Proceedings of the Design Society, Volume 5: ICED25
https://doi.org/10.1017/pds.2025.10041

A study on representation of services using the SAPPHIRE model of causality

Kausik Bhattacharya,✉ and Amaresh Chakrabarti

Indian Institute of Science Bangalore, India

✉ kausikb@iisc.ac.in

ABSTRACT: With increasing servitization, manufacturers are transitioning from solely selling products to integrated products and services. While the SAPPhIRE model of causality effectively represents technical systems and aids in product design activities like analysis, synthesis, and assessing design novelty, few studies have explored its extension to services. Previous research extended SAPPhIRE constructs to capture causality in Service Systems. This research compares SAPPhIRE models for services with Object Process Methodology (OPM), a benchmark for systems modeling. Results show that SAPPhIRE not only captures details represented by OPM but also provides additional useful information for service representation. The causal description of products and services using SAPPhIRE helps understand and improve existing service systems, trace root causes of issues, and foster creative ideation for new designs.

KEYWORDS: product-service systems (PSS), design methodology, ontologies, SAPPhIRE, product modelling / models

1. Introduction

Design transforms existing situations into desirable ones, resulting in tangible products or valuable services. Over time, manufacturers have increasingly combined products and services to gain market share and remain competitive (Shihundla, 2019). Product and Service Systems (PSS) offer superior user value and economic benefits, while promoting sustainable development (Geum & Park, 2011). A PSS includes products, services, infrastructure, processes, and a network of stakeholders (Coreynen et al., 2020; Gaiardelli et al., 2021). PSS value offerings are categorized as Product-Oriented, Use-Oriented, and Result-Oriented (Tukker, 2004). Industry 4.0's digitalization has led to data-driven smart PSS (Antonova, 2018; Carsten et al., 2018; Tomiyama, 2001). Six characteristics of products and services highlight PSS design drivers, with product design driven by functions and service design by user needs (Moreno Grandas et al., 2015).

Representation provides valuable information to designers and significantly aids idea generation (McKoy et al., 2001), making it a key area of research for services and PSS (Vasantha, 2012). As OEMs focus more on services around products, we aim to cover both under a common ontology, enhancing consistency, integration, and collaboration. This paper examines the SAPPhIRE model's (Chakrabarti et al., 2005) potential to represent services, building on its proven benefits in product design, and setting the stage for its application in service design.

2. Literature review

2.1. Design representation of service systems

A detailed study on ontologies in product design, service design, and PSS proposed a PSS ontology using ten root concepts: user needs, stakeholders, PSS-Design, product life cycle, use phase, infrastructure, business elements, business models, supply network, and benefits (Vasantha et al., 2011). Service modelling, like engineering modelling, describes service constituents: Service Provider, Service Content, Service Channel, and State Change, along with their interactions and graphical representation (Shimomura & Tomiyama, 2002). Molecular modelling for services includes Service Element, Product Element, and the bond between them (Shostack, 1982). Representation often considers multiple views or dimensions, such as stakeholder, service, systems, and operational views (Fakhfakh et al., 2021), or higher dimensions like product, customer, value, actor, service, business model, interaction context, and time (Kim, 2020). These works capture the complexities of modern service systems involving people, processes, and products (or technology). Service designers use service blueprints to graphically represent services, including User Actions, Onstage/Visible Contact Employee Actions, Backstage/Invisible Contact Employee Actions, Support Processes, and Physical Evidence (Bitner et al., 2008). Researchers extended service blueprints to smart services (Li & Lu, 2021) and PSS (Chuang et al., 2022). For example, a new PSS blueprint includes Product Area, Service Area, and Supporting Area, showing product usage throughout its lifecycle and the service flow from management to user (Geum & Park, 2011). Another PSS blueprint has four layers: user action, product area, PS Platform area, and Support process, each with Name, Attribute, and Operation information (Moon et al., 2013). A holistic PSS blueprint focuses on product functions instead of physical representation and includes Product Functions (Hardware and Software), User Activities, Provider Activities, and Supporting infrastructure (Herzberger et al., 2013).

The review shows that due to the similarity between PSS and services, the service blueprint concept is extended to PSS to illustrate connections between products, services, and actors. However, blueprints do not describe causality among constituents or connect to user state changes. They also do not explain the rationale for integrating products and services, including causal relationships. Understanding causality in services or product-service combinations is crucial as it explains their functionality and value addition by changing the user's situation or state. Therefore, a causal model can be useful for analyzing and synthesizing Services or PSS.

