CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Domain design principles for managing complexity in conceptual modeling
Authors	Kaul, Mala;Storey, Veda C.;Woo, Carson
Publication date	2017
Original Citation	Kaul, M., Storey, V., Woo, C. 2017. 'Domain Design Principles for Managing Complexity in Conceptual Modeling'. In: Maedche, A., vom Brocke, J., Hevner, A. (eds.) Designing the Digital Transformation: DESRIST 2017 Research in Progress Proceedings of the 12th International Conference on Design Science Research in Information Systems and Technology. Karlsruhe, Germany. 30 May - 1 Jun. Karslruhe: Karlsruher Institut für Technologie (KIT), pp. 1-10
Type of publication	Conference item
Link to publisher's version	https://publikationen.bibliothek.kit.edu/1000069452, http://desrist2017.kit.edu/
Rights	©2017, The Author(s). This document is licensed under the Creative Commons Attribution – Share Alike 4.0 International License (CC BY-SA 4.0): https://creativecommons.org/licenses/by-sa/4.0/deed.en - https://creativecommons.org/licenses/by-sa/4.0/deed.en
Download date	2026-11-16 11:53:41
Item downloaded from	https://hdl.handle.net/10468/4436

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Domain Design Principles for Managing Complexity in Conceptual Modeling

Research in Progress

Mala Kaul[1], Veda C. Storey[2], and Carson Woo[3]

[1]Department of Information Systems, College of Business, University of Nevada, University of Nevada, Reno, 1664 N. Virginia Street, Reno, NV 89557, USA.
mkaul@unr.edu
[2]Department of Computer Information Systems, J. Mack Robinson College of Business Georgia State University, Box 4015 Atlanta, GA 30302, USA.
vstorey@gsu.edu
[3]Sauder School of Business, University of British Columbia, 2053 Main Mall Vancouver, BC V6T 1Z2, Canada.
carson.woo@sauder.ubc.ca

Abstract. Complexity is a problem that can be found in many aspects of research that deals with design. In particular, complexity is found in various business processes that must be modeled and represented in a meaningful way. One of the ways to address complexity is by using decomposition, for which a number of decomposition principles have been proposed. However, there are two domain specific areas in which these principles are lacking: the scope and the context of the problem. This research addresses this problem by deriving two new principles for managing complexity, and evaluates the proposed principles through an example case to illustrate their potential use.

Keywords: Complexity · Design · Scope · Context, Conceptual modeling · Business process management · Decomposition principles · Domain design principles

1 Introduction

In today's business environment, designers and developers must cope with the complexity that is inherent in the processes they model, which requires the application of effective design rules at the conceptual modeling phase of systems analysis. Although there have been ideas and principles proposed to deal with complexity, there are still some aspects that remain challenging. The objectives of this research are to: define complexity with respect to conceptual modeling for business process management; examine existing design principles to identify where these design principles are lacking; and propose new design principles and demonstrate their potential effectiveness.

Motivated by real world problems in business process management, this research builds upon existing design principles from decomposition rules. Thus, this research

contributes to decomposition by identifying two new principles, scope and context. These principles are derived based on notions of decomposition from Simon's [29] inner/outer environment and Alter's [1] work system framework. Application of the proposed scope and context principles is intended to address the issue of complexity in conceptual modeling.

The rest of the paper proceeds as follows. It first defines complexity in terms of prior work on conceptual modeling and design science research. Then, existing principles are analyzed, and two new ones, scope and context, are derived. These principles are then applied to an existing problem to show their potential usefulness and value for teaching conceptual modeling, as well as for professional designers. We conclude with a summary and suggestions for future work.

2 Related Research

2.1 Complexity

The concept of complexity is inherently problematic since it arises from the interaction of a number of closely linked (sub)systems, which may have emergent properties which are distinctly different from the sum of the properties of the individual subsystems [29]. Within the context of information systems, complexity is a problem that is found in many different types of problems that deal with design [6, 11, 36]. Although a number of different methods have been proposed to explain complexity, such as, complexity theory [13], chaos theory [7], complex adaptive systems theory [11], dissipative structure [25] and living system theory [15], or to compute complexity, e.g. system dynamics [30], agent based modeling [8], none of these approaches directly provides a method to resolve information systems design complexity.

