Using Entropy’s Justificatory Knowledge for a Business Process Design Theory

Peter De Bruyn
University of Antwerp
peter.debruyn@uantwerp.be

Philip Huysmans
University of Antwerp
philip.huysmans@uantwerp.be

Gilles Oorts
University of Antwerp
gilles.oorts@uantwerp.be

Herwig Mannaert
University of Antwerp
herwig.mannaert@uantwerp.be

Jan Verelst
University of Antwerp
jan.verelst@uantwerp.be

Abstract

Business processes and the data they generate are often complex, causing uncertainty for managers who have to take decisions based on this information. At this moment, a theoretical understanding of the complexity associated with these business processes and their data is absent. In this paper, we therefore suggest the concept of entropy from thermodynamics as justificatory knowledge for governing design science research efforts. This might lead to a set of prescriptive guidelines for creating business processes exhibiting a low degree of complexity. Our analysis shows that entropy indeed seems a viable concept for this purpose.

1. Problem statement

Contemporary firms need high-quality information to control their business. Recently, the possibilities for gathering, storing and mining of large amounts of data have improved significantly (e.g., RFID technologies allowing to systematically collect data at predefined locations at large scale). However, the design of business processes which are the basis to gather and store this data should not be done arbitrarily. For example, the current interest in “Big Data” within the information systems (IS) domain indicates that many organizations apparently have issues to extract relevant information from the complex data they have at their disposal. Decisions regarding which information should be stored (and hence, how the underlying business processes should be designed) therefore become crucial. Nevertheless, very few clear guidelines or best practices are available to design business processes, mainly providing overall project management related best practices (see e.g., [23]). In addition, those guidelines which are practically oriented are mostly lacking empirical and/or theoretical grounding [29]. Consequently, a theoretical understanding of the complexity associated with these business processes is absent [4] and managers wanting to take important management decisions based on precise data might be covered in uncertainty.

The Design Science Research (DSR) methodology guides researchers who aim to create artifacts, complying with academic rigor, but providing practical relevance as well [14,16]. While some authors consider theory to be external to DSR efforts [28], there is growing consensus that theory should be an essential part of DSR [14,37,38]. As a consequence, the term Information Systems Design Theory (ISDT) was introduced to represent “a prescriptive theory based on theoretical underpinnings which says how a design process can be carried out” [38:37]. Hence, theory and knowledge can be seen as both a possible output of DSR or input [24]. In terms of the input knowledge base, it has been argued that the design requirements and/or process can be driven by descriptive “kernel theories” (originating from natural science, social science or mathematics) [38], prescriptive design theories [14,39], or even justificatory knowledge in general [14]. Justificatory knowledge aims to explain “why an artifact is constructed as it is and why it works” [14:328]. This further supports the characterization by Peffers et al. [32] of IS as an applied research discipline in which theory from other disciplines is frequently leveraged to solve problems between information technology and organizations. While DSR and ISDT have traditionally been proposed in an IS context, the methodology might be useful for designing business processes (e.g., towards low complexity) as well. First, business processes are often implemented or supported by information technology (IT), and might thus be argued to be at least related to the IS field. Next, some authors have argued that management and organizational research efforts should
be aimed at designing useful and practical solutions for their field, drawing upon the DSR methodology [2].

In engineering sciences, phenomena related to the complexity of modular structures have traditionally been studied by means of the concept of entropy [6]. Business processes have been argued to be modular structures as well [30]. Therefore, entropy and its related reasoning in other disciplines might constitute a relevant justificatory knowledge base as part of DSR efforts towards prescriptive guidelines for designing business processes exhibiting a low degree of complexity. In order to assess its suitability, we will first discuss some basic background on entropy with some related work in Section 2. Next, we elaborate in Section 3 on our approach of applying entropy (i.e., in terms of microstates and macrostates) to the business process level. The relevance and implications for business practice will be discussed in Section 4. An overview of some design principles to devise business processes in such way that the entropy during and after their execution is minimized, will be presented in Section 5. Finally, our discussion and conclusions are offered in Sections 6 and 7, respectively.

