

Consistency, Complementalness, or Conflictation of Enterprise Ontology and Normalized Systems Business Process Guidelines

Philip Huysmans, Dieter Van Nuffel and Peter De Bruyn

Normalized Systems Institute, University of Antwerp, Prinsstraat 13, B-2000 Antwerp, Belgium
{philip.huysmans,dieter.vannuffel,peter.debruyn}@ua.ac.be

Keywords: Normalized Systems, Enterprise Ontology, Business Process Modeling, Enterprise Engineering.

Abstract: Both Enterprise Ontology and Normalized Systems can be considered as design theories, which provide prescriptive guidelines to design systems. Enterprise Ontology explicitly focuses on the design of organizations as being social systems. Originally, Normalized Systems focused on the design of evolvable software systems. However, it has been shown that, building on the Normalized Systems design knowledge, prescriptions for other domains, such as the business process domain, can be proposed as well. This domain seems to overlap at least partially with the domain of Enterprise Ontology, which is used to establish claims concerning process design in various publications. However, both theories are based on completely different kernel theories. Therefore, this paper analyzes to which extent the guidelines proposed for the Normalized Systems Business Processes are consistent, complementing or conflicting with prescriptions from Enterprise Ontology. A consistent set of prescriptions could lead to a more integrated approach for designing integrated organizations, business processes and software systems.

1 INTRODUCTION

The design of organizations and their components such as the organizational structure, business processes, and software systems is an important topic in both practical and scientific communities (Galbraith, 1974; Avenier, 2010). Notwithstanding the attention for this subject, explicit design knowledge in these fields seems limited. For example, Mendling et al. argue that many theoretical frameworks are too abstract, and that more practically-oriented guidelines lack empirical and theoretical support (Mendling et al., 2010). As a result, design of organizational components is often considered as craftsmanship, rather than engineering.

The enterprise engineering paradigm introduces a set of prescriptive design theories which seek to remedy this issue (Dietz et al., 2013). It specifically mentions the β and ν theories as well-founded theories to guide design efforts. The ν -theory states that the design of a system is normalized when a change consists of a set of elementary changes, so that every elementary change does not trigger combinatorial effects (Dietz et al., 2013, p. 101). Normalized Systems (NS) provides concrete guidelines and design patterns to obtain such normalization in software systems (Mannaert and Verelst, 2009). Based on this ap-

proach, normalization of business processes has been researched as well (Van Nuffel, 2011). This research resulted in a set of guidelines which need to be adhered to during business process design. While both Normalized Systems and Normalized Systems Business Processes (NSBP) originally aimed at obtaining designs exhibiting stability as defined in systems theory (Kelly, 2006), it has been argued that the resulting guidelines are in line with existing heuristics of experienced designers. For example, the guidelines presented by Van Nuffel are related to the existing business process literature (Van Nuffel, 2011). Nevertheless, the main contribution of both approaches is the formulation of unambiguous and theoretically founded guidelines based on the single postulate of obtaining the systems theoretic concept of stability. As a result, an approach which resembles traditional engineering, rather than mere craftsmanship, arises on these levels.

The β -theory states that enterprise architecture should be defined as deliberate restriction of design freedom, which should address the function design, construction design, and implementation design of systems (Dietz et al., 2013, p. 100). For example, Enterprise Ontology (EO) prescribes how the construction design of an organization should be made (Dietz, 2006). EO prescribes a clear way of separating

different abstraction levels to be considered in organizations (i.e., ontological, datalogical and infological) and a systematic recurring pattern to model the ontological level.

While the formulation of such theories has been demonstrated to further the field in practice¹, several issues remain. One important issue is the current lack of integration between the specific methods which integrate the different theories (Dietz et al., 2013). This issue has been documented in many studies related to enterprise architecture (Kaisler et al., 2005; Dreyfus, 2007). While certain frameworks, such as TOGAF, focus explicitly on a method to integrate high-level activities such as strategy formulation with detailed software design, no prescriptive methods are present in such frameworks. As a result, the integration of different prescriptive methods can remain very complex. Enterprise architecture researchers have shown that most reports focus on a single architectural layer, and do not address this integration (Schenherr, 2008). In practical projects, this results in local optimizations, which restrict the success of an organizational design as a whole (Kaisler et al., 2005; Dreyfus, 2007).

This means that, within the enterprise engineering community, additional research is required which works towards an integrated method consisting of different prescriptive design theories. For example, both EO and NSBP seem to provide a similar kind of guidelines when used in practical projects, which could indicate that both approaches could be used as complements in various projects. However, a clear obstacle when aiming to apply both approaches simultaneously, is their difference in theoretical backgrounds and abstraction. Therefore, an in-depth analysis regarding the possible compatibility of the guidelines resulting from both approaches is required upfront. Such approach would investigate the extent to which these guidelines are (1) similar in both approaches (i.e., *consistent*), (2) providing additional guidelines (i.e., *complementary*), or contradicting one another (i.e., *conflicting*). It should be noted that this approach does not result in a *theoretical* analysis of EO and NS(BP). On the one hand, NSBP cannot be theoretically EO-compliant, since the distinction axiom is not adhered to: no separation of ontological, infological and datalogical concerns is made. On the other hand, EO has not been developed based on the concept of systems theoretic stability. Notwithstanding this reservation, we are convinced that an analysis of the consistency, complementalness, or conflictation of the practical guidelines of both approaches can contribute to an integrated use of both EO and NSBP in various projects.