2.2. Extending SAPPhIRE model to services

The SAPPhIRE model represents the causality of natural and engineered systems, detailing the interaction of physical entities with their surroundings (Venkataraman et al., 2009). It can link multiple models to describe complex systems and is useful for design ideation, synthesis, and measuring creativity. Its powerful nature extends its application to two situations: (a) Covering the entire product lifecycle (Design, Manufacturing, Operations/In-service) by introducing a new construct, 'Observation,' to measure state changes and decide on corrective or preventive actions. The 'Action' construct becomes optional, depending on 'State Change' observations (McSorley et al., 2014). (b) Modifying SAPPhIRE constructs to represent services, capturing causality among service constituents and assessing novelty (Borgianni et al., 2012).

The first research uses the SAPPhIRE model to describe product behaviour over its lifecycle, recognizing the need for follow-up actions based on state changes during the in-service phase. However, it does not integrate service offerings to enhance user value. The second research modifies SAPPhIRE constructs to explain service causality but does not detail the basic building blocks of service delivery processes or the conditions enabling service delivery. It also does not address how to model complex situations (or state changes) in services using SAPPhIRE. Importantly, neither research explains Product and Service integration rules or a schema using a causal ontology like SAPPhIRE. Therefore, previous research (Bhattacharya & Chakrabarti, 2023) investigates the causal relationships among service components in Service Systems, using the SAPPhIRE model of causality (Chakrabarti et al., 2005) with its seven layers of abstraction. This research extends SAPPhIRE constructs to include both services and products. However, the SAPPhIRE model has not been previously applied in service design, leaving its ability and effectiveness to represent services unknown.

Figure 1 shows the SAPPhIRE model, and definition of SAPPhIRE constructs:

Parts: In a product, parts are the constituent entities and their immediate interfaces. In service, these

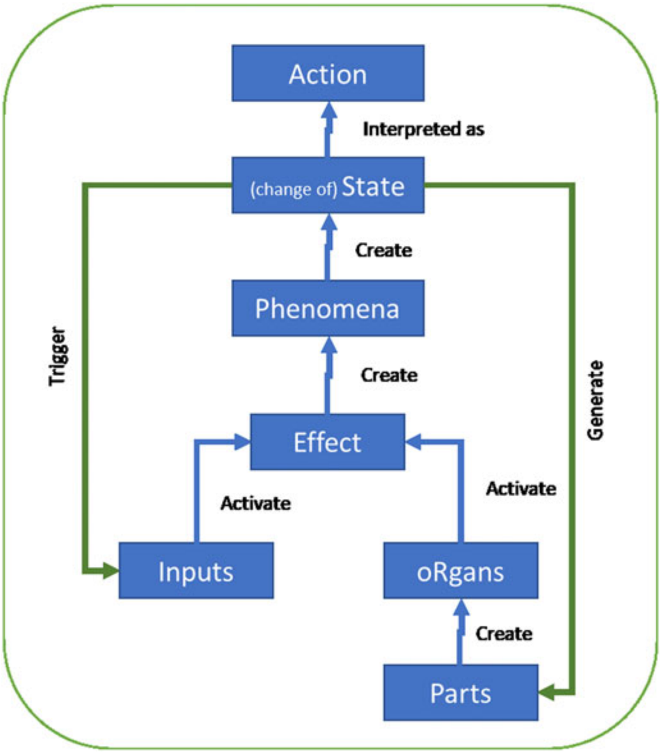


Figure 1. SAPPhIRE Model of Causality for Product and Service (Bhattacharya & Chakrabarti, 2023)

are the service actors and infrastructure carrying out the customer interactions.
oRgan: These elements act like enabling conditions without which the interactions will not occur. In a product, these are the attributes of the product Parts or conditions for the interaction of the Parts. In service, these are the attributes of the service Parts or conditions for the interaction of the Parts.
Input: These are the external elements (material, energy or information) necessary for the interactions.
Effect: Applicable Rules or Laws governing the interactions. In a product, these are the laws of nature (physical or biological), in service, these are the applicable rules or policies governing the service.
Phenomenon: These are the interactions in a Product or Services. These interactions create the transfer or transformation of material, energy or information. In a product, this is the physical interaction of an entity with its surrounding. In service, these are the interactions between customers and service actors.
State Change: Resulting Change of State caused by the Phenomena
Action: Interpretation of the State Changes

3. Research question and research method

3.1. Research opportunity and research question:

Designers often use solutions from analogous designs to innovate (Christensen & Schunn, 2007). Representation and modalities are crucial in analogies (Linsey et al., 2007), computational analogical reasoning (Hill et al., 2019), and data-driven design by analogy (Jiang et al., 2022). Design by Analogy is effective for generating novel ideas (Fu et al., 2015) and is applied in service design (Moreno et al., 2014) and product-service systems (Moreno et al., 2015). The SAPPhIRE model (Chakrabarti et al., 2005) is proven to enable design-by-analogy method by synthesizing and analysing design concepts and assessing novel product configurations (Venkataraman et al., 2009; Sarkar & Chakrabarti, 2011).