2. The Quest for Justificatory Knowledge

The notion of entropy, typically referring to the second law of thermodynamics, provides conceptualizations and measurements of complexity in modular systems and provides some general principles in order to control or reduced it. The concept is a fundamental property in engineering sciences and has been described and studied from many different perspectives, basically all describing the irreversibility of nature. More specific interpretations of the entropy concept include (1) complexity, perceived chaos or disorder [3], (2) uncertainty or lack of information [35] and (3) the tendency of constituent particles in a system to dissipate or spread out [25]. Most of these interpretations can essentially be traced to the phenomenon that modules (or in their most elementary form: particles) in a system have the inherent tendency to interact with one another (i.e., being coupled) in an uncontrolled way unless additional structure (i.e., energy or effort) is introduced in the system.

One of the operationalizations of entropy which appears to be the rather easily transferable to other application domains, seems the statistical thermodynamics perspective towards entropy. Here, it was defined by Boltzmann in 1872 as the number of possible microstates consistent to the same macrostate of that system [6]. The macrostate refers to the whole of microscopic properties of the constituent parts of the system (i.e., modules and particles). Generally, a myriad of different combinations of microstates may result in a particular macrostate (i.e., many different configurations of the molecules embedded into the container resulting in the same temperature). The higher the number of microstates consistent with a macrostate (i.e., multiplicity \( \Omega \)), the larger the degree of entropy becomes according to statistical thermodynamics.

Some authors already related entropy concepts to organizational phenomena. For example, Janow [20] used Shannon’s entropy approach [2] to claim that organizational productivity and decision making becomes slower as they grow. At an interorganizational level, entropy was found relevant when measuring the degree of industry concentration [13,18] and corporate diversification [19,31]. Lev [26], Ronen & Falk [33], and Abdel-Khalik [1] analyzed the design and information content of financial statements from an entropy viewpoint. Further, the concept has been employed to model both software development processes and project management activities [34,36] as the architecture of (enterprise) software applications themselves [5,12,15,27]. The papers of Jung [22] and Jung et al. [21] seem most related to our approach (cf. infra). They defined uncertainty measures of process models to grasp the variability in workflow process models (i.e., how much information, regarding which tasks will be executed, can we derive from a process?). However, the application of the concept of entropy as defined in statistical thermodynamics (in terms of microstates and macrostates) to the business process domain has been, to the best of the authors’ knowledge, not yet explored before. Also, our approach focuses on deriving design guidelines for business processes such that the uncertainty or complexity during or after execution is minimized. For this purpose, a run-time perspective (i.e., “space”) is needed, in addition to a mere static design perspective. Additionally, positioning the concept of entropy as a clear part of justificatory knowledge to prescriptively design (business process) IS artifacts has not been found in literature.

3. Leveraging Statistical Thermodynamics

Entropy to the Business Process Level

While Hevner states that “it is often a stretch to find kernel theories for the creative activities of design research” [17:90], Gregor and Jones argue that it is “difficult to envision situations where there is a complete absence of justificatory knowledge” [14:328].
Consequently, without claiming to fully adopt a kernel theory as the basis of our design research efforts, we adopt a limited part of natural science thermodynamics (i.e., the entropy concept and definition) serving as justificatory knowledge for our envisioned business process design theory. Indeed, besides offering inspiration for our design guidelines (see Section 5), it can help in explaining why our design theory was devised as it is and why it should lower business process complexity. In order to extend the concept of entropy to business process analysis, we thus need to propose a definition of entropy, microstates and macrostates in such business process context. However, we first need to define a business process instantiation space, as we will show that leveraging and operationalizing the entropy definition from statistical thermodynamics to control business process complexity seems mostly relevant in a run-time environment. Afterwards, we will define and interpret microstates, macrostates and statistical thermodynamics entropy in a business process context.

