Four Enterprise Modeling Perspectives and Impact on Enterprise Information Systems

Boris Shishkov[1,2,3], Aglika Bogomilova[3], and Magdalena Garvanova[1]

[1] Faculty of Information Sciences
University of Library Studies and Information Technologies, Sofia, Bulgaria
[2] Institute of Mathematics and Informatics
Bulgarian Academy of Sciences, Sofia, Bulgaria
[3] Institute IICREST, Sofia, Bulgaria
b.b.shishkov@iicrest.org, abogomilova@gmail.com,
m.garvanova@unibit.bg

Abstract. The alignment between Enterprise Modeling (EM) and Software Specification (SS) is still uncertain, this leading to enterprise information systems of low quality. Hence, only the EM-driven software generation could help aligning software functionalities to domain requirements. This inspires the emergence of innovative approaches, such as the SDBC (Software Derived from Business Components) approach, considered by us. It steps on a conceptual invariance (embracing concepts whose essence goes beyond the barriers between social and technical disciplines), while SDBC also builds upon this, to accommodate a modeling duality featuring (1) technology-independent EM rooted in social theories; (2) SS rooted in computing paradigms. The proposed EM-SS alignment is component-based, featuring a potential re-use of modeling constructs, such that the modeling effectiveness and efficiency are stimulated. We consider particularly *(1)*, observing insufficient EM maturity in general: many analysts conduct intuitive EM (not scientifically grounded); they often fail to be exhaustive (some mainly focus on behavior, others – on data, and so on); some analysts mix up essential business things with information exchange that is not featuring essential business things; other analysts are unaware of the importance of communicative acts; many analysts overlook regulations and values; and so on. We address 4 EM perspectives, namely language acts, regulations, public values, and energy – each of them is a theory/paradigm on its own and studying them in isolation is important. It is also important considering them in combination, identifying possibilities for bringing them together, in order to achieve a more exhaustive EM foundation with regard to corresponding SS. We argue that the 4 perspectives make our EM vision usefully broad but we do not claim exhaustiveness. We have studied each of them, providing accordingly theoretical justification and partially demonstrating their practical applicability (by means of an example). Thus, the contribution of our paper is two-fold: (i) We make a small contribution to the development of the SDBC approach; (ii) We analyze different EM perspectives.

Keywords: Enterprise modeling, Language acts, Norms, Public values, Energy.

1 Introduction

The alignment between enterprise modeling and software specification is still uncertain, this leading to enterprise information systems of low quality; hence, only the enterprise-modeling-driven software generation could help aligning software functionalities to domain requirements [1]. This has inspired the emergence of innovative approaches, such as the SDBC (Software Derived from Business Components) approach [2], referred to in the current paper.

On the one hand, the approach steps on a conceptual invariance (embracing concepts whose essence goes beyond the barriers between social and technical disciplines), while on the other hand, the approach builds upon that "common ground" to accommodate a modeling duality featuring (1) technology-independent enterprise modeling that is rooted in social theories and (2) software specifications that are rooted in computing paradigms. Further, the proposed alignment between enterprise modeling and software specification is component-based, featuring a potential re-use of modeling constructs, such that the modeling effectiveness and efficiency are stimulated [3].

In the current paper, we consider especially *(1)*, observing insufficient maturity as it concerns enterprise modeling in general: many analysts conduct intuitive enterprise modeling that is not scientifically justified; they often fail to be exhaustive in their modeling (some of them would only focus on modeling behavior and others would only focus on modeling data, and so on); some analysts would mix up essential business things (e.g., John paid for his service subscription) with information exchange that is not featuring essential business things (e.g., John entered his PIN incorrectly while using an ATM); other analysts would be unaware of the importance of communicative acts in real-life communication, through which commitments are generated, that are in turn crucially important with respect to the processes within an organization; many analysts would overlook regulations and public values as key restrictions with regard to the functioning of an organization [1].

In particular, we address **four enterprise modeling perspectives**, namely *language acts*, *regulations*, *public values*, and *energy* – each of them is a theory/paradigm on its own. Hence, studying them in isolation is important. Nevertheless, it is also important considering them in combination, such that possibilities are identified for bringing them together, in order to achieve a more exhaustive enterprise modeling foundation with regard to corresponding software specifications.

Even though we argue that those *four perspectives* make our enterprise modeling vision usefully broad, we do not claim exhaustiveness. What we have done in this paper is to study each of them, providing accordingly theoretical justification and partially demonstrating their practical applicability (by means of an illustrative example). Thus, the contribution of the paper is two-fold: (i) We make a small contribution to the development of the SDBC approach; (ii) We analyze different enterprise modeling perspectives.

Finally, we are to elaborate each of those perspectives:

● As studied in [1], one way of modeling an enterprise is to capture the entity-to-entity communications and related actions, as featured in [4]. For example, at a pizza desk, we observe customers going for pizzas, sandwiches, and so on. Imagine that John

is staying at the desk and Richard is a customer, and Richard would like to have for lunch a piece of Pizza Margherita. By asking for this, Richard is stating a REQUEST and in turn, John could either DECLINE this request (if they have no Pizza Margherita) or PROMISE to deliver a piece of Pizza Margherita to Richard. Then if promised, the pizza is to be delivered. If this is done, the mere act of delivering the piece of Pizza Margherita is a STATEMENT featuring the result of what was done, triggered by the request. Nevertheless, the statement is not "completing" the interaction because Richard may ACCEPT the result (if the piece of pizza was delivered in time and looks OK) or NOT (if the piece of pizza does not look OK and/or was delivered with a huge delay). Hence, all those communicative acts ("request", "promise", "state", and so on) are straightforwardly related to corresponding ACTIONS (for example: "a piece of Pizza Margherita is being delivered by John to Richard", "Richard is paying for the piece of pizza", and so on). Those actions in turn represent "building blocks" as it concerns the business processes "flowing" among enterprise entities. We therefore argue that by capturing communicative acts and corresponding actions, one is capable of identifying and modeling business processes. This in turn allows for delivering enterprise models accordingly.