¹ See for example www.demo.nl for case studies

2 BACKGROUND

2.1 Normalized Systems

Normalized Systems theory is aimed at studying how modular structures behave under change (Mannaert and Verelst, 2009). Initially, the theory was developed by studying change and evolvability at the software architecture level, by applying concepts such as stability and entropy to the study of the modular structure of the software architecture. Considering the application of systems theoretic stability to software architecture, stability implies that a bounded input function should result in bounded output values, even as $T \rightarrow \infty$. In other words, stability demands that the impact of a change is only dependent on the nature of the change itself. If the amount of impacts is related to the size of the system, a *combinatorial effect* occurs. Research has shown that it is very difficult to prevent CE when designing software architectures. More specifically, it has been proven that CE are introduced each time one of four theorems is violated (i.e., separation of concerns, data version transparency, action version transparency and separation of states).

Various studies have shown that combinatorial effects do not occur solely on the level of software architectures (Van Nuffel, 2011; Huysmans, 2011). On the business process level, it has been argued that business processes at their most basic level (i.e., the “elementary tasks and elementary sequencing and design of these tasks” (Van Nuffel, 2011)) can be considered to be modular structures as well. In this context, business processes have been compared to production lines (Van Nuffel et al., 2009a). In this analogy, a business process flow performs operations on instances of a specific *life cycle information object*. Although production lines may seem highly integrated, they are actually loosely coupling. Every single processing step requires the completion of the previous steps on that instance of a particular product, but it does not require any knowledge of the previous processing steps, nor of the subsequent steps. As a result, changes to individual processes or tasks do not impact other processes or tasks (Van Nuffel, 2011). Put differently, no combinatorial effects occur. More generally, a business process which does not contain combinatorial effects is called a Normalized Systems Business Process (NSBP). In order to achieve such processes, a set of guidelines has been developed, which are based on the more fundamental theorems of Normalized Systems. Together, these guidelines allow the design of business processes without introducing combinatorial effects.

2.2 Enterprise Ontology

Enterprise Ontology (EO) provides an organizational theory (Dietz, 2006) which is based on the Language-Action Perspective (LAP). Consequently, it considers an organization as a social system, and focuses on actor roles as the essential components of organizations. This is important for the goal of this paper, since this background results in the claim that EO provides “a modular framework for business processes” (Dietz, 2003b, p. 1). The EO theory consists of four axioms (i.e., the operation axiom, the transaction axiom, the composition axiom and the distinction axiom) (Dietz, 2006). These axioms allow to specify in more detail what is meant with the “modular construction of business processes” (Dietz, 2003b, p. 18). Business processes are considered to consist of three levels of building blocks. A first type of building block (the atoms) refers to the individual acts performed by actors, as explained by the operation axiom. These atoms can be combined in higher-level building blocks (i.e., molecules), which represent the transactions as explained in the transaction axiom. Multiple transactions can be required to fulfill a certain service to a stakeholder. The collection of these transactions (i.e., a fiber) is then considered to be a business process.

Rather than merely defining business processes using EO concepts, various studies have focused on the design of business processes. For example, the main research question of the paper *Basic Notions Regarding Business Processes and Supporting Information Systems* is “how business processes can be understood in such a way that their continuous and concurrent (re)designing and (re)engineering can be performed more effectively than what is currently the case” (Dietz and Albani, 2005). Another example is the paper *Enhancing the Formal Foundations of BPMN by Enterprise Ontology*, which states 11 propositions which can be derived from EO axioms (Van Nuffel et al., 2009b). Based on the axioms, additional prescriptions for designing business processes are available. For example, the operational cycle (Dietz, 2006, p. 163) states that an actor role needs to be added when a transaction cannot be performed in the same cycle of other transactions. Put differently, this implies that the executor actor of an enclosing transaction needs to be the initiator actor of an enclosed transaction (cf. the composition axiom). Consequently, EO prescribes that certain end-to-end processes which are often defined in practice (e.g., order-to-cash processes) need to be separated.

Various claims have been made that EO can indeed lead to better results when (re)designing pro-

cesses. The abstractions discussed in the distinction axiom are claimed to be “a tremendous advantage for discussing business process optimization” (Dietz, 2006, p. 183), (van Reijswoud, 1999). Moreover, the dedicated model within the DEMO methodology to represent business processes (i.e., the process model) has been claimed to “facilitate the discussion about the redesign of business processes” (Dietz, 2006, p. 183).