3.1. Defining the run-time instantiation space of a business process

Many (diagnostic) management decisions relate to the run-time perspective of business processes, as properties like a realized throughput time or the costs of specific instantiated products or services only exist after execution (and not in a mere design-time perspective). Consequently, run-time instantiation space of business processes has to be defined. For this purpose, we propose to adopt the notation we described in [7] and consider a business process type $BP_i$ as a flow which constitutes a set of consecutive task types $k$ (including selections, iterations, etcetera) on a lifecycle information object $IO_l$ (e.g., a delivery or an invoice) with the intent to attain a certain business goal. Such task types for a delivery business process might for instance comprise the assurance of the client’s creditworthiness or the analysis whether the requested good is still in stock.
Hence, we might express that business process type 1 consists out of tasks 1 to 5: $BP_1 = \{t_1, t_2, t_3, t_4, t_5\}$. Given our complexity control purpose, we propose to identify task types based on the concept of an “information unit”, i.e., each part of a business process space of which information regarding throughput time, quality, costs, resource consumption or executing actor is relevant. These tasks might all be executed in a pure consecutive way or exhibit a more advanced path of execution (e.g., tasks 3 and 4 might be part of an exclusive gateway, meaning that only one of them will be executed for the completion of the considered business process). Each time a delivery is requested by a client, a new instance $j$ of the business process is initiated and an instantiated information object (i.e., a specific product delivery instance) proceeds through each step in the business process. As a consequence, each business process instantiation $BP_{i,j}$ has an index $i$ referring to the business process type and an index $j$ referring to the business process instance of a particular business process type. In the same way, each business process instance contains a set of instantiations of its constituting tasks $t_{k,m}$ (index $k$ identifying the task type and index $m$ the task instance). When considering only 3 instantiations of our previously defined business process $BP_1$, one possible business process instantiation space might be:

$$BP_{1,1} = \{t_{1,1}, t_{2,1}, t_{3,1}, t_{5,1}\}$$

$$BP_{1,2} = \{t_{1,2}, t_{2,2}, t_{4,1}, t_{5,2}\}$$

$$BP_{1,3} = \{t_{1,3}, t_{2,3}, t_{3,2}, t_{5,3}\}$$

Ignoring the groupings depicted by the dotted lines, a graphical representation of the considered business process instantiation space is shown in Figure 1. In this figure, we have chosen to adopt the widely adopted Business Process Modeling Notation (BPMN). However, it is clear that our instantiation space is defined in such a way that it is independent of any specific modeling notation and can thus be represented in other modeling notations as well. The authors would like to remark that few or no business process modeling notations do incorporate a practical way to visualize the run-time dimensions of a business process and its instances. As this perspective is nevertheless needed for our purpose, we were obligated to employ an existing notation and make some pragmatic compromises regarding their modeling standards to allow for our instantiation space to be represented.

3.2. Defining the run-time instantiation space of a business process

We explained in Section 2 how entropy generation can be related to the degree of interaction and coupling between the particles or modules making up a system, during and after its lifetime. This nicely correlates with the behavioral characteristics of the entropy concept. Consequently, for each modular system potentially exhibiting behavior, entropy could be studied. In previous work, we therefore already briefly argued that entropy reasoning can be applied to organizational systems as well [7,8,9,10]. Accordingly, macrostates and microstates should be defined.

Reasoning by the thermodynamics analogy, the individual task instantiations can be regarded as the “particles” in the business process system. Whereas in typical thermodynamics, properties as speed and position of the particles are studied, its counterpart in the context of business tasks might include typical properties of individual task instantiations such as throughput or cycle time, quality and other output related measures, involved costs, other resources consumed, or the actor executing the task instance.

Hence, the microstate in the defined instantiation space is defined by the union of the values of above listed properties for each individual particle in the business process instantiation space (e.g., task instantiation). When for instance focusing on the costs $C$ of each individual task instantiation, the microstate then becomes: $\{C(t_{k,m})\}_{k,m}$.

The macrostate of this instantiation space is the (aggregated) information available for the observer, generally entailing unrecoverable loss of information. Consequently, in a business process context, a macrostate could be considered as referring to typical observable information of business processes such as the total throughput or cycle time, quality or output measures, total costs, resource consumption, et cetera, regarding a process as a whole or some of the combination of some of its tasks.