According to [22], "Complexity is the way the world is; systems thinking and rigorous design methodologies are the key to success". Design science research has a tradition of being especially suited for solving complex or "wicked" problems [28] while being engaged in knowledge production [2. 31]. Although the notion of complexity is well recognized in the design science research literature, it is mostly submerged in the intractable nature of the problem and the complexity of the information systems environment, with greater focus on resolving these intractable problems through the generation of novel artifacts [12, 16, 19].

2.2 Conceptual Modeling

One way to address complexity in designing novel solutions is simplification. [23], suggest that to capture the complexity of the problem so that the artifact appropriately addresses the problem requirements, it may be helpful to atomize or decompose the problem conceptually. [17] suggests that decomposability is a way to uncover simple answers to complex problems and lead to elegant, simple designs. Therefore, having a good understanding of the problem is key to a successful design. Conceptual models help provide a good representation of the real world problem scenarios. A good con-

ceptual model provides a means of communication and common understanding between system analysts, designers as well as the system's users [18]. Thus, conceptual models are the elements of a design science process which aims at solving real world problem situations. Not only do they help in representing the real world, they also help provide this representation at an appropriate level for a given situation; i.e. the level of detail provides adequate information, but is abstract enough to make sense of, without redundancy or complexity. This requires: 1) the right level of abstraction and 2) a syntactical diagram [21].

2.3 Existing Decomposition Principles

The Bunge-Wand-Weber (BWW) models [32-35] that originally drew on Bunge's ontology [3, 4], provides a formal approach to modeling information systems for practical application. [34] good decomposition model provides a way to address the complexity of a real world system by providing a method to represent it in such a way that users can achieve a better understanding of a system. According to Wand & Weber, there is one high-level object that should be decomposed into a set of sub-systems based upon some rules that guide as to what should/should not be included and also to what degree the system should be decomposed (and when to terminate this decomposition). Based upon the evaluation of the BWW model, five principles have been defined for good decomposition, thereby leading to effective management of complexity in systems design. These good decomposition principles [34] can be summarized as follows:

Minimality – the characteristic of a system with state variables where each sub-system, at every level of the decomposition, has no redundant state variables.

Determinism – the characteristic of a system where, for each of the subsystems with a certain level of the decomposition, an event is either an external event or a well-defined internal event.

Losslessness – the characteristic of a system where the decomposition into subsystems, results in the preservation of each hereditary or emergent state variable.

Minimal coupling – this characteristic of a system results when any environmental change has the least amount of impact on any of the subsystems in the decomposition.

Strong cohesion – the characteristic of the output variables being cohesive if they depend on a common input variable and if any additions to the output variables do not necessitate additional input variables.

3 Domain Design Principles: Scope and Cotext

3.1 Limitations of Existing Decomposition Principles

The BWW model has been richly examined, evaluated and applied [5, 10, 26, 27]. Although the BWW ontology is well established, work is still required to understand how processes and process-oriented systems can be decomposed [27] with a lack of generally accepted criteria for assessing the goodness of decomposed process models [14]. Practically, decomposition and conceptual modeling are well established methods

employed in systems analysis, for mapping business problems and processes. Conceptual modeling (e.g., drawing a data flow diagram) can be considered as decomposing a system.

In this context, we can use the minimality, determinism, losslessness, and minimum coupling principles. Nevertheless, strong cohesion might not be necessary at the top level diagram as this principle can potentially result in too many details at that level, but should be enforced in lower level diagrams. For example, based on [20], people have been using 7 ± 2 processes in level 0 data flow diagrams (DFDs). Enforcing strong cohesion at level 0 might result in significantly more processes at level 0 DFD, which is not recommended.