- As studied in [1], another way of modeling an enterprise is to establish what may, may not or must happen in a particular situation, as featured in [5]. For example, in case George is a VISA credit card holder, then: (i) If George has not reached his credit limit, he MAY pay (up to the limit) using his VISA credit card; (ii) If George has reached his credit card limit, he MAY NOT pay using his VISA credit card; (iii) If VISA has billed George for a minimal monthly payment, George MUST do the payment to VISA. Those are examples of norms (or rules) that in turn allow for bringing forward the REGULATIONS governing an enterprise. We therefore argue that by capturing norms, one is capable of modeling the potential processes as it concerns an enterprise.
- As studied in [1], public values, such as PRIVACY, TRANSPARENCY, and ACCOUNTABILITY are important as it concerns the CONTEXT-AWARENESS of enterprise information systems [6,7]. In the current paper, we would also like to address the potential for usefully considering public values in the enterprise modeling process, inspired by [8,9]. In this, we refer to previous work [10] where we superimpose such public values (labelled "atomic values") to what we call "composite values" that are featured in three bipolar dimensions, namely: hierarchy - egalitarianism, autonomy - conservatism, and harmony – mastery. We argue that by identifying the relevant bipolar dimension and positioning an enterprise accordingly, one is capable of modeling the general structure of business processes. Then this could be usefully related to atomic values that would be adequately weaved in as requirements with regard to the enterprise being modeled.
- And in the end, we argue that considering energy-related issues could also be useful in modeling enterprises. By this we mean intangible issues that are not "visible" but are of importance with regard to what the modeled enterprise actually is. Let us take an example featuring Tom who is a human being. One would agree that what Tom actually is extends far beyond what the eyes can perceive, far beyond the flesh, blood, and bones. There are also other things, both gross and subtle, which are claimed to constitute in combination what the person is. Inspired by Hindu Philosophy [11], we

see all this as related to five essential elements, namely: earth, water, fire, air, and ether. Through specific features related to each of those elements, we are able to describe a human being much better than a purely visual description. We hence argue that this also holds for enterprises: two enterprises (for instance) could be identical in terms of structure, business goals, turnover, and so on. Still, they could differ a lot and those differences are not always trivial to see. Through "five-elements glasses" we would be able to "see more": one of the enterprises could be managed in an "ether" style (space) and this means well-established teams, managers who delegate, equal importance as it concerns everyone in the hierarchy, and so on; the other enterprise could be managed in a "fire" style (for example) and this means much responsibility for the leader with less delegation, and also push for control of the leader as it concerns all hierarchical levels.

We study the appropriateness and strengths of modeling enterprises, driven by each of the above perspectives, and the remaining of the current paper is organized as follows: We consider related work in Section 2 and in particular – we present the SDBC modeling foundations and relevant works touching upon each of the abovementioned enterprise modeling perspectives. Our views concerning those perspectives are presented in Section 3, especially with regard to the SDBC-driven modeling. We provide partial exemplification in Section 4. Finally, we conclude the paper in Section 5.

2 Related Work

As mentioned already, in the current section we consider firstly (in Sect. 2.1) our way of modeling (driven by the SDBC approach) and secondly (in Sect. 2.2) – relevant works with regard to each of the four enterprise modeling perspectives, discussed in the previous section.

2.1 Way of Modeling

Our way of modeling is driven by the **`SDBC approach`** ("**`SDBC`**" stands for "*Software Derived from Business Components*"); *SDBC* is a software specification approach (consistent with MDA [12]) that covers the *early phases* of the software development life cycle and is particularly focused on the `derivation of software specification models on the basis of corresponding (re-usable) enterprise models` [1,2]. *SDBC* is based on three key ideas: (i) The software system under development is considered in its enterprise context, which not only means that the software specification models are to stem from corresponding enterprise models but means also that a deep understanding is needed on real-life (enterprise-level) processes, corresponding roles, behavior patterns, and so on. (ii) By bringing together two disciplines, namely `enterprise engineering` [4] and `software engineering` [13], *SDBC* pushes for applying `social theories` in addressing enterprise-engineering-related tasks and for applying `computing paradigms` in addressing software-engineering-related tasks, and also for integrating the two, by means of sound methodological guidelines. (iii) Acknowledging the essential value of re-use in current software

development, *SDBC* pushes for the identification of re-usable (generic) enterprise engineering building blocks whose models could be reflected accordingly in corresponding software specification models. We refer to [1] for information on *SDBC* and we are reflecting the *SDBC* outline in Figure 1.

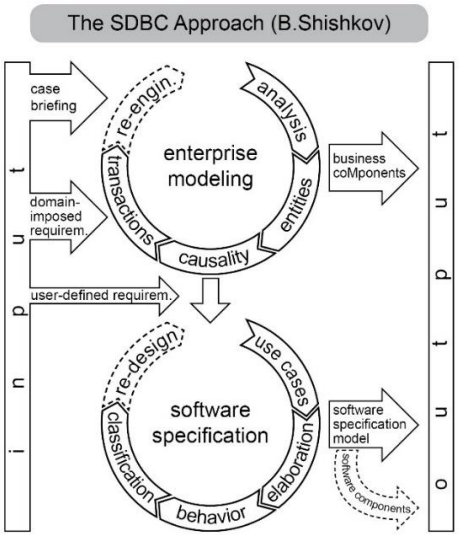

Fig. 1. Outlining the *SDBC* approach (Source: [14], p. 48)

As the figure suggests, there are two *SDBC* modeling milestones, namely `enterprise modeling` (*first milestone*) and `software specification` (*second milestone*). The first milestone has as input a *case briefing* (the initial (textual) information based on which the software development is to start) and the so-called *domain-imposed requirements* (those are the domain regulations to which the software system-to-be should conform).