3.3. Defining and interpreting entropy

Analogous to traditional thermodynamics, our entropy conceptualization here is also defined as being proportional to the number of microstates consistent with a single macrostate (i.e., multiplicity $\Omega$). Analyzing a set of instantiated business processes in terms of the microstates and macrostates as defined in Section 3.2, could then be associated with typical management questions such as “Which task or tasks in the business process was (were) responsible for the extremely slow (fast) completion of this particular instance of the business process?” or “Which task or tasks in the business process was (were) responsible for the failure of the considered instantiated business process?” [9].
Hence, in case answering these management questions is unambiguous and straightforward, the entropy in the system is low as a particular macrostate (e.g., a high throughput time) is related to only one or a few microstates (e.g., the considered throughput time is due to the extremely high throughput time of task 1). In constrast, when no clear answer to these questions can be found, entropy increases: multiple microstates (e.g., prolonged execution time of tasks 1 and/or 2 and/or 3) are consistent in the observed and possibly problematic macrostate (e.g., the lengthy execution of the overall process).

This phenomenon seems to align well with the three basic entropy interpretations we cited in Section 2. First, business process analysis is more complex as the analyst has to consider the complete system as a whole, not being able to refine his diagnose to certain clearly isolated parts of the system. Second, lack of information and uncertainty are present when he tries to engage in remedial measures or optimizations. This is due to the fact that it is simply not known where potential problems are located, and the outcome or success of specific adaptations in the business process repository is uncertain as well. Finally, the tendency of particles to dissipate can be found in the fact that the “traces” of one problematic activity are spread over the complete considered system. Unless a consciously introduced separation is adopted, the information and outcome of all possible problem causing activities (tasks 1 till 5) interacts before being observed (in this case the measurements are aggregated). From an (enterprise) engineering viewpoint, it therefore becomes an appealing endeavor to control and reduce the entropy in these business systems.


Section 2 discussed how entropy can basically be interpreted as the degree of interaction or aggregation between the particles of a system. The aggregation of information units occurs in business practice as well [7]. Therefore, based on the formalisms we introduced in Section 3, some typical aggregations within the scope of one business process type are visually represented by the groupings in Figure 1 and summarized in Table 1. In the exemplary business process instantiation space, aggregations based related to task instance t_{1,1} are considered. While dimension 1 represents the most fine-grained level of information gathering, the other aggregation dimensions are applicable to a business context as well. For instance, in dimension 2 only information regarding certain major phases (milestones) in a production process are monitored and in dimension 3, information is only recorded at the level of a business process instance (e.g., for custom made cost-plus pricing). Dimension 4 could represent the situation in which one actor is taking care of all instantiations of one particular task type, and his wage is considered as the total cost of all task instances (aggregated), without any further information regarding the individual task instances. Finally, in dimension 5, information is recorded by means of a “counter” (e.g., electricity consumption) and in dimension 6 no information regarding individual business process or task instances is available (e.g., a mere cash flow statement).

As aggregation dimensions 2 till 5 represent aggregations of multiple information units in one way or another, each of these dimensions can be associated with a certain degree of entropy (here: multiplicity > 1). This is illustrated in Table 1 for the exemplary instantiation space of Figure 1 (i.e., column 3) and the more generalized case of any business process type (i.e., column 4). One can notice that generally, the amount of entropy increases systematically from aggregation dimension 1 till 6. In its original definition, the occurrence of entropy as formulated in the second law of thermodynamics should not necessarily be interpreted as having a “pejorative” connotation: the law just describes the inability of entropy to decrease in a closed system. In our application of the entropy concept to business processes and its interpretation in terms of a complexity control and uncertainty reduction perspective, a high degree of entropy is nevertheless unwanted for two main reasons. First, as entropy increases, possible problems at the micro-level become less observable. Suppose that in task instantiation t_{1,1} has been executed irregularly (e.g., an extremely long execution time). In case this information unit is combined with other information units as well, chances increase that the irregular behavior remains unnoticed as normal execution times of the other information units might "compensate" for the irregular behavior of t_{1,1}. Second, given the assumption that the business process owner has become aware of a potential problem at the micro-level, the traceability of the problem to the root cause (i.e., the relevant information unit) becomes more difficult. This is in contrast with, for instance, aggregation dimension 2, where the attention of the observer is directed towards t_{1,1} and t_{1,2}, whereas he still has to guess which of the two information units caused the irregular behavior. In aggregation dimension 6, the observer is even left with no real indication as to where precisely in the full instantiation space the problem can be situated, and uncertainty prevails.
### Table 1: six business process aggregation dimensions (based on [7]).

<table>
<thead>
<tr>
<th>Aggregation dimension</th>
<th>Short description</th>
<th>Entropy = Multiplicity ($\Omega$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Information is recorded at its most fine-grained level: for each individual</td>
<td>$1$</td>
</tr>
<tr>
<td></td>
<td>instantiation of each business process type, information per task instantiation on</td>
<td>$1$</td>
</tr>
<tr>
<td></td>
<td>an individual information object is recorded.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Information regarding two or more “information units” $k$ is aggregated within</td>
<td>$2$ # combined information units $k$</td>
</tr>
<tr>
<td></td>
<td>the scope of one single business process instance $j$.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Information is aggregated over all tasks $k$ per business process instantiation</td>
<td>$4$ # tasks $k$ in $BP_i$</td>
</tr>
<tr>
<td></td>
<td>$j$. A more general case of aggregation dimension 2.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Information among all instances $m$ of a particular task $k$ within business</td>
<td>$3$ # task instantiations $m$ of $t_k$</td>
</tr>
<tr>
<td></td>
<td>process type $BP_i$ is aggregated.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Information is aggregated according to a certain time $t$ which has elapsed.</td>
<td>$4$ # tasks $t_{k,m}$ executed at point in time $t$</td>
</tr>
<tr>
<td>6</td>
<td>Information regarding all (task) instances of the considered business process</td>
<td>$12$ depends on (a) # instantiations $j$ of $BP_i$, (b) # tasks $k$ and instantiations $m$ in $BP_i$, and (c) business process types $i$ in repository</td>
</tr>
<tr>
<td></td>
<td>type(s) becomes aggregated.</td>
<td></td>
</tr>
</tbody>
</table>

However, it is trivial to recognize that other aggregation dimensions, exhibiting an even higher amount of entropy, might be conceived in case multiple business process types are taken into account.

### 5. Towards a Prescriptive Business Process Design Theory

In Section 3, we provided a definition of how the entropy concept as provided by statistical thermodynamics can be applied to the run-time perspective of business processes. The relevance of this concept for studying everyday business situations was provided in Section 4, thereby illustrating the usefulness of employing justificatory knowledge for the design of lowly complex business processes. In a next step, this part of the existing knowledge base should be taken as an input for our design research activities, and more specifically for deriving and/or iterating on the formulation of the principles of form and function of the design science theory. In our case, this means that we should try to formulate a set of design guidelines, prescribing how a business process architect should design a business process repository if the aim is to minimize entropy. One possible avenue would be to look for opportunities to leverage mechanisms from traditional engineering science to control entropy to our application domain. Also, we could use our entropy conceptualization on the business process level (cf. Section 3.3) to test whether and to which extent (preliminary) guidelines help in coping with the occurring business process entropy.

While we want to stress that our current design theory is still in one of its design cycles and further additional cycles might be required to optimize them, an initial set of guidelines for business processes exhibiting a low degree of entropy has been presented in some working papers [10]. Such guidelines should enable business process managers not only to interpret their business processes in terms of the degree of present entropy, but to offer them specific means (i.e., “guidelines”) to design their business processes with a low degree of entropy as well. Current research results indicate that such guidelines are required on at least four different levels: the modular, business process, task, and functional domain dependent level. Due to page limitations, we were not able to present our full set of guidelines here. Therefore, for each of these levels, we will provide one exemplary guideline. However, this should allow us to demonstrate the relevance and feasibility of using the concept from the existing knowledge base in a (future) fully developed design theory, which is the main purpose of the present paper. For each of the presented guidelines here, we will provide (1) the formulation of the guideline, (2) an illustration of this guideline, and (3) the rationale of the guideline in terms of entropy.

#### 5.1. Guidelines at the general modular level

Guidelines at this level are generally applicable to most types of modular structures, in the sense that they
are valid for controlling and reducing entropy related issues in other application domains as well, outside the business process domain. For instance, the exemplary guideline we highlight here, has proven its relevance within the software engineering domain as well [27].

**Exemplary guideline:** Every instance of a module instantiating an embedded module, should provide traceability from this embedded “child” module to the overarching “parent” module.

**Illustration:** Applied to the business process application domain this means, for instance, that every task instantiated within a particular business process instance, should be uniquely traceable to this business process instance.

**Rationale:** Consider a situation in which a particular business process instance is found to be problematic (e.g., a too long throughput time). In case the previous guideline is not adhered to and no traceability regarding the embedded task instantiations of this business process instance is provided, many different microstates (i.e., a myriad of different task instance configurations) are consistent with one macrostate (i.e., the problematic business process instance), and entropy increases. For instance, in our exemplary business process instantiation space, a problem observed for business process instantiation BP_{1,1} (the macrostate) without task instance traceability could comply with the set of tasks sequences \{t_{1,1}, t_{2,1}, t_{3,1}, t_{5,1}\}, \{t_{1,2}, t_{2,2}, t_{4,1}, t_{5,2}\}, etcetera (i.e., several microstates).

### 5.2. Guidelines at the business process type level

Guidelines at this level are generally applicable to most types of business processes. More specifically, they are aimed towards providing instructions at business process designers for the identification of task types. Stated otherwise: when should a particular part of a business process be identified as a distinct “information unit”, and hence: task?

**Exemplary guideline:** If two (sequential) activities within a business process are executed by different actors, these activities should be designed into separate business processes.

**Illustration:** Consider the manual assembly of a Wheel, in which activity A is performed by person X, and activity B is performed by person Y. Parts A and B of the considered business process should be designed into separate tasks.

**Rationale:** Suppose that the previous guideline is not adhered to in the production of a Wheel, and only one task type T is identified (i.e., being the aggregation of activities A and B). In case a systematic problematic throughput time regarding task type T is observed, the business process designer does not know whether the problem is situated in activity A (carried out by X) or B (carried out by Y). However, two different actors might have different characteristics (e.g., tendency to work more quickly or a lower/higher wage and hence, cost) and therefore depict distinct information units. As different aggregation levels, each going through their own lifecycle.

**Rationale:** Suppose that a problem in the production process of one of the Part types (e.g., the Wheel) causes its production to have systematically prolonged throughput times. In case the previous guideline is not adhered to and the production of the Bike and its Parts are modeled in one single business process with mere task sequences, this problem might not be observed macroscopically. Indeed, as no separate throughput times at the level of the Wheel, Steer, or Frame task sequences as a whole are registered, the clues regarding a prolonged throughput time of the task sequence of the Wheel, might possibly become compensated by normal throughput times of the task sequences of the other Parts (leading to a smaller observed total deviation). The prolonged throughput time of the Wheel is however systematic and problematic, as it might incur suboptimal throughput times for other Product types (e.g., a Go-kart) as well, without being noticed. Therefore, multiple microstates (i.e., the throughput times for each of the Part types are “OK” or “not OK”) become consistent with one macrostate (i.e., no problems at the Bike level are reported), and entropy increases.

### 5.3. Guidelines at the task type level

Guidelines at this level are generally applicable to most types of tasks. More specifically, they are aimed towards providing instructions at business process designers for the identification of task types. Stated otherwise: when should a particular part of a business process be identified as a distinct “information unit”, and hence: task?

**Exemplary guideline:** For two (sequential) activities, activities A and B (carried out by Y). Parts A and B of the considered business process should be designed into separate tasks.

**Illustration:** Consider the manual assembly of a Wheel, in which activity A is performed by person X, and activity B is performed by person Y. Parts A and B of the considered business process should be designed into separate tasks.

**Rationale:** Suppose that the previous guideline is not adhered to in the production of a Wheel, and only one task type T is identified (i.e., being the aggregation of activities A and B). In case a systematic problematic throughput time regarding task type T is observed, the business process designer does not know whether the problem is situated in activity A (carried out by X) or B (carried out by Y). However, two different actors might have different characteristics (e.g., tendency to work more quickly or a lower/higher wage and hence, cost) and therefore depict distinct information units. As
a consequence, multiple microstates (i.e., actor A works too slow or actor B works too slow or a combination of both) are consistent with one macrostate (i.e., task type T is executed to slowly), and entropy increases.

5.4. Guidelines depending on the specific functional domain

The specific application of the guidelines within Sections 5.1, 5.2 and 5.3, might still heavily rely on the specific context, and sometimes even the subjective interpretation of the business process designer. Additionally, different organizational stakeholders might have different information requirements, depending on the specific situation and functional domain (e.g., logistics versus accounting). Therefore, domain dependent guidelines might be identified, incorporating relevant domain knowledge to adequately design business processes.

Exemplary guideline: Within the logistics domain, all activities performed by a meat handling company at the level of each individual packaged piece of meat should be separately traceable, due to safety and legal requirements. Therefore, they should be designed in a separate (logistic) business process type.

Illustration: This guideline is domain dependent, as for instance the financial reporting or (cost) accounting domains within the meat handling company would not necessarily need this information to be recorded. Rather, they are typically only interested in information (e.g., costs and selling price) at a higher bulk level (e.g., for each individual lot).

Rationale: Suppose that the previous guideline is not adhered to (only information at the lot level is registered) and a low quality of one particular piece of meat is observed by a customer. From a logistics and operational point of view, this means that multiple microstates (e.g., the possible problematic execution of the packaging activities for every individual meat piece within the respective lot) are consistent with one macrostate (i.e., a quality problem regarding one individual lot) and entropy increases. From an accounting or financial reporting viewpoint, this multiplicity (and hence, entropy) will not arise as only costs and selling price at the bulk level might be considered relevant. This implies that for different application domains, different aggregation levels (cf. Section 5.2) might be relevant. Therefore, some appropriate aggregation levels are to be determined by domain or industry experts.

A complete overview of all identified guidelines at each of the four proposed guidelines is out of scope for this paper. However, the interested reader can find a more elaborate discussion of additional guidelines in our working papers [10].


We started this paper by highlighting the importance of the existing knowledge base for design science research. As we have now presented our approach to analyzing and designing business processes in terms of the concept of entropy, it might be interesting to first of all reflect on the extent to which our proposed approach already constitutes a complete design theory. In this respect, further elaborating on the work of Walls et al. [38], six core components of a design theory have been described by Gregor and Jones [14]. First, the purpose and scope in this case are clear: the goal or “meta-requirement” of our design science research endeavor is to design a set of business processes (i.e., the artifact) in such way that they exhibit low complexity and uncertainty (operationalized by the entropy concept). Second, our constructs consist of modules, information objects, tasks (types and instances), business processes (types and instances) and the entropy measure (and hence, microstates and macrostates). Third, the principles of form and function, which should determine our artifact’s architecture, consist of the guidelines at the four levels (i.e., modular, business process, task and functional domain). As mentioned before, additional design cycles might be necessary to further optimize the guidelines. Also, at this point in time, no particular “sequence of order” for the considered guidelines is available. However, it is conceivable that a business process designer might not only be needing a set of guidelines to design his business processes, but also a particular order in which it is most efficient to apply them. Fourth, the different levels of guidelines equally indicate the degree of artifact mutability. More specifically, the guidelines at a general modular level have a rather broad applicability (i.e., also outside the business process domain) and thus it is anticipated that these guidelines can cope with changes of the artifact in terms of application domains, abstraction levels and functional domains. The business process and task level guidelines are deemed to be relevant for several kinds of processes (e.g., industrial processes, services, etcetera). Therefore, they can cope with changes in the artifact regarding different functional domains. Finally, the guidelines which are dependent on a particular functional domain, have the most narrow applicability. As a consequence, they should only be able to cope with a small range of changes of the artifact, i.e., those within their functional domain. For each different
functional domain, the set of applicable guidelines at this level, will differ. Fifth, the testable propositions in our theory are that the application of the guidelines should result in business processes exhibiting less complexity or uncertainty (compared to when the guidelines would not be applied). Our propositions are mainly proven through deductive logic, i.e., we show that, similar to a reductio ad absurdum, not adhering to the guidelines results in entropy increase based on logical reasoning. Additionally, we already performed one theoretical case simulation in which the entropy concepts and guidelines were applied to an imaginary case organization [8], showing how the reasoning can be applied meaningfully in practice. While this might serve as a first step towards expository instantiations, extra real-life cases should ideally be performed in the future to further ground the feasibility of our claims. These ways of evaluation are also recognized in the design science community and applied in other research projects [11]. We further extensively discussed that our justificatory knowledge, as a sixth component, is situated in thermodynamics. More specifically, we leverage the entropy concept, in this case coming from natural science and guiding our design process. It therefore “provides an explanation of why an artifact is constructed as it is and why it works” [14:328].