2.3 Is it possible to Compare Both Theories?

Caution should be applied when comparing the Enterprise Ontology and Normalized Systems theory, since their intentional application domains vary greatly. Normalized Systems theory focuses on evolvability of software architectures, while Enterprise Ontology attempts to describe coordination in organizations. Nevertheless, the Design Science paradigm argues that the application of theories of related fields is useful to make scientific progress. Moreover, Winter and Albani claim that different design theories can be combined in certain projects (Winter and Albani, 2013). Both the Normalized Systems and Enterprise Ontology theory have already been positioned in a Design Science research framework (Huysmans et al., 2012; Winter and Albani, 2013). Comparing these frameworks indicates an important difference between both theories: Enterprise Ontology builds on communication theories (i.e., the theory of communicative action, the language-action theory and systemic ontology) while Normalized Systems builds on system theoretic and thermodynamic concepts such as stability and entropy.

Notwithstanding this clear difference in kernel theories, remarkable similarities between Normalized Systems and Enterprise Ontology have been discussed as well (Huysmans, 2011). For example, consider the Separation of States theorem. It states that “the calling of an action entity by another action entity needs to exhibit state keeping in normalized systems” (Mannaert and Verelst, 2009). It therefore prescribes how action elements can interact. This impacts, for example, the workflow element, which aggregates action elements. A workflow can reach different states by performing state transitions. A state transition is realized by an action element. The successful completion of that action element results in a defined life cycle state. The workflow specification determines which state transitions can be made. Similarly, the state of a transaction in enterprise ontology is determined by the successful performance of acts. The result of such an act results in the creation of a de-

finer fact. Despite the different terminology, a clear resemblance between Normalized Systems and Enterprise Ontology emerges: state-keeping is enforced in Normalized Systems by defining states, and in Enterprise Ontology by creating facts. These Normalized Systems states are the result of executing actions, whereas the Enterprise Ontology facts are the result of executing acts. Which actions can be performed is determined by the state transitions in Normalized Systems, and occurrence laws in Enterprise Ontology.

Moreover, other attempts have been made to integrate Normalized Systems and Enterprise Ontology theory more directly (Huysmans et al., 2010; Krouwel and Op't Land, 2011; Op't Land et al., 2011). It should be noted that in these efforts, an inductive approach based on concrete artifacts is used, which can be contrasted to a more theoretical approach. Similarly, this paper does not attempt to provide a theoretical comparison, but aims to compare *similar components* of both theories on an *overlapping domain*. The similar components refer to the formulation of *prescriptive guidelines* by both theories. This is important, given the different kinds of theories available in literature (e.g., descriptive theories, explanatory theories, or design theories). In Normalized Systems, such guidelines are referred to by stressing the determinism of design (Van Nuffel, 2011). In Enterprise Ontology, we find clear references to the importance of such guidelines in the definition of architecture, which is "the normative restriction of design freedom" (Dietz, 2006). The *overlapping domain* is the domain of business processes, which is clearly addressed in Normalized Systems Business Processes (cf. Section 2.1). While business processes are defined within EO as well, it should be noted that we interpret the prescriptions of EO not only on the ontological level. In any organization, the ontological models eventually need to be extended to include the infological and datalogical layers, and to specify an implementation. Implementation means "the particular subjects that fulfill the actor roles at a particular time, the particular way in which C-acts are performed, and the particular way in which P-acts are performed." Several publications focused on this subject, which show that a design is obtained which is influenced by EO prescriptions, but which can no longer be considered to be a design of a social system by itself, or be entirely on the ontological level. For example, we mention research to define use cases for information systems based on DEMO models (Dietz, 2003a). This is in line with insights from the generic systems development process (GSDP) (Dietz, 2006, p. 71), which states that a functional specification of an object system needs to be made based on the constructional model of the using

system.

3 APPROACH

In order to compare the guidelines of EO and NSBP, four categories should be considered: (1) *Consistent*: guidelines from NSBP and EO prescribe the same design; (2) *EO-ignorant*: an NSBP guideline which has no similar EO guideline; (3) *NSBP-ignorant*: an EO guideline which has no similar NSBP guideline; (4) *Conflicting*: a NSBP guideline, which prescribes a different design than an EO guideline, or vice versa. Certain guidelines are expected to be consistent, since both EO and NSBP consider business processes as modular structures, and propose guidelines to optimize their design. However, given the different kernel theories of both approaches, and their non-identical goals, certain conflicting guidelines could be identified. Moreover, neither EO or NSBP claim to be complete. The claim from Dietz that "we do not intend to claim that ... even the whole ψ -theory is a sufficient basis for achieving optimally performing enterprises" (Dietz, 2006, p. 81) indicates the validity of the EO-ignorant category. The claim from Van Nuffel that NSBP guidelines are necessary, but not sufficient, indicates the validity of the NSBP-ignorant category. We will adopt the work of Van Nuffel as our starting point as it explicitly lists a set of 25 guidelines, whereas the guidelines from EO have not been formally consolidated in such list exhaustively enumerating all guidelines incorporated in the method. Further, given this starting point to determine for each guideline to which category it belongs, the NSBP-ignorant category will not be required in this paper.