However, its application in service design remains unexplored. Evaluating the SAPPhIRE model's ability to represent services could pave the way for its use in design of services and product-service systems. This study aims to investigate whether the SAPPhIRE model can capture and represent all necessary details of a service. Hence, the main research objective is to find out, "*Can a SAPPhIRE model capture and represent all the necessary details of a Service?*"

3.2. Research method

To find an answer, the SAPPhIRE representations of three services are compared with the Object Process Methodology (Dori, 2011). This comparison seeks to answer: (a) Can the SAPPhIRE model represent the service information in the benchmark model (overlapping information)? and (b) Does the SAPPhIRE model require additional information not in the benchmark (non-overlapping information)? First, SAPPhIRE models are created for the services. Then, these models are compared against the benchmarks to check how much information required by the SAPPhIRE model is covered by the benchmark and how much is not. Table 1 shows the metrics used for this comparison in each example.

Table 1. Metrics used in model comparison

Objectives	Metrics	Definition
To find the Overlapping Information	Directly given in Benchmark	% of the total number of SAPPhIRE constructs, whose information is given in the Benchmark
	Indirectly given in Benchmark	% of the total number of SAPPhIRE constructs, whose information is curated using the available information given in the Benchmark
To find the Non-Overlapping Information	Not given in Benchmark	% of the total number of SAPPhIRE constructs whose information is NOT given in the Benchmark

4. Validation results and discussion

4.1. Validation results

SAPPhIRE models are reconstructed for three example services: (a) Online Retailing, (b) Credit Card Processing, and (c) Credit Card Authorization (Dori, 2001). Multi-instance SAPPhIRE models are used to represent the causal chain of service interactions. Since 'Action' is a subjective interpretation of a State Change, it is excluded from this study, leaving six SAPPhIRE constructs with internal connections. Representations of the three examples using the SAPPhIRE Model and OPM are provided in the Appendix. Table 2 summarizes the comparison of the SAPPhIRE models for all three examples.
The breakdown of the 'Directly Given', 'In Directly Given' and 'Not Given' categories by SAPPhIRE constructs is shown in Figure 2.

Table 2. SAPPhIRE model comparison summary

Examples	Total Number of SAPPhIRE Constructs used in representations	SAPPhIRE Information		
		Directly Given	Indirectly Given	Not Given
Online Retail	24	11	7	6
Credit Card Processing	24	9	8	7
Credit Card Authorizing	18	6	3	9
Grand Total	66	26	18	22

The three examples are representative services whose models are created by benchmarks, From the SAPPhIRE representations of the three examples, we see that SAPPhIRE representations can represent 50% more information than represented by OPM model, thereby producing a richer representation. The following are the salient observations from the performance comparison study:

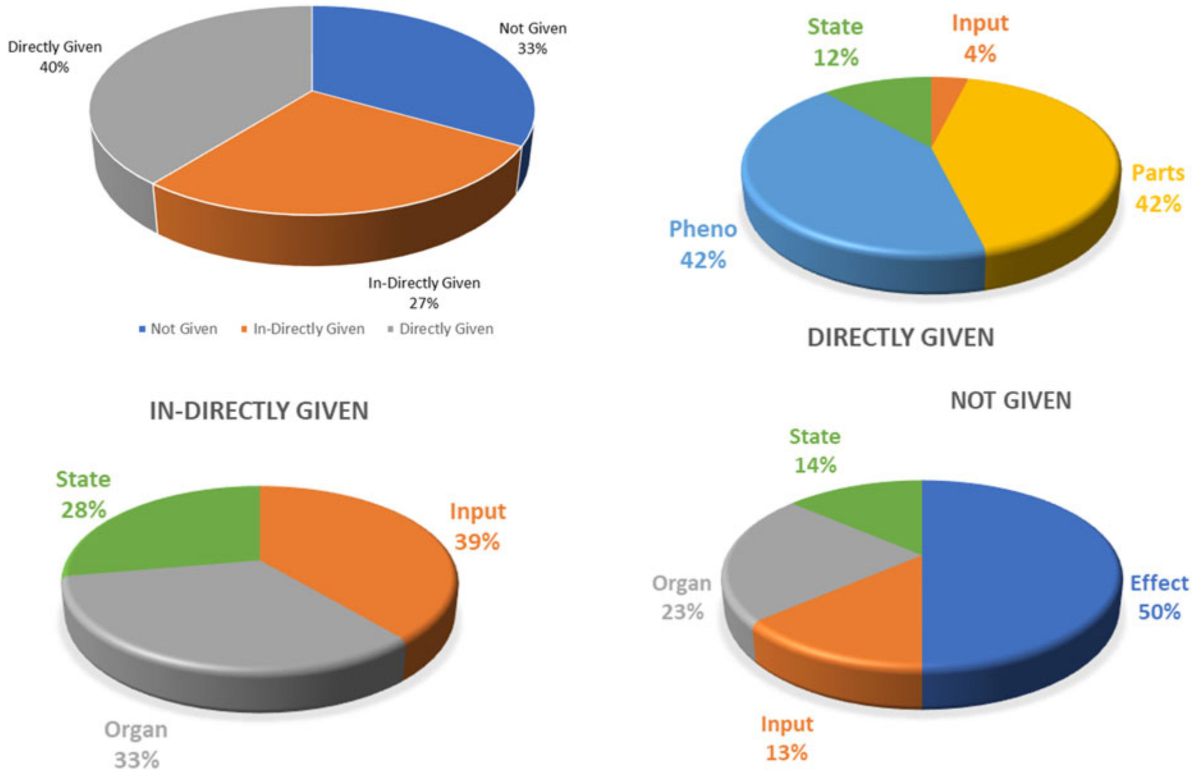


Figure 2. Breakdown of the ‘Directly Given’, ‘Indirectly Given’ and ‘Not Given’ information

- Nearly one-third of the SAPPhIRE constructs did not have any relevant information found from the OPM models. ‘Parts’ and ‘Phenomena’ have most of the relevant information available. For state change, input, and organ, we can construct the relevant information for some of the models. However, for the construct Effect, relevant information is missing totally.
- The SAPPhIRE Constructs can be used to describe all the service elements given in the examples and we did not find any service element that could not be described using SAPPhIRE. SAPPhIRE model increases the richness of the service description by providing specific details.
- We see that the service actors and service activities or interactions are always given. However, these are not enough to describe the services. SAPPhIRE helps to complete the description by bringing in other required details, namely, the applicable service rules (Effect), the external inputs for the service, the state change produced and the conditions or attributes (Organ).
- In the ‘Online Retail’ example, input, conditions or attributes (Organ) and the State Change of every Service Interaction are explicit. However, service rules are missing. In the ‘Credit Card Processing’ example, the conditions or attributes (Organ) are missing for some of the Service Interactions in addition to Service rules. In the ‘Credit Card Authorizing’ example, State Changes are missing in addition to the service rules and the conditions or attributes (Organ). Additionally, the interdependencies of the Service Activities in this example are not given.
- The SAPPhIRE constructs can capture the purpose of a service element. For example, a ‘service catalogue’ and a shopping ‘cart’ are necessary for online retailing. A ‘service catalogue’ is necessary for selecting and adding items to the shopping ‘cart’. The shopping ‘transaction amount’ will not be known if the ‘cart’ is empty.
- The input construct of the SAPPhIRE model explains what external input is necessary for the service to initiate. For example, customer choice from the catalogue is necessary to select and add items to the shopping cart. Another example is the transaction amount and the credit card details, which are input for credit card processing.
- The service rules set by law or company policies that govern service offerings and activities are not called out explicitly in the OPM model. This is necessary to capture in a service description to make the service activity compliant. If the law changes, these service activities and service offerings can change. For example, while ‘processing the credit card’, the credit card details cannot be saved on the merchant server without the consent of the customer, as per Indian law.

- Services produce a state change in the customer's situation. While the ultimate state change is meant for the customer, there could be internal state changes that will lead to the final state change. For example, when the shopping cart is filled, the transaction amount will be known, and the transaction amount input will be necessary for payment to change the status of the goods to 'sold'. Hence, knowing all the state changes is necessary.
- Real-life services have more than one service action. Each of these service actions involves interactions with the customer and the processing of material, energy or information. For example, in 'Online Retail', the customer is involved in 4 service actions, namely, selecting and adding items to the virtual cart, checking out the virtual cart, submitting the Credit Card and processing it. Through multi-instance models, SAPPhIRE can represent the interdependencies of service interactions.
- Conditions and attributes are necessary for service interactions. For example, having sufficient funds on the credit card is necessary for the credit card processing company to process the payment for a transaction. The SAPPhIRE model, with its construct called Organ, can capture these conditions and attributes.

4.2. Discussion

We observed that SAPPhIRE model can represent the operation of a service with a great deal of detail compared to the benchmark. SAPPhIRE model can represent not only the service actors, service artifacts and service activities but also the external input to the service and the state change it produces. It captures the service rules that are the foundation for all service interactions. SAPPhIRE model describes not only how does the service work but also why it works in that way. In addition to service components such as service actors, service artifacts and service activities, SAPPhIRE also explains underlying conditions or attributes through its Organ construct, the Service Rules that are necessary for the service interactions to take place through the Effect construct, the external input to the service through its Input construct and the output of the service through its State Change construct.