To sum up, we conclude that we can clearly position most of the design theory core components of [14] within our approach. Regarding the additional design theory components mentioned by Gregor and Jones [14], we note that supplementary instantiated artifacts (resulting from our method) would be preferable to show the viability and practicality of our design theory.

Several other issues should be mentioned as well, thereby indicating some of the limitations of this paper as well as future research opportunities. For instance, we assumed that minimal entropy is the desired situation for a business process repository in any situation of every company. However, entropy control requires additional structure and fragmentation of the business processes and their tasks. This demands extra resources, both in terms of design efforts, as well as registration at this more fine-grained level during runtime. Therefore, in future case studies, it should be investigated whether the benefits by realizing our theoretical meta-requirement of minimal entropy offsets the additional costs they require.

Next, while we claimed in Section 1 that prescriptive business process design guidelines in literature are rare, a few of them do exist. These guidelines might typically concentrate on other optimization criteria (i.e., meta-requirements), such as adaptability or evolvability [30]. Therefore, it might be interesting to investigate whether these guidelines are consistent, complementary or conflicting with one other. Indeed, some of our guidelines for controlling entropy are as such not new, as very similar guidelines were already proposed in the past [27,30]. However, our contributions are situated in the fact that we derived them from a new theoretical perspective, i.e., thermodynamics entropy. Other guidelines might be conflicting with other existing approaches. In such cases, the appropriate trade-off decisions should be made. This relates to the issue we highlighted in Section 5.4, namely that some of our proposed guidelines might be dependent on the specific functional domain. Most organizations would typically be engaged with multiple of these functional domains (e.g., logistics, accounting, production, etcetera) simultaneously. Therefore, integration issues regarding the guidelines of these multiple functional domains should be addressed in future research as well.

Finally, we noticed in Section 3.1 that widely adopted standards for visually representing run-time behavior models of business processes and their tasks seem to be lacking. Most standards merely offer means to visualize the design-time of business processes. However, our analysis shows that much complexity of business processes is situated at their run-time level and reflection on the possible aggregation dimensions within a business process space can only take place when this perspective is taken into account. Therefore, we hope to stimulate the business process design community to extend current business process modeling standards for this purpose.

7. Conclusion

Business processes and the data they generate are often complex, causing uncertainty for managers who have to take decisions based on this information. In this paper, we suggested the concept of entropy from thermodynamics as a useful concept from the knowledge base, acting as justificatory knowledge for governing DSR efforts in this context. Therefore, the main contribution of this paper is situated in providing, for the first time in an integrated whole, how entropy might be used as a relevant concept for business process design. We were able to link a business process entropy definition, its implications in practice, as well as an outline of how a resulting design science theory aiming to control business process complexity, might look like. Therefore, we conclude that entropy constitutes an interesting avenue for formulating a design science theory in this context. Future research efforts should be directed towards further iterating the design guidelines, as well as performing real-life case studies based on them.
8. Acknowledgments

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

9. References