMETHING', where 'SOMETHING' is replaced by e.g. an action that is exposed to others – meaning in our example that an entity 'I' with a specific IP can do a specific action.

Further, the descriptions build upon Δ_{SP} are formalized in e consisting of $n \in \mathbb{N}$ semantic primes used to explain a subject of interest. This is similar to the conventions proposed by M.P. Singh [40] as communicative acts, but lacks the attempt to be used domain-independently.

The next example shows how semantic information can be extended by contextual information. This example is located in a bottle or security domain and uses the concept 'OPEN' which might be of interests concerning doors or bottles. The example ontology "messemodul.owl" of the annual S3 contest [29] defines the meaning of 'OPEN' as follows:

```
<owl:Class rdf:about="#Open">
 <rdfs:subClassOf rdf:resource="&owl;Thing"/>
 <owl:disjointWith rdf:resource="#Closed"/>
</owl:Class>
```

Listing 1 XML representation of the meaning of the concept 'OPEN' taken from the example ontology "messemodul.owl" of the annual S3 contest [29]

As one can imagine the meaning extracted here is quite general. From the metalanguage three concepts are used: 'Class', 'subClassOf' and 'disjointWith', disregarding the domain-specific effect of this concept. We argue that adding such domain-specific and context dependent meaning increases the reasoning capabilities. For example adding information like: 'If something is open for a long time, something bad can happen' might be considered in the security domain. The classification in good or bad is domain-specific. The 'something' is context dependent, which should be replaced at runtime, to the actual thing being opened. This makes it necessary to identify the 'variables' which can be specialized in the context (at runtime). Further the context dependent information that if something is opened, implying that something else is opened as well is normally formulated in the effect of a capability descriptions (e.g. opening a door, opens the room which the door is part of), or described as a property of the concept. The metalanguage of OWL describes this as a 'owl:TransitiveProperty'. We argue that such a property is used by reasoners to reason upon the state 'OPEN' but the change of meaning of other concepts is neglected. Thus the change of meaning of other concepts can be specified giving the concepts context dependent meaning. In our example the meaning of 'INSIDE' changes if something is 'OPEN' connecting all entities which are affected by the something being open to incorporate all transitive things to be inside. The NSM description of this could start with:

> *'If something is open the things inside the something being open, maybe inside all other things being opened'.*

This can be imagined like a door opening and with that joining two rooms to one. At this point we want to highlight that this is not a natural language, since all the concepts used in such an NSM description are part of the metalanguage used to describe meanings. It seems like natural language since a natural like syntax has been chosen for these examples.

The following extensions of the implication in semantics to the implicature in pragmatics, holds a main benefit of using context dependent meaning, but is out of scope for this work.

For the syntax of explanation: To reason upon those explanations, the meaning of every concept and the use of those concepts in explanations, a syntax need to be determined. Swartout *et al*. [41] uses a case-grammar introduced by C.J. Fillmore [42] to structure the descriptions in natural human readable manner. A fitting domain specific syntax is subject to research. As for now we will use the Manchester Syntax [43] meanwhile to ensure the compatibility with OWL2 and with that with existing reasoner.

One might argue that this is equal to a standardization process. We argue that reducing the amount of concepts which are subject of standardization to about 63 concepts might help reducing the effort on creating domain-specific ontologies and ease the reasoning on the created domain-specific language. As further argument, the explanation consists solely of those semantic primes, rather than the descriptions which require Natural Language Processing to be reasoned upon.

As in organic computing an observer/controller pattern is often used [13], we assume that some kind of precondition and goals are defined, as well as some mechanism observing if preconditions are reached. In our surveillance example such a precondition might be that there should never be unauthorized personal in room A. Opening a door to room A might be used as an example on how changes at runtime might change the meaning of the concept 'INSIDE'. It is our thesis that giving an explanation using concepts from NSM might enable reasoners to better identify change in meaning of concepts used. We postulate that with some description like the one outlined above a reasoner might be able to reason that if opening one door to room A, all entities in the 'new' joint room need to be authorized. This kind of inference is called implicature.