5. Conclusions and future work

The research question of this paper is, "*Can a SAPPhIRE model capture and represent all the necessary details of a Service?*" Validation results in section 4 show that the SAPPhIRE model can create rich service representations and explain outcomes through causal relationships between constructs. This causal representation fosters a deeper understanding of system behaviour, enabling targeted and effective solutions. It allows designers to predict intervention effects, conduct root cause analysis, and assess configuration novelty, leading to robust and adaptable designs. Relevant information for SAPPhIRE constructs can be obtained from past design documents or ideation sessions with experts. However, potential downsides like added complexity, resource demands, and uncertainty need to be studied. Future research will empirically study the application of the SAPPhIRE model in service design, PSS design, and its broader interdisciplinary relevance, such as in business operations.

References

Annamalai, G., Hussain, R., Cakkol, M., Roy, R., Evans, S., & Tiwari, A. (2011). An ontology for product-service systems. *In Functional Thinking for value creation* (pp. 231–236). Springer, Berlin, Heidelberg.

Antonova, A. (2018). Smart services as scenarios for digital transformation. *Industry 4.0*, 3(6), 301–304.

Bhattacharya, K., & Chakrabarti, A. (2023). Application of SAPPhIRE Model of Causality in the Design of Product-Service Systems. *In International Conference on Research into Design* (pp. 527–539). Singapore: Springer Nature Singapore.

Bitner, M. J., Ostrom, A. L., & Morgan, F. N. (2008). Service blueprinting: a practical technique for service innovation. *California Management Review*, 50(3), 66–94.

Borgianni, Y., Cascini, G., & Rotini, F. (2012). A proposal of metrics to assess the creativity of designed services. *In DS 73-1 Proceedings of the 2nd International Conference on Design Creativity Volume 1*.

Carsten, S., Monika, J., Jörg, N., & Benedikt, R. (2018). Smart Services. *Proce-dia-Social and Behavioral Sciences*, 238, 192–198.

Christensen, B. T., & Schunn, C. D. (2007). The relationship of analogical distance to analogical function and preinventive structure: The case of engineering design. *Memory & cognition*, 35, 29–38.

Chuang, L. M., Lee, Y. P., & Yao, F. T. (2022). Intelligent Machinery Product Service Blueprint Development and Verification: An Empirical Study of Machine Tool Industry. *IEEE Access*, 10, 19796–19811.

Coreynen, W., Vanderstraeten, J., van Witteloostuijn, A., Cannaerts, N., Loots, E., & Slabbinck, H. (2020). What drives product-service integration? An abductive study of decision-makers motives and value strategies. *Journal of Business Research*, 117, 189–200.

Dori, D. (2001). Object-process methodology applied to modeling credit card transactions. *Journal of Database Management (JDM)*, 12(1), 4–14.

Dori, D. (2011). Object-process methodology. *In Encyclopedia of Knowledge Management*, Second Edition (pp. 1208–1220). IGI Global.

Fakhfakh, S., Jankovic, M., Hein, A. M., Chazal, Y., & Dauron, A. (2021). Proposition of an ontology to support product service systems of systems engineering. *Systems Engineering*, 24(5), 293–306.

Fu, K., Murphy, J., Yang, M., Otto, K., Jensen, D., & Wood, K. (2015). Design-by-analogy: experimental evaluation of a functional analogy search methodology for concept generation improvement. *Research in Engineering Design*, 26, 77–95.

Gaiardelli, P., Pezzotta, G., Rondini, A., Romero, D., Jarrahi, F., Bertoni, M., ... & Cavalieri, S. (2021). Product-service systems evolution in the era of Industry 4.0. *Service Business*, 15(1), 177–207.

Geum, Y., & Park, Y. (2011). Designing the sustainable product-service integration: a product-service blueprint approach. *Journal of Cleaner Production*, 19(14), 1601–1614.

Herzberger, P., Behncke, F. G. H., Schenkl, S., & Lindemann, U. (2013). Interactive modeling and evaluation of product-service systems. *In DS 75-4: Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 4: Product, Service and Systems Design*, Seoul, Korea, 19-22.08. 2013 (pp. 359–368).

Hill, F., Santoro, A., Barrett, D. G., Morcos, A. S., & Lillicrap, T. (2019). Learning to make analogies by contrasting abstract relational structure. *arXiv preprint arXiv:1902.00120.*

Jiang, S., Hu, J., Wood, K. L., & Luo, J. (2022). Data-driven design-by-analogy: state-of-the-art and future directions. *Journal of Mechanical Design*, 144(2), 020801.

Kim, Y. S. (2020). A representation framework of product–service systems. *Design Science*, 6, e3.

Li, F., & Lu, Y. (2021). Engaging end users in an ai-enabled smart service de-sign-the application of the smart service blueprint scape (SSBS) framework. *Proceedings of the Design Society*, 1, 1363–1372.