Based on such an input, an analysis should follow, aiming at structuring the information, identifying missing information, and so on. This is to be followed by the identification (supported by corresponding social theories) of enterprise modeling *entities* and their inter-relations. Then, the *causality* concerning those inter-relations needs to be modeled, such that we know what is required in order for something else to happen [14]. On that basis, the dynamics (the entities' behavior) is to be considered, featured by *transactions* [3]. This all leads to the creation of enterprise models that are elaborated in terms of composition, structure, and dynamics (all this pointing also to corresponding data aspects) – they could either feed further software specifications and/or be "stored" for further use by enterprise engineers. Such enterprise models could possibly be reflected in corresponding *business coMponents* (complete models of business components) [1]. Next to that, re-visiting such models could possibly inspire enterprise re-engineering activities – see Figure 1.

Furthermore, the second milestone uses as input the enterprise modeling output (see above) and the so-called *user-defined requirements* (those requirements reflect the demands of the (future) users of the software system-to-be towards its functioning) [14].

That input feeds the derivation of a *use case model* featuring the software system-to-be. Such a software specification starting point is not only consistent with the Rational Unified Process - RUP [15] and the Unified Modeling Language – UML [16] but is also considered to be broadly accepted beyond RUP-UML [17]. The use cases are then *elaborated* inspired by studies of Cockburn [17] and Shishkov [3,1], such that *software behavior models* and *classification* can be derived accordingly. The output is a *software specification model* adequately elaborated in terms of *statics* and *dynamics*. By applying de-composition, such a model can be reflected in corresponding *software components*, as shown in the figure. Such an output could inspire software engineers to propose in a future moment software re-designs, possibly addressing new requirements.

Further, in bringing together the first milestone of *SDBC* and the second one, we need to be aware of possible *granularity mismatches*. The enterprise modeling is featuring business processes and corresponding *business coMponents* (for the sake of brevity we do not provide further elaboration as it concerns the business coMponent concept; for more information, interested readers are referred to [1]) but this is not necessarily the level of granularity concerning the software components of the system-to-be. With this in mind, an ICT (Information and Communication Technology) APPLICATION is considered as matching the granularity level of a business component – an ICT application is an implemented software product realizing a particular functionality for the benefit of entities that are part of the composition of an enterprise system and/or a (corresponding) enterprise information system [1]. Thus, the label "software specification model" as presented in Figure 1, corresponds to a particular ICT application being specified. Hence, software components are viewed as implemented pieces of software, which represent parts of an ICT application, and which collaborate among each other driven by the goal of realizing the functionality of the application [1] (functionally, a software component is a part of an ICT application, which is self-contained, customizable, and composable, possessing a clearly defined function and interfaces to the other parts of the application, and which can also be deployed independently [13]). Hence, a software coMponent is a conceptual specification model of a software component [1]. Said otherwise, THE SECOND SDBC MILESTONE is about the identification of software coMponents and corresponding software components.

2.2 Language Acts, Regulations, Public Values, and Energy

As already mentioned, in the current Sect. we consider works relevant to the *four enterprise modeling perspectives* discussed in the Introduction.

As it concerns **`language acts`**, we are mainly considering the works of **Dietz** [4] that touch upon *Enterprise Ontology*, aiming at extracting the *essence* of an enterprise from its actual appearance. The *Organization Theorem* has crucial importance with regard to that and it is in turn backed by four axioms, namely: the *operation axiom*, the *transaction axiom*, the *composition axiom*, and the *distinction axiom*. The operation axiom states that the operation of an enterprise is constituted by actors who perform

two kinds of acts, namely *production acts* (that contribute to bringing about the goods/services being delivered) and *coordination acts* (that are about complying with commitments regarding the performance of corresponding production acts). Referring to **LAP** – the *Language-Action Perspective* [18] and to the *operation axiom*, the transaction axiom poses that a coordination act is performed by one actor (called "producer") and directed towards another actor (called "customer"). Hence, the notion of **transaction** refers to the question how production acts and coordination acts are related to each other, and this all points to two "conversations", namely: an *actagenic conversation* (it is about the order) and a *factagenic conversation* (it is about the result). Finally, it is established that the INITIATOR of a *transaction* is the customer while the EXECUTOR of the transaction is the producer. The composition axiom concerns the notion of a *business process* seen as a structure of causally related transactions. Hence, next to the *operation axiom* (that considers the elementary acts) and the *transaction axiom* (that considers putting them together as *transactions*), the *composition axiom* considers structures of transactions driven by causality and this is what is meant by "business process". *Causality* can be illustrated in a simple way: In order to configure a local-area network, one would need to have firstly provided personal computers, a switch, a server, and so on, and in order to configure in turn a personal computer, one would need to have firstly provided a monitor, a hard disk, a motherboard, and so on. In a similar way, *causality* is considered when addressing structures of *transactions*. Finally, the distinction axiom serves to separate the distinct human abilities playing a role with regard to communication, namely: PERFORMA (the actual act of *evoking an attitude*), INFORMA (it is about *conveying semantics*), and FORMA (it is about *conveying information*).