The five decomposition principles assume the existence of a system to be decomposed. In reality, during conceptual modeling, the scope of the system is not necessarily known. This is observed, for example, in problems that occur with student modeling exercises. For example, in analyzing a warehouse, assume that many customers complain about the long delay in getting their items. If the root of the problem is delivery, then drivers should be included in the analysis scope; if the root cause is locating the items, then drivers will not be included in the analysis scope, but the scope should include how items are being stored. Similarly, after identifying the root problem, the solution to the problem could be different. For example, if the problem is delivery, but drivers are already doing their best, the solution might be to provide customers a better estimation of arrival date/time. Thus, the initial customer interactions must be included in the scope of the analysis. Scope identification is an important practical issue in systems analysis [1, pp. 33-34], but is not addressed in existing decomposition principles.

Similarly, the five decomposition principles do not consider a major objective of conceptual modeling, which is for the purpose of understanding and communication [21]. The understanding and communication can be among management, users, analysts, and developers. Using the warehouse example as above, when describing the steps of the warehouse operations, two students capture different steps:

1. Check customer authority, check warehouse items, check items, arrange transportation, delivery, arrange pick-up, and process document.
2. Check authority, pass request, process request, arrange transportation, and receive order for transport.

Common challenges are: i. How many steps, and ii. What steps, are needed? The above examples are 6-7 steps, yet they convey a different picture of the warehouse operation. Knowing what to represent for the purpose of understanding and communication, is a practical issue, but not addressed in the existing decomposition principles.

3.2 The Two Domain Principles

There are two domain specific limitations in the deployment of these five principles for achieving good decomposition. This leads to the proposal of two new principles to supplement the existing five principles, which we call the scope and context principles.

None of the five decomposition principles can guide scope determination as shown in the above examples. Since scoping is very much an issue in systems analysis, we utilize one of them to derive the scope principle. Based on Alter's [1] work system framework, a work system should be the smallest unit in which the problems or opportunities occur, we propose:

Scope principle: A conceptual model should be the smallest system, in which the problems or opportunities occur.

Similarly, the five decomposition principles do not guide the identification of subsystems (e.g., a process in a data flow diagram) for the purpose of understanding and communication. To resolve this issue, we turn to Simon's [29] decomposability in the context of the inner and outer environment, where his strategy is to hide the details of the inner environment so one can study the outer environment without having to know much about the inner environment. Considering the outer environment as a purpose of understanding and communication, Simon's strategy will be applicable here with the inner environment containing the subsystems. This implies that subsystems should be designed so that we do not need to know much about them, but they are sufficient for us to study the outer environment.

Using Simon's inner/outer environments terminology, the determinism principle captures the inner environment of a decomposed subsystem, but does not provide guidance to the outer environment, which is the interaction of the subsystems. Interactions of subsystems are guided by the minimum coupling principle. A data flow diagram (DFD), for example, with straight-forward linear steps of transforming data (i.e., each process has only one data flowing in and one data flowing out) can fulfill minimum coupling, but provides no guidance as to what constitutes a good set of processes to be used at the top level diagram. Thus, we propose the following principle.

Context principle: A conceptual model should contain the smallest number of subsystems (with a minimal knowledge of each subsystem) and their interactions so that it is sufficient to be used for the purpose of understanding and communication.

4 Application of the Domain Principles

An important part of design science research is to demonstrate the applicability and utility of the model [9, 12, 24]. To assess the validity and utility of the two proposed principles, we apply the principles to an example case that was given as a student assignment in a large North West university as a proof-of-concept. Our findings indicate that these principles can delineate diagrams that communicate more with less. See Appendix 1 for the assignment and Appendix 2 for 2 sample DFDs. For the purpose of this discussion, the DFD processes are listed below, because they are considered subsystems in the DFDs.

Example 1 DFD processes:	**Example 2 DFD processes:**
Check stock availability	Assist customer
Process payment	Place shoes on hold

Update inventory	Process sales transaction
Process return	*Sign up member*
Produce restock request	Create receipt
Submit request	Return shoes
Generate sales report	Restock inventory

Both example solutions conform to the 5 decomposition principles. Nothing in the processes is redundant, thus conforming to the minimality principle. The solutions conform to the determinism principle because they have a deterministic reaction to each data flowing in each process. The suggestion of a strong verb to name each DFD process usually addresses this principle. Losslessness is already taken care in the DFD rule: each child diagram should have the same input and output data flow as the parent process. Similarly, weak coupling or minimum coupling is already taken care in the DFD rule that all data flows should be named. This minimizes the unnecessary interactions among processes. Since each process in both diagrams only has one input data flow or one output data flow except when the process accesses a data store, this conforms to the strong cohesion principles because all output depends on all input data flows. Nevertheless, the above two examples violate at least one of the two new principles.

In both examples, the DFDs fulfill the rule of 7 ± 2 processes at the top level DFD. Clearly, they convey two different stories although they are the same assignment. Example 1 did not include "signing up member", while Example 2 did not include "generating sales report". One can argue that "signing up member" occurs when processing payment and thus is part of "process payment" in Example 1. It can also be argued that "generating sales report" is beyond the scope of this DFD, thus not included in Example 2. Hence, new principles are needed. According to the scope principle, if the problem to be addressed in this DFD is to increase sales, then "generate sales report" should be within the scope of this DFD. However, if the problem is to improve customer shopping experience, then "generate sales report" can be outside the scope of this DFD. Thus, pinpointing the problem or opportunity is important in determining the scope of DFDs.

According to the context principle, we should not need to know the details of a process, to understand and communicate DFD at that level. If signing up a member is important for improving sales or providing a better customer experience, then this should not be hidden inside the process payment process in the Example 1 DFD. On the other hand, submit request receives its data flow from produce restock request. It does not seem necessary to know whether a restock request is being submitted at this top level DFD as we assume that someone analyzing restocking requirements, will do something about it. Therefore, according to the context principle, Example 1 DFD should combine restock request and submit request into one process. Similarly, create receipt should be part of process sales transaction in Example 2. More importantly, according to the context principle, Example 2 conveyed a better understanding of how the shoe store works as it provided signing up member and placing a hold on shoes services (not seen in Example 1 DFD). Interestingly, Example 2 received a lower grade from the grader than Example 1. This shows that traditional rules for drawing DFDs did not consider the main purpose of conceptual modeling, which is for understanding and communication.

5 Conclusion

Conceptual modeling has always been difficult because of the challenges in modeling the real world in a manner that can be understood. Although a number of techniques and approaches have been proposed for conceptual modeling, challenges remain for how to develop "good" and accurate models that facilitate understanding and communication. A well-recognized challenge in doing so is dealing with complexity. This research has analyzed existing decomposition approaches for complexity and proposed two new principles: scope and context, both of which are derived from the literature on design science. The utility of the principles was demonstrated through an application to a DFD exercise. Further research will require empirical studies (e.g. as those carried out by [5]) that use the principles and assess their value.

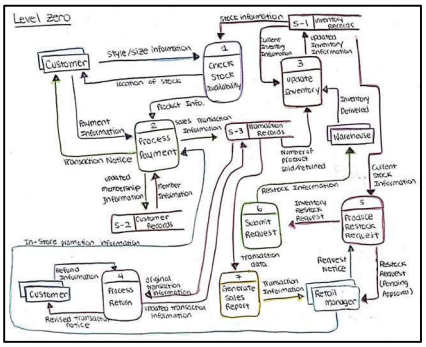
Appendix 1: The Shoe Store Case for the DFD Assignment.