Linsey, J. S., Wood, K. L., & Markman, A. B. (2008). Modality and representation in analogy. *Ai Edam*, 22(2), 85–100.

McKoy, F. L., Vargas-Hernández, N., Summers, J. D., & Shah, J. J. (2001, September). Influence of design representation on effectiveness of idea generation. *In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 80258, pp. 39–48). American Society of Mechanical Engineers.

McSorley, G., Fortin, C., & Huet, G. (2014). Modified SAPPhIRE model as a framework for product lifecycle management. *In DS 77: Proceedings of the DESIGN 2014 13th International Design Conference* (pp. 1843–1852).

Moon, S. K., Oh, H. S., Kim, S., & Hwang, J. (2013). A product-service system design framework using objective-oriented concepts and blueprint. *In DS 75-4: Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 4: Product, Service and Systems Design*, Seoul, Korea, 19-22.08. 2013.

Moreno Grandas, D. P., Blessing, L., Yang, M., & Wood, K. (2015). The potential of design-by-analogy methods to support product, service and product service systems idea generation. *In DS 80-5 Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 5: Design Methods and Tools-Part 1*, Milan, Italy, 27-30.07. 15 (pp. 093–104).

Moreno Grandas, D. P., Blessing, L., Yang, M., & Wood, K. (2015). The potential of design-by-analogy methods to support product, service and product service systems idea generation. *In DS 80-5 Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 5: Design Methods and Tools-Part 1*, Milan, Italy, 27-30.07. 15 (pp. 093–104).

Moreno, D. P., Yang, M. C., Blessing, L., & Wood, K. L. (2014). Analogies to Succeed: Applications to a Service Design Problem. *DS 81: Proceedings of NordDesign 2014, Espoo, Finland 27-29th August 2014*, 520–529.

Sarkar, P., & Chakrabarti, A. (2011). Assessing design creativity. *Design studies*, 32(4), 348–383.

Shihundla, T. B., Mpofu, K., & Adenuga, O. T. (2019). Integrating product-service systems into the manufacturing industry: Industry 4.0 perspectives. *Procedia CIRP*, 83, 8–13.

Shimomura, Y., & Tomiyama, T. (2002, November). Service modeling for service engineering. *In International Working Conference on the Design of Information Infrastructure Systems for Manufacturing* (pp. 31–38). Springer, Boston, MA.

Shostack, G. L. (1982). How to design a service. *European Journal of Marketing*.

Tomiyama, T. (2001, December). Service engineering to intensify service contents in product life cycles. *In Proceedings Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing* (pp. 613–618). IEEE.

Tukker, A. (2004). Eight types of product–service system: eight ways to sustainability? Experiences from SusProNet. *Business strategy and the environment*, 13(4), 246–260.

Venkataraman, S., & Chakrabarti, A. (2009). SAPPhIRE–an approach to analysis and synthesis. *In DS 58-2: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 2, Design Theory and Research Methodology*, Palo Alto, CA, USA, 24.-27.08. 2009 (pp. 417–428).

Appendix:

SAPPhIRE Representation of Services

SAPPhIRE Representation and Object Process Methodology (OPM) Representation of Service Examples reported in the Literature (Dori, 2001).

Example 1 - Online Retailing:
OPM Representation (Dori, 2001):