As it concerns **regulations**, we are mainly considering the works of **Liu** [5] that touch upon *Organizational Semiotics* (**OS**), in general, and the *Norm Analysis Method* (**NAM**) – in particular. Actually, *OS* is a branch of **semiotics** while *NAM* is one of the two *OS* methods. *OS* focuses on the nature, characteristics, and behavior of signs – it is claimed that in contrast to the concept of *information*, **signs** offer a more rigorous and solid foundation to understand *information systems*. For example, within a business context, a bank note is much more than a piece of colored paper with digits on it; it stands for the bank note holder's wealth and ability to pay, as well as the issuing bank's authority and credibility, and much more. Next to *signs*, *OS* considers the notion of **affordance** featuring *dependencies* – for example, in the context of a university library, a book *affords* to be borrowed. Finally, it is through *NAM* that *OS* addresses the **norms** based on which behaviors are realized – norms are the rules and patterns of behavior, either *formal* or *informal*, *explicit* or *implicit*, existing within a society, an enterprise, or even a small group of people working together to achieve a common goal; *four types* of norms are considered, namely: evaluative norms, perceptual norms, cognitive norms, and behavioral norms. Each type of *norm* governs human behavior from different aspects. A **norm analysis** is normally carried out on the basis of the results of a prior **semantic analysis** (featuring the concepts under study and their interrelations). The *semantic* model delineates the area of concern of an enterprise. The patterns of behavior specified in the *semantic* model are part of the fundamental *norms* that retain the ontologically determined relationships between agents and actions

without imposing any further constraints. In general, a complete *norm analysis* can be performed in four steps: responsibility analysis (it enables one to identify and assign responsible entities to each action), proto-norm analysis (it helps one to identify relevant types of information for making decisions concerning a certain type of behavior), trigger analysis (it is to consider the actions to be taken in relation to the absolute and relative time), and detailed norm specification (it concerns the actual specification of *norms* in two versions, a *natural language* and a *formal language*).

As it concerns **`public values`**, we are mainly considering the works of **Schwartz** [9,19] touching upon *three universal human needs* – the needs of individuals as biological organisms, the needs of coordination of social interaction, and the needs of preservation and well-being of a social group. Through socialization, those *needs* are reflected in as **`public values`** ("**`values`**", for short), to give them different significance, and to use culturally shared concepts in the communication process [9]. Schwartz observes consensus in behavioral sciences about some of the leading characteristics of values, namely that they are: 1) beliefs closely related to emotions; 2) motivational construct – refer to the preferred goals; 3) have an abstract nature that distinguished them from norms and attitudes; 4) function as standards or criteria in the selection or evaluation of behaviors, people or events; 5) rank in order of importance and build a hierarchical system of *value* priorities; 6) many relevant *values* are involved in the formation of a particular attitude or behavior [19]. The organization of *values* at the individual level is the result of the psychological dynamics of conflict and compatibility that humans experience in the process of pursuing different goals in daily life. On the contrary, the structure of the *value* system at the societal level reflects in particular the various models that communities use to solve problems, arising from the regulation of human activity, such as: *a)* the relationship between the individual and the group – to what extent people are autonomous or included in their groups, described by embeddedness vs. autonomy value orientations (the undifferentiated vs. the differentiated from the group individual); *b)* ensuring responsible social behavior – how to motivate people to coordinate their actions and respect the other's right, described by hierarchy vs. egalitarianism value orientations (inequality vs. equality), and *c)* the role of the individual in the natural and social environment – whether it is more important to adapt to the outside world and accept it as it is or to constantly strive to change and exploit it, described by harmony vs. mastery value orientations (adaptation to the environment vs. control and change). *Embeddedness*, *egalitarianism* and *harmony* are collectively oriented values, while *autonomy*, *hierarchy* and *mastery* are individually oriented ones [9,19]. The ways in which those alternatives are solved is reflected in the social value priorities.

As it concerns "**`energy`**"-related issues, we mainly refer to **Satyasangananda** [11] featuring the **`Hindu philosophy`**. According to it, all matter is composed of a combination of *five tattwas*, i.e. **`elements`**. The *Shiva Swarodaya* (an ancient Sanskrit tantric text) explains that creation takes place due to these five elements and by them it is sustained. Further, *Tantraraja Tantra* stipulates that the five elements permeate the entire body and mind. Everything we do and think is under the influence of these *elements*. The *five elements* are known as **`ether`**, **`air`**, **`fire`**, **`water`** and **`earth`**. Nonetheless, the *five elements* should not be mistaken for physical or chemical elements. They should rather be regarded as a consequence of emanations which are created by

different energies or life-force vibrations. However, according to *Hindu* studies (particularly **`Yoga`** and **`Tantra`**) that have examined the elements in detail, we are composed of those *elements* and are continuously subject to their influences; they are not different *energies* but are different aspects of the same *energy* manifesting itself in infinite various combinations. The *elements* constantly interact and when there is a balance between them, we feel *physically healthy*, *mentally calm* and *aware*. As they are constantly changing, their *environment* and their *balance* are disturbed and one or two *energies* may prevail over the rest. As for the *elements* themselves: The **ether** element concerns issues such as *space of mind* and *inspiration*. This *element* governs mental aspects among which are *balance*, *care*, *support*, and so on. The **air** element is "connected" to parts of the brain that are responsible for *creativity* and *art*. This element is about *motion*, *development*, *rhythm*, and so on. The **fire** element nourishes *growth*, *change*, and *evolution*. This element governs *conscious actions*, *dynamics*, and *will*. The **water** element concerns *emotions* that in turn undergo different "phases" and may change unexpectedly. This element is about *directing the focus inwards*, *possessing the power to overcome obstacles*. Finally, the **earth** element is about *patience*, *stability*, and *sustainability*. This element is an essential *security* issue.