4.2.4 Contextual Information

The fourth and fifth element of our theoretical model are the conceptual context Γ_d and the tangible context Ξ_d^t.

As argued above, those semantic models as part of a self-explanatory description can be extended with the changes in context. The change of meaning is caused by the different context decompositions attached from the different reasoner. This context dependent meaning can be seen like the pragmatic extension of semantics in linguistics.

This section motivates why domain specific meaning should be explicitly described. We use as example the concept 'INSIDE': A factoring plant filling bottles and a surveillance system might both have a sensor announcing a statement using the concept 'INSIDE'. Since those are two different domains, the developers of a system need to specify the meaning of 'INSIDE' as part of the semantic primes (here inside a bottle or a room) in different ways. As for now, this semantic is left to the developers since they use their common sense to install a movement sensor to determine if someone is inside a room not in a bottle and reflect this in a ontology used later by some component to reason upon. So there is no explanation needed if the interpretation of the sensor values are fixed. But in a dynamic and adaptive scenario this is not the case and descriptions become necessary to determine if the system component might be of use to reach a certain goal.

Coleman *et al*. [33] identifies three types of context (linguistic, individual, social) that can be interpreted in two forms: Internally and externally. Internally from the view of the entity explaining and externally for the view of the audience. Since in classical descriptions none of this contextual information is explicitly given, a reasoner needs to infer the contextual information, which can be necessary to define the meaning of a concept or to identify an SOI uniquely. As an extension of the approach introduced by Coleman *et al*. to the run-time, we identified two additional granularities of contextual information needed in a self-explanatory description:

- Conceptual context Γ_d: Describes the features of a system under study, which might be of interest to the adaption and are effected by some action. The conceptual context thereby is formalized as Γ_d the contextual information in the domain *d*.

- Tangible context Ξ_d^t: during execution the contextual information gets filled with particular values and linguistic references are resolved, which are used by the adaption. The tangible contest is formalized as Ξ_d^t being dependent of the time t at which a value of a fluent (or variable) $\xi_d^t \in \Xi$ is held for domain *d*.

The conceptual context (the domain: bottles or rooms) and the tangible context are modeled separately by our approach. In our surveillance example the tangible context of 'room A' might be filled with an instance of a room for example 'the hallway' described by its concrete coordinates. Further for example the alarm might have different signals, which are by default, free of meaning. This meaning can be set statically, by programming it into the system, or as proposed by this work, specified in the description. The change of having a semantic meaning of the concept, to having a context dependent meaning is the extension of a syntactical, semantic interpretation to a pragmatic one [31]. Further the identification of fluents (variables effected by some action) and the effect of action on them need to be explained so that reasoner can relate given actions to their goals. As descriptions of actions might include the description of its effect, the tangible contextual information might help to identify fluents.

4.3 Technical Implementation of Context and Domain

Technically the concepts formulated in NSM are then used in OWL2 expressions. Using the direct semantics of OWL2 the expressiveness of the description logic $\mathcal{SROIQ}(\mathcal{D})$ is available. Reducing the expressiveness ensures the decidability [44] and with that the use of existing OWL reasoner likes Pallet[6] or FACT++[7]. As an expression type, a NSM expression can be used like any other expression in OWL2. Introducing to OWL2 a language with the focus on describing context dependent meaning of new concepts. The representation of such an explanation might look like the XML in Listing 2.

■5.0 CONCLUSIONS

In summary an explanation is a kind of description that gives information about a subject of interest. We can conclude that an explanation *e* transports information *i* to an audience of reasoner. Since the audience reasons context dependently about the observed explanation, the explanation has to be extended with context dependent meaning. The quality of an explanation can be measured by how much information the audience can extract from the explanation. So far, we define that an explanation becomes of higher quality if the degree of explanation rises.

```
<nsm:Class rdf:ID="Expression">
 <rdfs:subClassOf>
  <nsm:Domain>
   <nsm:onProperty rdf:resource="#NSM"/>
   <nsm:uri rdf:resource="#URI">
    domain expressions
   </nsm:uri>
 </nsm:Domain>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <nsm:ConceptualContext>
   <nsm:onProperty rdf:resource="#
       expressionBody"/>
   conceptual context
  </nsm:ConceptualContext>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <nsm:TangibleContext>
   <nsm:onProperty rdf:resource="#
       expressionBody"/>
   tangible context
  </nsm:TangibleContext>
 </rdfs:subClassOf>
</nsm:Class>
```

Listing 2 XML representation of a NSM expression

The presented approach uses context-dependent meaning in addition to semantics to describe the concepts used in an explanation. The descriptions are built upon semantic primes – atomic elements of a natural language with an indefinable word-meaning – that are identified by NSM, a natural semantic meta-language. We formalized our approach for self-explanatory description as a five tuple that can be used to build such description using contemporary techniques like OWL2. Figure 4 describes our approach as an abstract process enriching currently available descriptions with more information. In a nutshell, service descriptions, e.g. used in the SOA community, already provide semantic information (as illustrated in Figure 4 at Level 1). This semantic information can be formalized using e.g. Ontologies and is used to describe context independent meaning. This is represented by the tree. In our approach we plan to enhance the concepts described in such Ontology by enriching the already available information with a domain specific description (as illustrated in Figure 4 at Level 2). Those

[6] The Pellet OWL 2 Reasoner for Java – For more information visit: http://clarkparsia.com/pellet/

[7] The FaCT++ OWL-DL reasoner – For more information visit: http://owl.man.ac.uk/factplusplus/

descriptions are built upon the semantic primes provided in NSM. This line of action offers the advantage NSM provides for computer readable descriptions. That is, that every explained concept can be decomposed into the atomic level of semantic primes leading to a decomposition tree where all leafs are semantic primes (as illustrated in Figure 4 at Level 3). That is not only a structure which is efficient for computers to cope with but it might also lead to improvements for the match-making of concepts as only 63 concepts must be matched not all concepts buildable in a language (e.g. OWL-S based descriptions can be used to describe an infinite number of concepts).

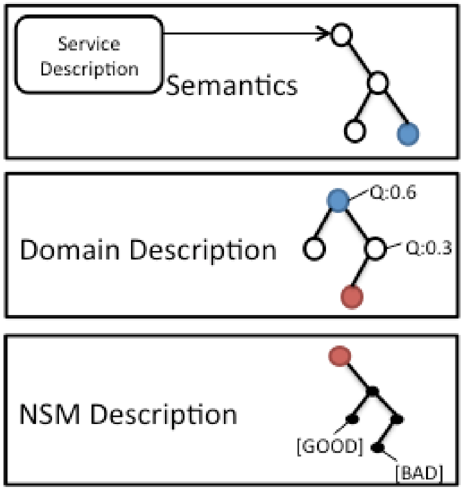


Figure 4 Our approach for self-explanatory descriptions enhances currently available descriptions (level 1) with domain information (level 2) using NSM in order to reduce the complexity of concepts described (level 3)

A concrete example on how well we as humans do understand such explanations is given by this (hopefully) self-explaining description:

> *'This is something very big / people can see it / people can think like this about this something: "it is a place / is is above all other places / it is far from people"'*

Depending to the background knowledge the interesting reader might reason that we search for a concept like *the sky* or *the universe*. An artificial reasoner will rather need a more extensive explanation since most of our background knowledge is missing. However, the reasoning process is the same and maybe powerful enough to match such an explanation with a prior known concept.

To prove this thesis we have to create self-explanatory descriptions on which artificial reasoners can make inference. Yet the creation of such descriptions and the development of tools is subject to research. Based on the presented formalism we can state that one needs to fulfill the following three steps to implement a self-explanatory description:

- Define meaning of the semantic primes
- Describe meaning of concepts used with only those semantic primes, including context dependent (conceptual and tangible) and context independent meaning
- Use described concepts to build up an description

For the purpose of evaluation a sufficient amount of services or agent capabilities need to be described using the approach presented in this work. Then the reasoning upon those descriptions can be subject of research.

Our future work will be concerned with properties of explanation, in the attempt to make the definition more tangible. Further we want to integrate the existing structures of explanations like BDI and IOPE into explanations, to become able to connect an explanation to the goal of an agent, and with that fill the conceptual context with concrete instances.

The decomposition process needs to be automated. Thus a formal syntax has to be chosen and data sources like DBPedia and Wordnet can to be used to automate the decomposition process. This leads to a learning component, which adapts the descriptions at runtime and extracts heuristics out of the explanation.

References

[1] Kephart, J., and D. Chess. 2003. The Vision of Autonomic Computing. *IEEE Computer*. 36(1): 41–50.

[2] Hinchey, M.G., and R. Sterrit. 2006. Self-Managing Software. *IEEE Computer*. 39(2): 107–109.

[3] Fähndrich, J., S. Ahrndt, and S. Albayrak. 2013. Towards Self-Explaining Agents. Trends in Practical Applications of Agents and Multiagent Systems. *Advances in Intelligent Systems and Computing*. 221: 147–154.

[4] Oaks, P., A. Hofstede, and D. Edmond. 2003. Capabilities: Describing What Services Can Do. Service-Oriented Computing - ICSOC 2910(1): 1–16.

[5] Goddard, C. 2008. *Cross-Linguistic Semantics*. Amsterdam: John Benjamins Publishing Company.

[6] Chi, M. 2000. Self-explaining Expository Texts: The Dual Processes of Generating Inferences and Repairing Mental Models. *Advances in Instructional Psychology*. 5(1): 161–238.

[7] Overton, J.A. 2011. Scientific explanation and computation, WS on Computational Explanations. International Joint Conference on Artificial Intelligence.

[8] Moore, J., and W. Swartout. 1989. A Reactive Approach to Explanation. Proceedings of the Eleventh International Joint Conference on Artificial Intelligence. 2(1): 1504–1510.

[9] Salehie, M., and L. Tahvildari. 2009. Self-Adaptive Software: Landscape and Research Challenges. *ACM Transactions on Autonomous and Adaptive Systems*. 4(2): 1–42.

[10] Heckerman, D., E. Horvitz, and B. Nathwani. 1992. Toward Normative Expert Systems: Part I The Pathfinder Project. *Methods of Information in Medicine*. 31(2): 90–105.

[11] Cole, W. 1989. Understanding Bayesian Reasoning via Graphical Displays. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 20(SI): 381–386.

[12] Druzdzel, M. 1996. Qualitative Verbal Explanations in Bayesian Belief Networks. *Artificial Intelligence and Simulation of Behavior Quarterly* – Special Issue on Bayesian belief networks. 94: 43–54.

[13] Müller-Schloer, C., and H. Schmeck. 2010. Organic Computing: A Grand Challenge for Mastering Complex Systems. *It – Information Technology*. 52(3): 135–141.

[14] Müller-Schloer, C. 2004. Organic Computing–On the Feasibility of Controlled Emergence. IEEE/ACM/IFIP International Conference on Hardware/Software CoDesign and System Synthesis. 2–5.

[15] Cheng, B., R. Lemos, *et al*. 2009. Software Engineering for Self-Adaptive Systems: A Research Roadmap. Software Engineering for Self-Adaptive Systems. 1–26.

[16] Duda, R., D. Stork and P. Hart. 2000. Pattern Classification and Scene Analysis Part 1 Pattern Classification. Wiley-Interscience. Edition 2.

[17] Leake, D. 1991. Goal-based Explanation Evaluation. *Cognitive Science*. 15(4): 509–545.

[18] Ghallab, M., D. Nau, and P. Traverso. 2004. Automated Planning: Theory & Practice. *Morgan Kaufmann Series in Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers.

[19] Sutton, R., and A. Barto. 1996. Reinforcement Learning–An Introduction. *Adaptive Computation and Machine Learning Series*. Cambridge: The MIT Press.

[20] Cambridge Dictionary Online. 2012. URL http://dictionary.cambridge.org/dictionary/british/self- explanatory?q=self-explanatory.