The authors of this paper independently made a classification of the NSBP guidelines. After integrating the result, differences were discussed, and the assessment was iteratively refined. All three authors have a sufficient background in both EO and NSBP. The NSBP PhD dissertation (Van Nuffel, 2011) and EO book (Dietz, 2006) were used as reference materials. Several academic publications were used for additional details. Moreover, several cases (see e.g., (Van Nuffel, 2011), (Dietz, 2006), <http://www.demo.nl>) were consulted as an application of the guidelines.

4 COMPARISON

Within this section, the actual comparison between the practical guidelines resulting from the two theoretical approaches is made. Our discussion will

follow the division made within the PhD of Van Nuffel (Van Nuffel, 2011): first, the general guidelines with respect to identifying business processes are discussed. Second, the comparison continues with the three additional guidelines that in specific cases identify business processes. Third, the comparison continues with the guidelines determining individual tasks, and finally, the auxiliary guidelines are investigated. The business process patterns discussed in the PhD of Van Nuffel (Van Nuffel, 2011) focus on issues not discussed by Enterprise Ontology, and are therefore not taken into account. This section lists the names of the guidelines in *italic* and **bold** font. Next, the guideline is summarized in *italic*. Then, the consistency, complementalness or conflict with EO is discussed. An overview of these discussions is provided in Table 1.

4.1 General Business Process Guidelines

1.1 Elementary Business Process. *A Business process denotes a constrained sequence — i.e., sequence, iteration or selection — of individual tasks representing state transitions in the life cycle of a single life cycle information object.* Within Enterprise Ontology (EO), a P-fact is a factum, which is defined as “*the result or the effect of an act*” (Dietz, 2006, p. 42). Therefore, facta “can be conceived as status changes of ... an object in some class” (Dietz, 2006, p. 42). Furthermore, the order in which facta occur is determined by so-called occurrence laws (Dietz, 2006, p. 43). The transaction is thus about a unique P-fact transcending the transaction pattern, which can be considered to be somewhat consistent with a NS business process which is about state transitions of a single life cycle information object as stated by NSBP. The one-to-one relationship between a transaction and a P-fact is in our opinion conceptually consistent with the one-to-one relationship between a single life cycle information object and a business process.

1.2 Elementary Life Cycle Information Object. *an information object not exhibiting state transparency is a life cycle information object.* Whereas NSBP prescribes the criterion of state transparency (i.e., when no proper state transitions should be made explicit (Van Nuffel, 2011, p. 118)) to define whether an information object is a genuine life cycle information object processed in a business process, Enterprise Ontology does not explicitly state a rule, criterion or law that in all circumstances denotes what a single P-fact is. There are evidently ways and requirements a

P-fact should adhere to, but no general identification mechanism seems to be made explicit:

- “We conceive the result of a production act as a particular change in the state of the system’s object world” (Dietz, 2006, p. 58);
- “The object world reflects the produced things (e.g., goods or services) that are delivered to the elements in the environment” (Dietz, 2006, p. 58).

As a consequence, – although it could be argued that only most fine-grained production facts exist (and therefore, that production facts are defined unambiguously), but that they can be aggregated to simplify models – it seems that identification of production acts in EO is not unambiguous: it depends on what is considered to be the system and environment, and different production facts can be identified depending on the aggregation level taken into account. Moreover, elementary life cycle objects can also refer to infological and datalogical production facts. Therefore, the authors categorize this guideline as EO-ignorant.

1.3 Aggregated Business Process. *In order to represent an aggregated business process, an aggregated life cycle information object has to be introduced.* In EO, a business process is based on the composition axiom: “a business process is a collection of causally related transaction types, such that the starting step is either a request performed by an actor role in the environment or a request by an internal actor role to itself” (Dietz, 2006, p. 103). Based on this definition, the operational cycle (Dietz, 2006, p. 163) can be understood, which specifies that certain end-to-end processes cannot be considered as causally related transactions. Since the NSBP guideline is explicitly aimed towards representing *any* required end-to-end process, both theories are conflicting in most situations.

1.4 Aggregation Level. *Tasks performed on a different aggregation level denote a separate business process.* Although in the PSD-diagrams the causal and conditional links are enriched with cardinalities that describe the relationship between different transactions, nowhere is indicated that when an analyst discovers an one-to-many relationship between two candidate transactions, both should be separated. Furthermore, this latter relates to the aggregation level on which production facts are defined, since a production fact defines a transaction. Again, this does not result in a guideline to actually separate the transactions. Therefore, EO seems to be ignorant with respect to this design issue.