3 Four Enterprise Modeling Perspectives and an SDBC-Driven Software Specification

As mentioned already, in the current section we will present our modeling proposal featuring the consideration (with regard to the enterprise modeling challenge) of language acts, regulations, public values, and energy, as part of our *SDBC-driven software specification* that is essentially based on *underlying enterprise modeling*. This enterprise modeling stays on many "pillars" and modeling *language acts*, *regulations*, *public values*, and *energy* are only some of them. Hence, we do not claim exhaustiveness, "keeping the doors open" for further research, as illustrated in Fig. 2:

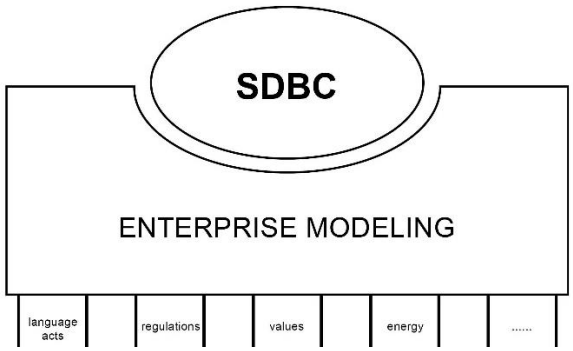

Fig. 2. Addressing four enterprise modeling perspectives

As the figure suggests, we consider each of those perspectives as reflecting corresponding *enterprise modeling foundations*, such that in the end we have a solid enough *enterprise model* that can be reflected in adequate *software specifications*, following the SDBC guidelines. Nevertheless, for the sake of brevity, in the current paper we only limit ourselves to addressing the enterprise modeling perspectives.

3.1 Language Acts

Referring to the discussion presented in the previous section, we go for particularly considering the Transaction Axiom and especially, the Transaction Pattern as a key modeling element to capture language acts in the context of business process modeling, as studied in [1].

We interpret the transaction concept (see *Section 2*) as centered around a particular *production fact*. The reason is that the actual output of any *enterprise system* represents a set of *production facts* related to each other. They actually bring about the useful value of the business operations to the outside world and the issues connected with their creation are to be properly modeled in terms of *structure*, *dynamics*, and *data*.

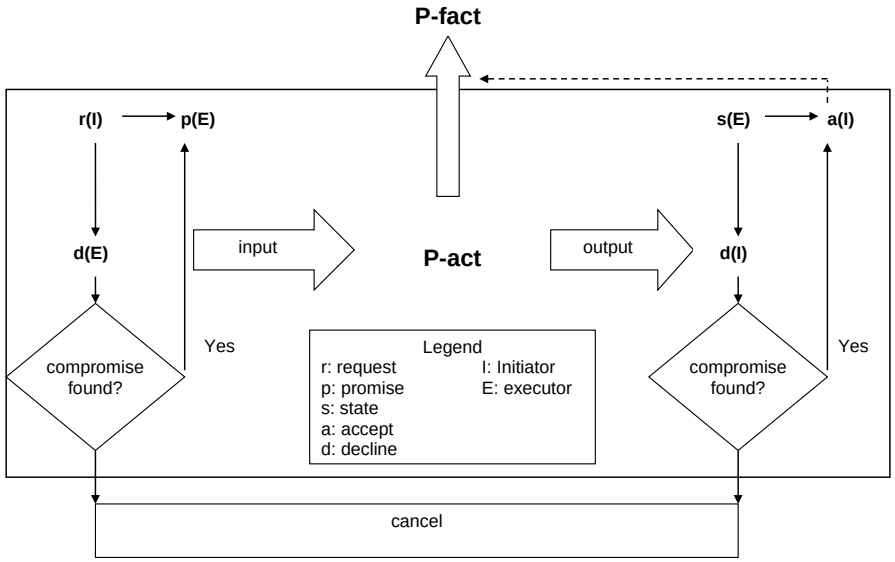

Fig. 3. A proposed interpretation of the transaction concept (Source: [3], p. 70)

However, the already justified necessity of considering also the corresponding *communicative* aspects is important. Although they are indirectly related to the *production facts*, they are to be positioned around them. As already stated, we address this through our interpretation of the *transaction* concept, as depicted in Figure 3; as seen from the figure, the *transaction* concept has been adopted, with a particular stress on the *transaction's* output – the *production fact*. The *order phase* is looked upon as an input for the *production act*, while the *result phase* is considered to be the *production act*'s

output. The dashed line shows that a *transaction* could be successful (which means that a *production fact* has been (successfully) created) only if the *initiator* (the one who is initiating the *transaction*, as presented in Figure 3) has accepted the *production act* of the other party (called *executor*). As for the (*coordination*) *communicative acts*, grasped by the *transaction*, they are also depicted in the figure. The *initiator* expresses a *request* attitude towards a proposal (any *transaction* should concern a *proposition* – for example, a shoe to be repaired by a particular date and at a particular price, and so on). Such a *request* might trigger either *promise* or *decline* - the *executor* might either *promise* to produce the requested product (or service) or express a *decline* attitude towards the *proposition*. This expressed attitude actually triggers a discussion (negotiation), for example: "I cannot repair the shoe today, is tomorrow fine?... and so on". The discussion might lead to a compromise (this means that the *executor* is going to express a *promise* attitude towards an updated version of the *proposition*) or might lead to the *transaction's* cancellation (this means that no *production fact* will be created). If the *executor* has expressed a *promise* attitude regarding a *proposition*, then (s)he must bring about the realization of the *production act*. Then the *result phase* follows, which starts with a *statement* expression from the *executor* about the requested *proposition* that in his/her opinion has been successfully realized. The *initiator* could either accept this (expressing an *accept* attitude) or reject it (expressing a *decline* attitude). Expressing a *decline* attitude leads to a discussion which might lead to a *compromise* (this means that finally the *initiator* is going to express an *accept* towards the realized *production act*, resulting from *negotiations* that have taken place and *compromise* reached) or might lead to the *transaction's cancellation* (this means that no *production fact* will be created). Once the realized *production act* is *accepted* the corresponding *production fact* is considered to have appeared in the (business) reality.

Hence, we adopt language acts in our enterprise modeling, by considering transactions as the elementary building blocks in our modeling.

3.2 Regulations

Referring to the previous discussion (see Section 2), we essentially count on semiotic norms for capturing, expressing, and establishing regulations because regulations represent sets of rules and rules in turn can be adequately brought forward via semiotic norms, as justified by Liu [5].

Further, in *enterprise modeling*, most *rules* and *regulations* fall into the category of **`behavioral norms`**. Those *norms* prescribe *what people must, may, and must not do*, which are equivalent to *three deontic operators*: "*is obliged*", "*is permitted*", and "*is prohibited*". Hence, the following format is considered suitable for specifying *behavioral norms*.

```
whenever <condition>
if <state>
then <agent>
is <deontic operator>
to <action>
```

To exemplify the above, we consider a credit card holder - Josh. Imagine that: (i) The credit card limit is 5000.00 EURO; (ii) Josh has used already 4285.58 EURO; (iii) Josh needs to purchase an airplane ticket for the price of 425.19 EURO. Hence, a norm derived based on the above information is:

```
whenever Josh has a valid credit card
if Josh has not reached the credit card limit
then Josh
is permitted
to use the credit card for an amount up to the difference between the credit card
   limit and the amount currently used
```

In our enterprise modeling, we use norms in elaborating transactions.

3.3 Public Values

Values are considered to be desires of the general public (or public institutions / organizations that claim to represent the general public), that are about properties considered societally valuable, such as respecting the privacy of citizens or prohibiting polluting activities [7]. Even though values are to be broadly accepted (that is why they are public), they may concern individuals (for example: considering privacy) [1]. Hence, put broadly, values concern the societal expectations with regard to the way services should be delivered [20]. Further, we argue that "values" become actual "values" only if resources are committed for this (for example, a government finds privacy so important that time and money are invested to regulate and enforce privacy); otherwise things only remain at the level of "hollow" abstract desires (such as for example: "Make the World a better place") that are stated but are never effectively realized.

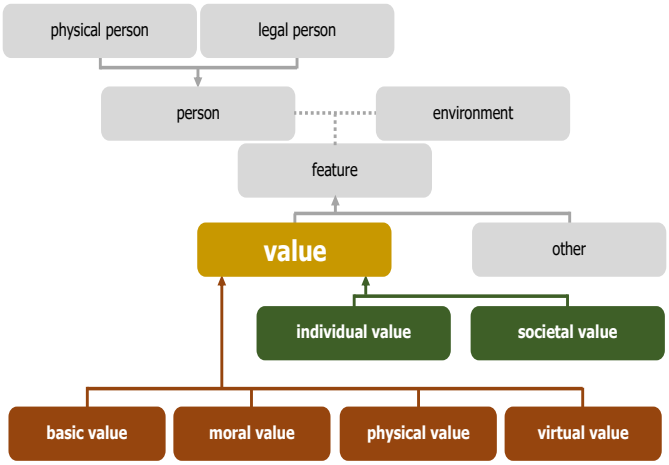

Fig. 4. Categorizing values (Source: [7], p. 404)

We consider a value categorization (Figure 4) according to which values are desires relevant to particular *persons* (either *physical* or *legal* persons) or their *societal environment*. As such, values may either concern a particular individual or society altogether. Hence, we can distinguish between *individual values* (for example, privacy) and *societal values* (for example, sustainability). We also distinguish between *basic values* (for example, love), *moral values* (for example, justice), *physical values* (for example, nature), and *virtual values* (for example, intelligence).

The way to reflect values in enterprise modeling is to "translate" them in functional requirements – even though values themselves are non-functional in nature, the only way to expose them is by means of functional (software) solutions that are to reflect underlying requirements. More information featuring the SDBC-driven requirements specification can be found in [1].

3.4 Energy

Referring to the previous discussion (see Section 2), we explicitly consider the five elements, namely: *earth*, *water*, *fire*, *air*, and *ether*.

We consider each enterprise as an "organism", similarly to a human; we observe similarities between human behavior and organizational (enterprise) behavior, at least as it concerns "influences" from those five elements, and in particular:

- An earth-driven enterprise would be stable and conservative, in the sense that business processes would be evolutionary, changing slowly over time. Examples for this are enterprises driven by narrow expertise and specific business processes, such as hand-made souvenir production, air-co compressor repair, and so on. IT (software) support for such enterprises would most often be aligned with the specific business processes characterizing the enterprise.
- A water-driven enterprise would be solid but unstable and changeable, in the sense that business processes may stay essentially stable but often changeable as realization. Examples for this are enterprises whose business can be realized through different channels, such as consultancy (it can be realized face-to-face, distantly, and so on). IT (software) support for such enterprises would most often be variant-driven, assuming the same software core.
- A fire-driven enterprise would be totally unpredictable, in the sense that business processes can significantly change (often driven by personal decisions). Examples for this are enterprises with strong personal presence, such as art agencies, campaigns, and so on. IT (software) support for such enterprises would most often assume new software instances starting from scratch.
- An air-driven enterprise would be dynamic and fast developing, in the sense that business processes may change even essentially. Examples for this are enterprises whose business processes are subject to technological and/or legal influences, such as road-traffic-related enterprises, e-Businesses, and so on. IT (software) support for such enterprises would most often assume powerful interfaces towards integration with other technologies.

- Finally, an ether-driven enterprise would be balanced and communicative, in the sense that business processes assume integration and coordination with regard to complex environments. Examples for this are tourist agencies, car rental companies, and so on. IT (software) support for such enterprises would most often assume standardization and component-based solutions, allowing for fast "plug-and-play" replacements.

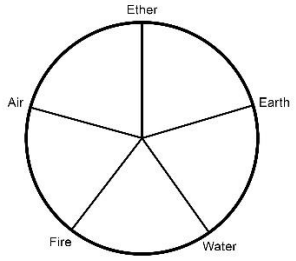

Figure 5. Considering the 5 elements

All five elements are reflected in Figure 5 and the circle suggests "continuity" in the sense that none of those elements pops up in isolation – most often it is the case that more than one of them have impact. Hence, our discussion in the current sub-section is slightly simplified, assuming influence from just one element. We do this for the sake of brevity, driven by the purpose to just outline our ideas on weaving the 5-elements-analysis in enterprise modeling.

What we would do in the end with all this is to facilitate our design as it concerns the software specifications and also the enterprise modeling preceding them. The assumption we make is that we are able to "sense" which element is predominantly influencing the enterprise under study. Discussing this further is left beyond the scope of the current paper.

4 Illustrative Example

As already mentioned, we do partial exemplification in order to illustrate our enterprise modeling touching upon: (i) language acts; (ii) regulations; (iii) public values; (iv) energy. For this we use an illustrative example (following guidelines of Yin [21]) running throughout the current section, featuring the challenge of specifying a financial e-Mediator that offers advices for purchasing insurance products. To do this, it is needed to realize match-making between what the customer wants and what products are available.

We represent the *Customer*, *Advisor*, *Match-maker*, *Request Processing Unit* (we call it "Request Handler", for short) and *Data Search and Processing Unit* (we call it "Data Searcher", for short), as just entities and put them in named boxes, as follows: Customer (**C**); Advisor (**A**); Match-maker (**MM**); Request handler (**R**); Data searcher (**D**) – see Figure 6.

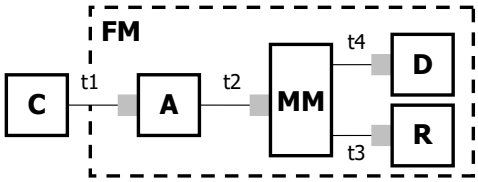

Figure 6. FM business entity model (Adapted from: [1], p. 223)

On the figure, the connections indicate the need for *interactions* between *entities*, in order to achieve the business objective of financial mediation; with each connection, we associate a single *transaction* (**t**): C-A (**t1**); A-MM (**t2**); MM-R (**t3**); MM-D (**t4**). Further, **C** is positioned in the *environment* of the financial mediation system – **FM**, and **A**, **MM**, **R** and **D** together form the **FM** system. Through **t1**, **FM** is related to its *environment* (represented by **C**). Thus, from the perspective of **C**, there is no difference between **FM** and **A**.

That is how we weave language acts in our enterprise modeling, since behind each transaction "stays" the transaction pattern – see Figure 3.

Further, we go for straightforwardly elaborating the above model in terms of semiotic norms, by providing (just for illustrative purposes) several norms:

-------Whenever C has requested advice
If MM has realized match-making
Then A
Is obliged to formulate and deliver an advice
-------Whenever C has requested advice
If R has received submitted customer information
Then R
Is obliged to deliver standardized customer specification
-------Whenever C has requested advice
If D has received information about the type of a customer need
Then D
Is obliged to deliver a candidate-matches list

This norm elaboration is partial – we have only identified several norms to demonstrate how transactions could be usefully elaborated in terms of regulations.

Further, as it concerns public values, our exemplification will also be partial, for the sake of brevity. We will consider just one public value, namely: ACCOUNTABILITY.

In the case of automated financial mediation, it is expected that any design/maintenance/operational failure would be easily traceable and reportable, thus leading to corresponding accountabilities. This would concern responsibility for directing the customer to an inappropriate (with regard to his/her requirements) insurance product, violations with regard to his/her privacy-sensitive data, and so on. Accountability requires the curation of software and algorithms, and also failure of components should be traced.

In this regard, we argue that DESIGN should be important and for this reason, we lean towards weaving accountability in the FM system design – this represents a Value-Sensitive Design – VSD [8].

Our VSD-inspired view on accountability's implications with regard to automated financial mediation featuring insurance products, is depicted in Figure 7:

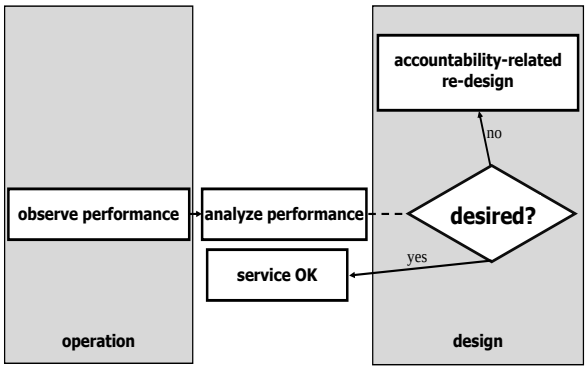

Figure 7. Accountability implications

As it is seen from the figure: (i) at design time we specify what is DESIRED while at run time it shows up what ACTUALLY HAPPENED – what was observed; (ii) if the observed performance corresponds to what was desired, then the service delivery has been adequate; (iii) otherwise, the desired performance was not achieved and corresponding ACCOUNTABILITY would need to be considered and this would only be possible if the accountability value has been reflected in the design, such that the customer can effectively trace back what happened and identify the responsible "actor"(s).

Next to that, establishing accountability through re-designs could possibly lead to some value "tensions", if tracing back what happened would lead to: (i) disclosing privacy-sensitive information; (ii) making technical data explicit including such data that represents copyright-protected "know-how"; (iii) reducing the system availability (during the traceability-related actions).

Finally, as it concerns energy, the FM case is clearly a WATER-driven one because: (i) The business entity model, depicted in Figure 6, would look absolutely the same no matter if the advising is delivered by a human (who in turn collaborates with other humans for the match-making, request processing, and so on) or by a software component (that in turn collaborates with other software components for the match-making, request processing, and so on) => The business processes are ESSENTIALLY STABLE. (ii) At the same time, those business processes can be realized through different "channels", such as human-driven and software-driven (see above) and therefore, the BUSINESS OPERATION IS CHANGEABLE.

In summary, through the FM example, we have demonstrated enterprise modeling activities in 4 perspectives, namely: language acts, regulations, public values, and energy, such that we not only demonstrate how this can be done (and how different modeling activities could be considered in combination) but we have also implicitly justified the importance of each of those enterprise modeling perspectives.

5 Conclusions

Building upon previous research of the authors, this paper concerns the SDBC approach that is about the enterprise-modeling-driven specification of software. Abstracting from the software specification challenge, in the paper, we have mainly focused on the modeling of enterprises, considering in particular four enterprise modeling perspectives, namely: language-acts-driven modeling, regulations-driven modeling, public-values-driven modeling, and energy-driven modeling. Each of those is rooted in particular underlying theories. Not claiming exhaustiveness, we have studied enterprise modeling in all those four perspectives, in isolation and in combination, justifying their importance and illustrating possible modeling activities. Our plans for future research include: (i) Better incorporation of those issues in the SDBC approach; (ii) Realization of bigger case studies, such that a better justification is achieved as it concerns the adequacy of our proposed ways of modeling.

Acknowledgement

This work is supported by: (i) Bulgarian National Science Fund, Project: KP-06-N 32/4/2019; (ii) Ministry of Education and Science, Project of the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES)".

References

1. Shishkov, B.: Designing Enterprise Information Systems, Merging Enterprise Modeling and Software Specification. Springer, Cham (2020).
2. Shishkov B., Quartel D.: Combining SDBC and ISDL in the Modeling and Refinement of Business Processes. In: Manolopoulos Y., Filipe J., Constantopoulos P., Cordeiro J. (eds) Enterprise Information Systems. ICEIS 2006. Lecture Notes in Business Information Processing, vol 3. Springer, Berlin, Heidelberg (2008).
3. Shishkov, B.: Software Specification Based on Re-usable Business Components. Delft University Press - Sieca Repro, Delft (2005).
4. Dietz, J.L.G.: Enterprise Ontology, Theory and Methodology. Springer, Heidelberg (2006).
5. Liu K.: Semiotics in Information Systems Engineering. Cambridge University Press, Cambridge (2000).
6. Shishkov, B., Larsen, J.B., Warnier, M., Janssen, M.: Three Categories of Context-Aware Systems. In: Shishkov, B. (ed.) BMSD 2018. LNBIP, vol. 319, pp. 185–202. Springer, Cham (2018).
7. Shishkov B., Mendling J.: Business Process Variability and Public Values. In: Shishkov B. (ed) Business Modeling and Software Design. BMSD 2018. Lecture Notes in Business Information Processing, vol 319. Springer, Cham (2018).

8. Van den Hoven, J.: Value Sensitive Design and Responsible Innovation. In: Owen, R., Bessant, J., Heintz, M. (eds.) Responsible Innovation: Managing the Responsible Emergence of Science and Innovation in Society. Wiley, Hoboken (2013).
9. Schwartz, S.H.: An Overview of the Schwartz Theory of Basic Values. Online Read. Psychol. Cult. 2(1), 11 (2012).
10. Garvanova M., Shishkov B., Janssen M.: Composite Public Values and Software Specifications. In: Shishkov B. (ed) Business Modeling and Software Design. BMSD 2018. Lecture Notes in Business Information Processing, vol 319. Springer, Cham (2018).
11. Satyasangananda, Swami: Tattwa Shuddhi. Yoga Publications Trust. Munger (2007).
12. MDA: The OMG model driven architecture. http://www.omg.org/mda (2020).
13. Szyperski C.: Component Software, Beyond Object-Oriented Programming. Addison-Wesley, Boston (1998).
14. Shishkov B., Janssen M., Yin Y.: Towards Context-Aware and Privacy-Sensitive Systems. In: Shishkov B. (ed) Business Modeling and Software Design. BMSD 2017. SCITEPRESS (2017).
15. Kruchten, P.: The Rational Unified Process, An Introduction, Addison-Wesley (2003).
16. UML: The unified modeling language. http://www.uml.org (2020).
17. Cockburn, A.: Writing Effective Use Cases, Addison-Wesley (2000).
18. Searle, J.B.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press (1969).
19. Schwartz S.H.: The Refined Theory of Basic Values. In: Roccas S., Sagiv L. (eds) Values and Behavior. Springer, Cham (2017).
20. Friedman B., Hendry D.G., Borning A. (2017) A Survey of Value Sensitive Design Methods. In A Survey of Value Sensitive Design Methods , 1, Now Foundations and Trends, 2017, pp.76-.
21. Yin, R.: Case Study Research: Design and Methods. Sage Publications (1994).