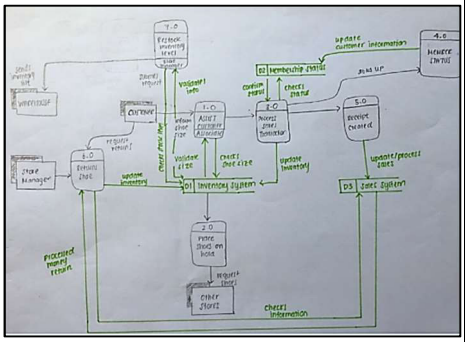
Kits Footwear is a Canadian shoe retailer based in Vancouver. The company has seven stores and two warehouses across Canada. Each store is run by a store manager; the retail manager supervises all the seven store managers. Depending on the size of each store, between 4-6 sales associates and 2-3 cashiers are employed by the store manager. When a customer enters a Kits Footwear store, the sales associates welcome the customers, explain the existing promotions and discounts, and answer their questions. When a customer decides to try a pair of shoes, the sales associate asks the customer his/her shoe size and check the store stock availability. If the shoes are in stock, the sales associate brings them to the customer to try on. If not, the sales associate suggests the customer pick other styles, or if the customer prefers, the sales associate contacts other Kits Footwear branches to check for availability. If another branch has the right size in stock, the sales associate places a hold request and asks the customer to visit the other branch within 24 hours. After the customer decides to purchase a pair of shoes, the sales associate escorts him/her to the cashier. The cashier asks if the customer has any coupons or Kits Footwear membership card. Seniors and students with proper identification are also eligible for extra discounts. If the customer would like to apply for a membership card or renew it, the cashier proceeds with the request. Finally, the cashier collects the payment in cash or by credit/debit card, files the sales receipt, and gives a copy of it to the customer.

Kits Footwear has the return policy for unused items within 30 days of purchase. When a customer visits the cashier to return an item, the cashier should call the store manager, who has the authority to process the return requests. The store manager asks for the return reasons, and if the product is eligible for return, confirms the refund. The cashier completes the refund transaction, requests the customer to sign the return form, and gives a copy of the new receipt to the customer. The store manager is responsible for managing the business performance of the store. The retail manager meets weekly with all the seven store managers to analyze sales in the last week and set new sales goals. If the marketing department has created new promotions or discounts, the retails manager passes on this information to the store managers. Each store manager then uses

the sales goals to schedule the store personnel and define daily goals for the sales associates. Finally, the store manager should maintain an appropriate stock level of products in the store and reorder products if necessary. Weekly, she provides a list of required items and submits the list to the retail manager. After the retail manager confirms the reorder, the store manager sends a request to the closer warehouse. When the ordered items arrive, the sales associates refill the store stock.

Appendix 2: Two sample DFDs .





DFD Solution 1	DFD Solution 2

References

1. Alter, S. 2006. The Work System Method: Connecting People, Processes, and IT for Business Results. Larkspur, CA: Work System Press.
2. Baskerville, R.L., Kaul, M., and Storey, V.C. 2015. "Genres of Inquiry in Design-Science Research: Justification and Evaluation of Knowledge Production," Mis Quarterly (39:3), pp. 541-564.
3. Bunge, M.A. 1977. "Treatise on Basic Philosophy: Ontology I: The Furniture of the World. Dordrecht, Holland: D." Reidel Publishing Company.
4. Bunge, M.A. 1979. "Treatise on Basic Philosophy: Ontology Ii: A World of Systems. Dordrecht, Holland: D." Reidel Publishing Company.
5. Burton-Jones, A., and Meso, P.N. 2006. "Conceptualizing Systems for Understanding: An Empirical Test of Decomposition Principles in Object-Oriented Analysis," Information Systems Research (17:1), pp. 38-60.
6. Dakin, G. 1993. "Shaping the Future: Business Design through Information Technology," Journal of the Operational Research Society (43:9), pp. 1249-1250.
7. Dervin, B. 1999. "Chaos, Order and Sense-Making: A Proposed Theory for Information Design," Information design), pp. 35-57.
8. Druckenmiller, D.A., and Acar, W. 2009. "An Agent-Based Collaborative Approach to Graphing Causal Maps for Situation Formulation," Journal of the Association for Information Systems (10:3), p. 221.
9. Gill, T.G., and Hevner, A.R. 2013. "A Fitness-Utility Model for Design Science Research," ACM Transactions on Management Information Systems (TMIS) (4:2), p. 5.

10. Green, P., and Rosemann, M. 2000. "Integrated Process Modeling: An Ontological Evaluation," Information systems (25:2), pp. 73-87.
11. Hanseth, O., and Lyytinen, K. 2010. "Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet," Journal of Information Technology (25:1), pp. 1-19.
12. Hevner, A.R., March, S.T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," MIS Quarterly (28:1), Mar 2004, pp. 75-105.
13. Jacucci, E., Hanseth, O., Lyytinen, K., Benbya, H., and McKelvey, B. 2006. "Toward a Complexity Theory of Information Systems Development," Information Technology & People (19:1), pp. 12-34.
14. Johannsen, F., and Leist, S. 2012. "Wand and Weber's Decomposition Model in the Context of Business Process Modeling," Business & Information Systems Engineering (4:5), p. 271.
15. Knoll, K., and Jarvenpaa, S.L. 1994. "Information Technology Alignment or "Fit" in Highly Turbulent Environments: The Concept of Flexibility," Proceedings of the 1994 computer personnel research conference on Reinventing IS: managing information technology in changing organizations: managing information technology in changing organizations: ACM, pp. 1-14.
16. Kuechler, B., and Vaishnavi, V. 2008. "On Theory Development in Design Science Research: Anatomy of a Research Project," European Journal of Information Systems (17:5), pp. 489-504.
17. Lewis, K. 2012. "Making Sense of Elegant Complexity in Design," Journal of Mechanical Design (134:12), p. 120801.
18. Maes, A., and Poels, G. 2007. "Evaluating Quality of Conceptual Modelling Scripts Based on User Perceptions," Data & Knowledge Engineering (63:3), pp. 701-724.
19. March, S.T., and Smith, G.F. 1995. "Design and Natural Science Research on Information Technology," Decision support systems (15:4), pp. 251-266.
20. Miller, G.A. 1956. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," Psychological review (63:2), p. 81.
21. Mylopoulos, J. 1992. "Conceptual Modeling and Telos. Conceptual Modeling, Databases, and Case. An Integrated View of Information Systems. P. Loucopoulos and R. Zicari." John Wiley & Sons.
22. Norman, D.A. 2010. Living with Complexity. MIT press.
23. Peffers, K., Tuunanen, T., Rothenberger, M.A., and Chatterjee, S. 2007. "A Design Science Research Methodology for Information Systems Research," Journal of Management Information Systems (24:3), pp. 45-77.
24. Pries-Heje, J., Baskerville, R., and Venable, J. 2008. "Strategies for Design Science Research Evaluation," ECIS 2008 proceedings), pp. 1-12.
25. Prigogine, I. 1967. "Introduction to Thermodynamics of Irreversible Processes," New York: Interscience, 1967, 3rd ed.).
26. Recker, J.C., Indulska, M., Rosemann, M., and Green, P. 2005. "Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of Bpmn,").
27. Recker, J.C., Rosemann, M., Indulska, M., and Green, P. 2009. "Business Process Modeling-a Comparative Analysis," Journal of the Association for Information Systems (10:4), p. 1.
28. Rittel, H.W., and Webber, M.M. 1973. "Dilemmas in a General Theory of Planning," Policy sciences (4:2), pp. 155-169.
29. Simon, H.A. 1969. The Sciences of the Artificial. Cambridge, MA: MIT Press.
30. Sterman, J. 2000. Business Dynamics: Systems Thinking and Modeling for a Complex World. Boston: Irwin/McGraw-Hill.

31. Van Aken, J.E. 2005. "Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management," British journal of management (16:1), pp. 19-36.
32. Wand, Y., and Weber, R. 1989. "An Ontological Evaluation of Systems Analysis and Design Methods,").
33. Wand, Y., and Weber, R. 1990. "An Ontological Model of an Information System," IEEE transactions on software engineering (16:11), pp. 1282-1292.
34. Wand, Y., and Weber, R. 1993. "On the Ontological Expressiveness of Information Systems Analysis and Design Grammars," Information Systems Journal (3:4), pp. 217-237.
35. Wand, Y., and Weber, R. 1995. "On the Deep Structure of Information Systems," Information Systems Journal (5:3), pp. 203-223.
36. Zachman, J.A. 1987. "A Framework for Information Systems Architecture," IBM systems journal (26:3), pp. 276-292.