1.5 Value Chain Phase. *The follow-up of an organizational artifact resulting from a value chain phase denotes a different business process.* While some arguments can be made for the consistency of this guideline, the most important argument seems to indicate a conflict. For example, the operation axiom might indicate value chain phases as separate transactions, although it is dependent on the aggregation level on which the P-facts are defined. Moreover, the composition axiom illustrates the possible nesting required to integrate the different phases. However, the transaction axiom results in design decisions like explicitly stating that the Order phase belongs to the Delivery process in a typical Customer Order process scope. With respect to the latter, NSBP clearly state that these phases should be separated as they denote separate concerns (Van Nuffel, 2011, p.132-34). In this way, NSBP seems to consider concerns a level “deeper” as it explicitly considers a delivery not to belong to the Order Phase, but as a separate process in the aggregated business process Customer Order. Therefore, both theories seems to disagree with respect to this design issue.

1.6 Attribute Update Request. *A task sequence to update an attribute of a particular life cycle information object that is not part of its business process scenarios, is represented by an Attribute Update Request business process.* The guideline prescribes to separate state transitions dealing with modifying an attribute of a life cycle information object that does not belong to the business process scenarios (i.e., included in the process). Enterprise Ontology however, considers such requests to change to be part of the transaction. Mostly, they can be represented by one of the four cancelation patterns. As such, this represents a conflict between the two theories, although they comply with each other on modifications that do belong to the business process scenarios.

1.7 Actor Business Process Responsibility. *Actor business process responsibility indicates a separate business process if different actors are responsible for a different set of tasks, of which the task allocation belongs to different process owners.* The operation axiom declares actor roles to denote chunks of authority, responsibility and competence. Furthermore, following Enterprise Ontology, a single transaction can only be executed by a single actor role. In this way, this notion is equivalent to stating that state transitions of a particular life cycle information object being part of the responsibility of a particular process owner denote a separate business process. Furthermore, in addition to EO, also NSBP identifies the only vaguely

described notion of process ownership within literature. As a consequence, NSBP opts for a clear identification of such process ownership, which seems to be very closely related to EO’s notion of authority.

1.8 Notifying Stakeholders. *Because notifying, or communicating a message to, stakeholders constitutes an often recurring functionality in business processes, a designated business process will perform the required notification.* EO considers notifying stakeholders as performing coordination acts, which are part of an ontological transaction that creates a single P-fact. However, NSBP identifies the concern of notifying stakeholders to clearly differ from the concerns taken care of by other business process (e.g., delivering an order, recruiting an employee, etc.): “delivering a message in the correct format to the intended recipients at the right time in an unchanged format, with the related fault handling” (Van Nuffel, 2011, p.143). These concerns refer to implementation details, which are not considered on the ontological level. Therefore, EO theory is ignorant with respect to this design guideline.

1.9 Payment. *Because paying a particular amount of money to a particular beneficiary constitutes an often recurring (technical) functionality in business processes, a designated (technical) business process will perform the required payment.* The payment business process/transaction is identified by both theories, and can be considered as consistent. Various DEMO cases illustrate this. It should be noted that NSBP requires that at least the execution phase of payment processes is implemented using a reusable business process, in order to prevent combinatorial effects. This is not clear from the DEMO cases, which explicitly define multiple payment transactions.

4.2 Business Process Guidelines

2.1 Product Type. *A different type of product or service denotes a main concern, and thus indicates a different business process.* The composition axiom seems to indicate that Enterprise Ontology also recognizes the existence of transactions that although being enclosed by the same transaction, do constitute individual and independent transactions based on a product structure. But again, no clear rules could be identified, implicitly stated by “one could apply a finer-grained product structure” (Dietz, 2006, p. 170). The notion of a product type defined by Van Nuffel (Van Nuffel, 2011, p. 149) allows some interpretation as well, namely the domain expert who will identify the characteristic dimensions on which product types

exhibit similar properties. As a consequence, we categorize this design issue as an EO-ignorant one.

However, if the Logistics example discussed by NSBP is taken into account, the design issue also seems to indicate conflicting statements by the two theories. The NSBP separate the Logistics processes based on the following types: non-food, food, quickly rotating, slowly rotating, and so on. On the other hand, EO theory seems to declare that these product types do not cause another type of P-fact to be created, and thus no separate transaction to be executed. This could indicate a potential conflict.

2.2 Stakeholder Type. *Stakeholder type should principally be considered a cross-functional concern (i.e., a concern which does not require a life cycle information object by itself), expect for those business processes where the stakeholder type denotes the life cycle information object.* Whether the theories comply, comes down to the question: does EO consider a transaction to be independent from the actor role for which it is potentially performed? In the PhD of Van Nuffel, a case about Human Resources (HR) processes is discussed in which it is clearly demonstrated that the assignment processes for a statutory employee and a non-statutory employee differ. Based on the authors' knowledge, EO does not provide any rule or prescription about the potentially different nature of a transaction. For example, in the Educational Administration case, no separate transactions are created based on different student types.

2.3 Access Channel. *The concept of an access channel indicates a cross-functional concern.* In EO publications no explicit referral to this design question could be found. However, implementation is explicitly out of scope for EO: EO "fully abstracts from the implementation [of C-acts]", which includes "the particular way in which C-acts are performed" (Dietz, 2006, p. 83). Consequently, it can be argued that the theories comply as EO does not explicitly states a different access channel denotes a separate transaction. Consider in this context the pizzeria case (Dietz, 2006, p. 166). The transaction *T01: Completion* contains all access channels to place an order.

4.3 Task Guidelines

3.1 A Single Functional Task - Overview. *A task represents a functional entity of work that either results in a single state transition of a single information object type, or refers to an Update or Read task on a single information object type.* Where NSBP specifically describes what a single task (or

step within a business process) can be, our analysis of the EO fails to find equivalent rules. Of course, the transaction axiom identifies single acts (resulting in facts) within the transaction which might indicate consistency. However, the authors seem to find more evidence to categorize it as EO-ignorant. For instance, consider the acceptance of a stated P-fact consisting of an evaluation of its quality by performing three quality tests and then communicating the outcome to the initiator actor role which is authorized, responsible and competent to accept the P-fact, who will communicate it to the executor. EO considers this example to be part of the Accept C-act whereas NSBP prescribes to separate it in five different tasks, and two instances of the Notification business process. Thus, based on our analysis, we consider it to be EO-ignorant.

3.2 CRUD Task. *Each of the Create - Read - Update - Delete (CRUD) operations constitutes a single task.* Since these tasks are on the infological and datalogical layers, this guidelines is EO-ignorant.

3.3 Manual Task. *Every manual task of which the initiation and completion has to be known, has to be designed as a separate task.* EO makes abstraction of the implementation of C- and P-acts (also see discussion of 2.3 Access Channel). Therefore, EO is ignorant with respect to this guideline.

3.4 Managing Time Constraint. *The management of a time constraint denotes a separate task because it represents the individual concern of managing a particular time constraint.* In EO, a time aspect only seems present in the time-aspect of the proposition of a P-fact (Dietz, 2006, p. 84) and self-initiating transactions (Dietz, 2006, p. 99). However, EO makes no claims whatsoever with respect to (not) separating an individual time constraint. As such, we categorize the guideline to be EO-ignorant.

3.5 Business Rule Task. *A single business rule should be separated as a single task.* An individual business rule should be isolated in its designated task following NSBP. EO acknowledges that business rules can sometimes be existential laws, as expressed in the state model, or action rules, which are expressed in the action model (Dietz, 2006, p. 196). In this sense, both seem to be consistent. However, EO does not explicitly states that every single business rule should be isolated. Therefore, EO seems to be rather ignorant to this design issue.

3.6 Bridge Task. *When a business process instance operating on an instance of life cycle information object type I has to create a business process instance of another life cycle information object type L, this functionality is designed as a bridge task that initiates the creation of the instance of the life cycle information object L, and represents a state transition on the instance of I.* As already illustrated above, the composition axiom of EO denotes the nesting of transactions. As such, it is illustrated that the Request C-act can be “triggered” by another transaction (i.e., the executor of an enclosing transaction can initiate an enclosed transaction). The Result structure analysis step of the DEMO methodology also adds to this. Conceptually, this is what a bridge task represents: it triggers the execution of another business process.

3.7 Synchronization Task. *When a business process instance operating on a life cycle information object I has to inform a business process instance of another life cycle information object L, a synchronization task, representing a state transition on the instance of I, alters the state of the business process instance of L.* The NSBP synchronization task conceptually equals the waiting conditions specified in the EO model based on the Result structure analysis, following the composition axiom.

3.8 Synchronizing Task. *A synchronizing task represents the task receiving information from another business process’s execution, in order to continue the business process control flow.* Equivalent to the Bridge task, also the Accept C-act in the EO transaction pattern represents conceptually the same as a synchronizing task. It allows the enclosing transaction/business process to continue, and thus is the end of the waiting condition.

3.9 Actor Task Responsibility. *A task cannot consist of parts that are performed by different actors.* Here NSBP is consistent with EO, as the operation axiom states that actor roles are elementary chunks of authority, responsibility and competence. Thus the fact that another actor role is authorized, responsible and competent to perform a particular task, suffices to split this task from any other task another actor role is authorized, responsible, and competent to execute.

4.4 Auxiliary Guidelines

4.1 Unique State Labeling. *Each state of a life cycle information object has to be unique.* The first auxiliary guideline, *Unique State Labeling*, states that each state of a life cycle information object should be

unique. Thus, it indicates the necessity to uniquely define the states a business process can transverse. Also EO identifies unique labels as each coordination act and each transaction are uniquely labeled; and even more it states that facts can be created, but cannot be undone (Dietz, 2006, p. 82). Thus theories are considered to be consistent.

4.2 Unique State Property. *A life cycle information object instance can only be in a single state at any time.* Also EO declares a transaction has a unique status: the last performed fact, which is defined in EO as a state transition in the C- or P-world (Dietz, 2006, p. 82). Thus theories are considered to be consistent.

4.3 Explicit Business Process End Point. *If a business process type has multiple possible outcomes, each of these scenarios should have its dedicated end point reflecting the respective end state of a business process instance.* EO specifies through its transaction patterns (basic-standard-cancelation) that every scenario should be explicitly described. In this way, it is consistent with NSBP as every business process’ execution results in a specific end point/state, and not in a general state “finished”.

4.4 Single Routing Logic. *A split/join element in a business process’s control flow should only represent a single split or join routing expression.* Essentially EO does not discuss this proposed guideline, so it is considered to be EO-ignorant. However, it can be argued that both theories are consistent because within the transitions between the different C-facts and P-fact that are exhaustively described in the transaction pattern, no violation to the NSBP guideline was identified. Further research should identify whether this non-violation is purposefully – and thus the theories are consistent – or rather by chance – and thus remains EO-ignorant.

5 DISCUSSION

Table 1 summarizes the comparison made in the previous section. A bullet denotes that the identified category is determined without any doubt. An open circle means the categorization still needs further elicitation as a unique categorization could not be identified.

When scanning the table, it can be argued that the theories comply on many points (i.e., at least 10 out of 25 guidelines are consistent), indicating that a surprising overlap exists between guidelines prescribed by EO and NSBP, given their different the-

Table 1: Consistency of NSBP guidelines and EO.

	Consistent	EO-ignorant	Conflict
1.1 Elementary Business Process	●		
1.2 Elementary Life Cycle Information Object		●	
1.3 Aggregated Business Process			●
1.4 Aggregation Level		●	
1.5 Value Chain Phase			●
1.6 Attribute Update Request	○		○
1.7 Actor Business Process Responsibility	●		
1.8 Notifying Stakeholders		●	
1.9 Payment	●		
2.1 Product Type		○	○
2.2 Stakeholder Type		●	
2.3 Access Channel		○	○
3.1 A Single Functional Task - Overview		●	
3.2 CRUD Task		●	
3.3 Manual Task		●	
3.4 Managing Time Constraint		●	
3.5 Business Rule Task		●	
3.6 Bridge Task		●	
3.7 Synchronization Task		●	
3.8 Synchronizing Task		●	
3.9 Actor Task Responsibility		●	
4.1 Unique State Labeling		●	
4.2 Unique State Property		●	
4.3 Explicit Business Process End Point		●	
4.4 Single Routing Logic		○	○

oretical backgrounds. The EO-ignorant category is mostly discovered in the NSBP task rules. Almost all observations can be contributed due to the different abstraction level (EO does not consider these design questions), or the lack of a clear available answer in the different publications (e.g., Stakeholder Type). Consequently, NSBP seems to answer design questions EO does not answer or does not consider. Regarding the conflicting guidelines, some genuine contradictions (e.g., Aggregated Business Process) were identified. These conflicts should be clarified in future research, especially because most conflicts occur in the core (i.e., the first twelve) NSBP guidelines.

Moreover, more in-depth analysis is required, since the reason for consistent design decisions may differ. For example, consider the library case. The transaction *T03: Reduced fee approval* is a separate transaction in EO because it is executed by a different actor (Dietz, 2006, p. 144). In NSBP, it is a

different transaction because “it denotes a separate concern, because it recurs in at least two situations” (Van Nuffel, 2011, p. 217) (i.e., when creating a new member and when collecting the yearly fee). Additionally, the NSBP-ignorant category needs to be elaborated upon as well. For example, various coordination acts are not required to be modeled in NSBP, for example when they are implicit. The explicitation of this category could especially aid the completeness of NSBP models.

Nevertheless, the authors hypothesize that — given the consistency between both theories and under the condition that the different abstraction levels on which they clearly operate do outweigh the contradictions, or that contradictions could be resolved by clearly identifying the abstraction levels on which both theories have their proven scientific importance — a method combining both theories to analyze businesses can be proposed. We will further elaborate on this method and its applications in future research.

6 CONCLUSIONS

In this paper, we explored to which extend the prescriptive guidelines related to the business process domain of EO and NSBP are consistent, complementary, or conflicting. We explained how both approaches offer theory-based guidelines to design business processes, and discussed in detail the assessment of the various NSBP guidelines. Moreover, we suggested several possibilities for further research, to work towards an integrated method for organizational design.

ACKNOWLEDGMENTS

P.D.B. is supported by a Research Grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

REFERENCES

- Avenier, M.-J. (2010). Shaping a constructivist view of organizational design science. *Organization Studies*, 31(9-10):1229–1255.
- Dietz, J. (2003a). Deriving use cases from business process models. In Song, I.-Y., Liddle, S., Ling, T.-W., and Scheuermann, P., editors, *Conceptual Modeling - ER 2003*, volume 2813 of *LNCS*, pages 131–143. Springer Berlin Heidelberg.
- Dietz, J. L. G. (2003b). The atoms, molecules and fibers of organizations. *Data Knowl. Eng.*, 47:301–325.

- Dietz, J. L. G. (2006). *Enterprise Ontology: Theory and Methodology*. Springer.
- Dietz, J. L. G. and Albani, A. (2005). Basic notions regarding business processes and supporting information systems. *Requir. Eng.*, 10:175–183.
- Dietz, J. L. G., Hoogervorst, J., Albani, A., Aveiro, D., Babkin, E., Barjis, J., Caetano, A., Huysmans, P., Iijima, J., Van Kervel, S., Mulder, H., Op't Land, M., Proper, H. A., Sanz, J., Terlouw, L., Tribolet, J., Verelst, J., and Winter, R. (2013). The discipline of enterprise engineering. *International Journal of Organisational Design and Engineering*, 3(1):86–114.
- Dreyfus, D. (2007). Information system architecture: Toward a distributed cognition perspective. In *ICIS 2007 Proceedings*. Paper 131.
- Galbraith, J. R. (1974). Organization design: An information processing view. *Interfaces*, 4(3):28–36.
- Huysmans, P. (2011). *On the Feasibility of Normalized Enterprises: Applying Normalized Systems Theory to the High-Level Design of Enterprises*. PhD thesis, University of Antwerp.
- Huysmans, P., Bellens, D., van Nuffel, D., and Ven, K. (2010). Aligning the constructs of enterprise ontology and normalized systems. *Lecture notes in business information processing*, 49(1):1–15.
- Huysmans, P., Oorts, G., and De Bruyn, P. (2012). Positioning the normalized systems theory in a design theory framework. In Shishkov, B., editor, *Proceedings of the Second International Symposium on Business Modeling and Software Design (BMSD2012)*, pages 33–42, Geneva, Switzerland.
- Kaisler, S. H., Armour, F., and Valivullah, M. (2005). Enterprise architecting: Critical problems. In *Proceedings of the 38th Hawaii International Conference on System Sciences*, volume 8, page 224b, Los Alamitos, CA, USA. IEEE Computer Society.
- Kelly, D. (2006). A study of design characteristics in evolving software using stability as a criterion. *Software Engineering, IEEE Transactions on*, 32(5):315–329.
- Krouwel, M. and Op't Land, M. (2011). Combining demo and normalized systems for developing agile enterprise information systems. In Albani, A., Dietz, J., and Verelst, J., editors, *Proceedings of EEWC 2011*, pages 31–45. Springer-Verlag.
- Mannaert, H. and Verelst, J. (2009). *Normalized Systems—Re-creating Information Technology Based on Laws for Software Evolvability*. Koppa, Kermt, Belgium.
- Mendling, J., Reijers, H. A., and van der Aalst, W. M. P. (2010). Seven process modeling guidelines (7pmg). *Inf. Softw. Technol.*, 52(2):127–136.
- Op't Land, M., Krouwel, M. R., van Dipten, E., and Verelst, J. (2011). Exploring normalized systems potential for dutch mod's agility. In Harmsen, F., Grahlmann, K., and Proper, E., editors, *Proceedings of PRET 2011*, pages 110–121. Springer-Verlag.
- Schenherr, M. (2008). Towards a common terminology in the discipline of enterprise architecture. In *ICSOC Workshops'08*, pages 400–413.
- Van Nuffel, D. (2011). *Towards Designing Modular and Evolvable Business Processes*. PhD thesis, University of Antwerp.
- Van Nuffel, D., Mannaert, H., De Backer, C., and Verelst, J. (2009a). Deriving normalized systems elements from business process models. In Boness, K., editor, *Proceedings of ICSEA 2009*, pages 27–32. IEEE Computer Society.
- Van Nuffel, D., Mulder, H., and Van Kervel, S. (2009b). Enhancing the formal foundations of bpmn by enterprise ontology. In *CIAO! / EOMAS*, pages 115–129.
- van Reijswoud, V. (1999). Model based business system transformation. In Pries-Heje, J., Ciborra, C., Kautz, K., Valor, J., Christiaanse, E., and Avison, D., editors, *Proceedings of ECIS 1999*, pages 600–615, Copenhagen.
- Winter, R. and Albani, A. (2013). Restructuring the design science research knowledge base. In Baskerville, R., De Marco, M., and Spagnoletti, P., editors, *Designing Organizational Systems*, volume 1 of *Lecture Notes in Information Systems and Organisation*, pages 63–81. Springer Berlin Heidelberg.