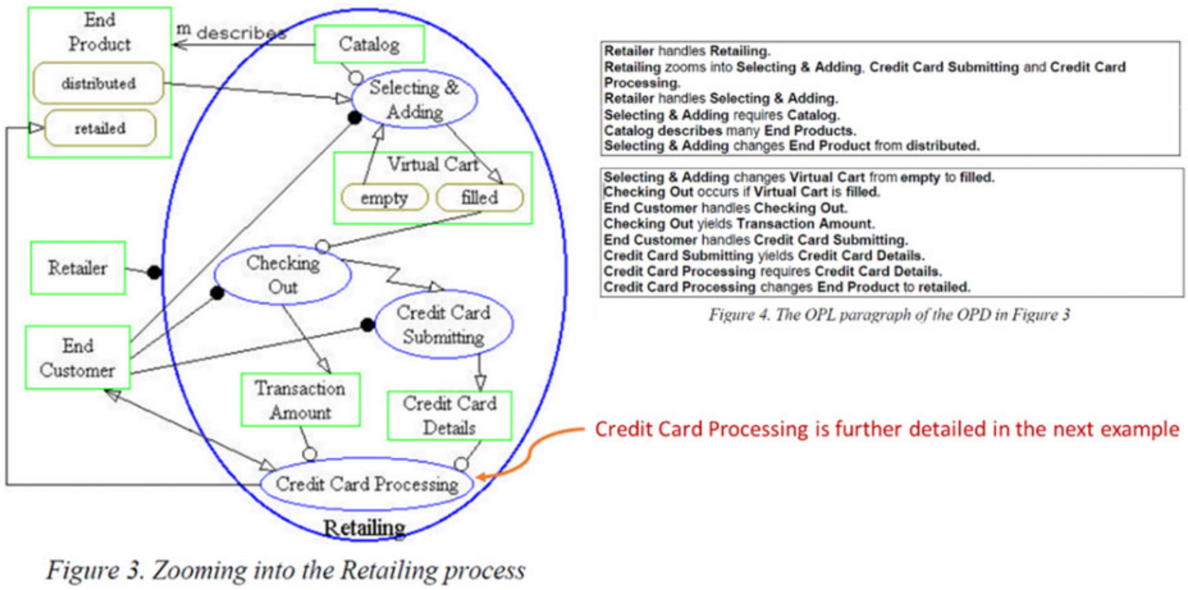
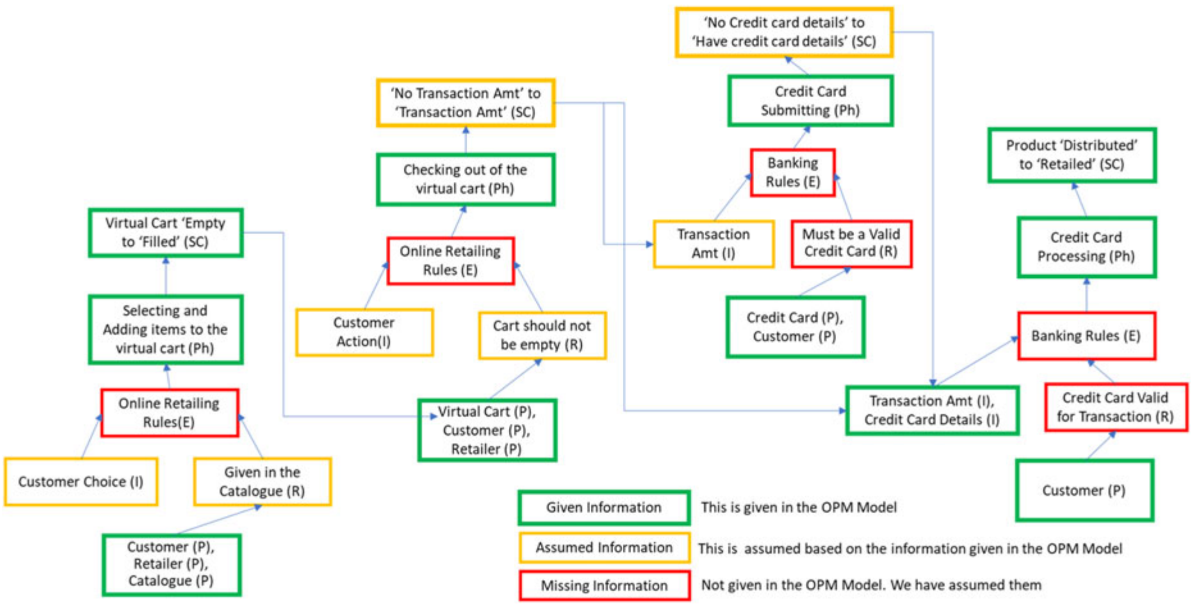


Figure 3. Zooming into the Retailing process

Figure 4. The OPL paragraph of the OPD in Figure 3

SAPPhIRE Representation:



Example 2 - Credit Card Processing:
OPM Representation (Dori, 2001):