[21] Peano, G. 1889. Arithmetices principia, nova methodo exposita (The Principles of Arithmetic, presented by a new method).

[22] Glass, D. 2009. Inference to the Best Explanation: A Comparison of Approaches. In: Proceedings of the 2nd Symposium on Computing and Philosophy. 22–27.

[23] Glass, D., 2007. Coherence Measures and Inference to the Best Explanation. In: *Synthese*. 157(3): 275–296.

[24] Chajewska, U., and J. Halpern. 1997. Defining Explanation in Probabilistic Systems, Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence. 62–71.

[25] Olsson, E. 2005. Against Coherence: Truth, Probability, and Justification. Oxford: Oxford University Press.

[26] Olsson, E. 2002. What is the Problem of Coherence and Truth? *Journal of Philosophy*. 99(5): 246–272.

[27] Fitelson, B. 2003. A probabilistic theory of coherence. *Analysis*. 63(3): 194–199.

[28] Leake, D.B. 1992. *Evaluating Explanations: A Content Theory*. New-York: Psychology Press.

[29] Klusch, M., U. Küster, A. Leger, D. Martin, and M. Paolucci. 2012. 4th International Semantic Service Selection Contest - Performance Evaluation of Semantic Service Matchmakers. http://www-ags.dfki.uni-sb.de/~klusch/s3/s3c-2012-summary-report.pdf. Last visited: 2013-06-30.

[30] Masuch, N., B. Hirsch, M. Burkhardt, A. Heßler, and S. Albayrak. 2012. SeMa2: A Hybrid Semantic Service Matching Approach. *Semantic Web Services*. Berlin: Springer. 35–47.

[31] Morris, C.W. 1938. *Foundations of the Theory of Signs*. Chicago: University of Chicago Press. 1(2): 66.

[32] Sooriamurthi, R., and D. Leake. 1994. Towards Situated Explanation. Proceedings of the Twelfth National Conference on Artifical Intelligence. 2(1): 1492.

[33] Coleman, J., M. O'Rourke, and D.B. Edwards. 2005. Pragmatics and Agent Communication Languages. 50–62.

[34] Laddaga, R. 2000. Active Software. First International Workshop on Self-Adaptive Software. New-York: Springer. 11–26.

[35] Braubach, L., A. Pokahr, and D. Moldt. 2005. Goal Representation for BDI Agent Systems. Programming Multi-Agent Systems. *Lecture Notes in Computer Science*. 3346(1): 44–65.

[36] Sycara, K., M. Klusch, S. Widoff, and J. Lu. 1999. Dynamic Service Matchmaking among Agents in Open Information Environments. SIGMOD Record. 28(1): 47–53.

[37] Grüninger, M. Hull, and S. McIlraith. 2008. A Short Overview of Flows: A First-order Logic Ontology for Web Services. *Data Engineering*. 31(3):1–5.

[38] Martin, D., M. Paolucci, S. McIlraith, and M. Burstein. 2004. Bringing Semantics to Web Services: The OWL-S Approach. First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004). 26–42.

[39] Wierzbicka, A. 1996. *Semantics: Primes and Universals*. Oxford: Oxford University Press.

[40] Singh, M. 1998. Agent Communication Languages: Rethinking the Principles. *IEEE Computer*. 31(12): 40–47.

[41] Swartout, G, G. Yolanda, and A. Valente. 1999. Representing capabilities of problem solving methods. IJCAI Workshop on Ontologies and Problem-Solving Methods. 1–8.

[42] Fillmore, C. 1968. The Case for Case. New York: Holt, Rinehart & Winston.

[43] Horridge, M., N. Drummond, J. Goodwin, A. Rector, R. Stevens, and H. Wang. 2006. The Manchester OWL Syntax. Proceedings of the 2006 OWL Experiences and Directions Workshop OWL-ED. 216(1).

[44] Horrocks, I, O. Kutz, and U. Sattler. 2006. The even more irresistible SROIQ, Proceedings of the 10th International Confonference on Principles of Knowledge Representation and Reasoning (KR 2006), AAAI Press. 57–67.