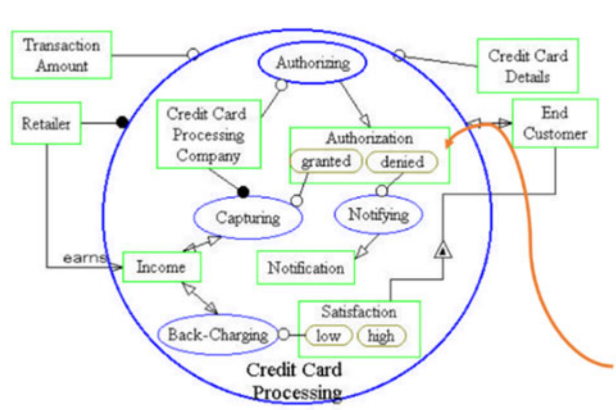


Figure 5. Zooming into Credit Card Processing of Figure 3

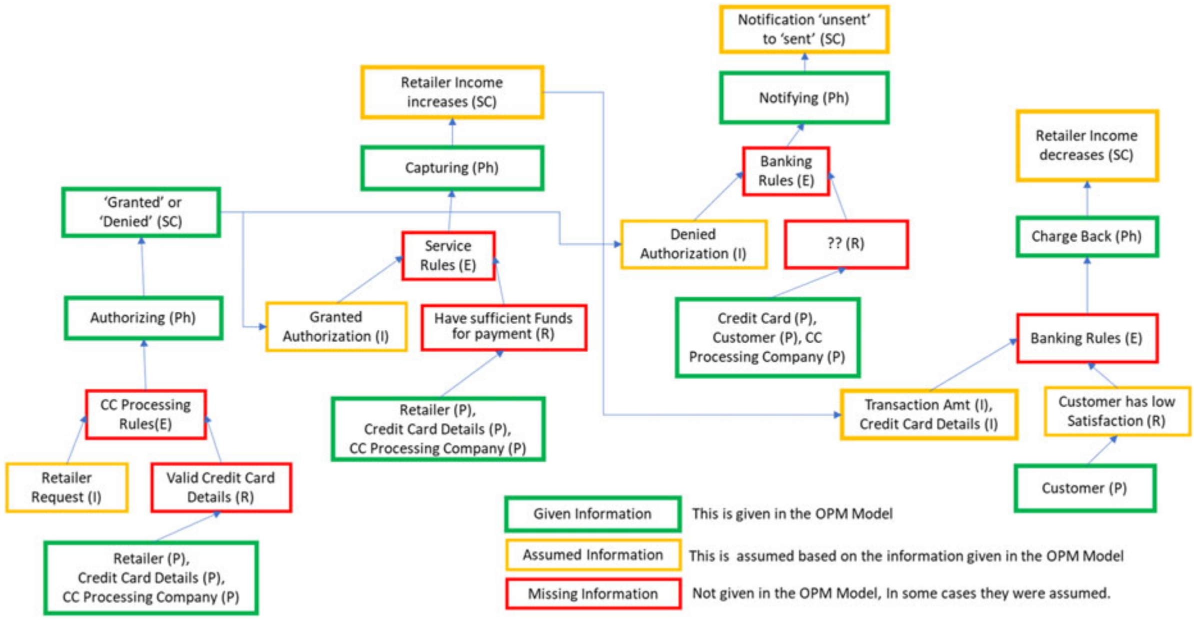
Retailer handles **Credit Card Processing.**
Credit Card Processing Company handles **Credit Card Processing.**
Credit Card Processing requires **Transaction Amount.**

Credit Card Processing requires **Credit Card Details.**
Credit Card Processing affects **End Customer.**
Credit Card Processing zooms into **Authorizing, Capturing, Notifying** and **Back-Charging.**
Authorizing yields **Authorization.**
Authorization can be **granted** or **denied.**
Notifying occurs if **Authorization** is **denied.**
Notifying yields **Notification.**
Capturing occurs if **Authorization** is **granted.**
Capturing affects **Income.**
Retailer earns Income.
End Customer exhibits **Satisfaction.**
Satisfaction can be **high** or **low.**
Back-Charging occurs if **Satisfaction** is **low.**
Back-Charging affects **Income.**

Figure 6. The OPL paragraph of the OPD in Figure 5

Authorization is further detailed in the next example

SAPPhIRE Representation:



Example 3 - Credit Card Authorization:
OPM Representation (Dori, 2001):

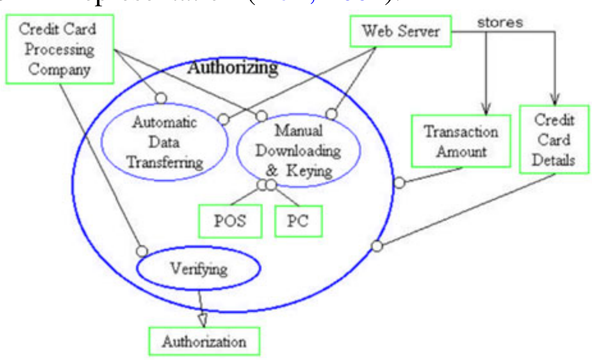


Figure 8. Zooming into Authorizing of Figure 5

Web-Server holds Transaction Amount and **Credit Card Details.**
Authorizing zooms into **Manual Downloading & Keying, Automatic Data Transferring** and **Verifying.**
Manual Downloading & Keying requires **Web-Server** and **Credit Card Processing Company.**
Automatic Data Transferring requires **Web-Server** and **Credit Card Processing Company.**
Credit Card Processing Company enables either **Manual Downloading & Keying** or **Automatic Data Transferring.**
Web-Server enables either **Manual Downloading & Keying** or **Automatic Data Transferring.**
Manual Downloading & Keying requires either **Point POS** or **PC.**
Verifying requires **Transaction Amount** and **Credit Card Details.**
Verifying yields **Authorization.**

Figure 9. The OPL paragraph that corresponds to the OPD in Figure 8

SAPPhIRE Representation: