

Metamodel for Artefacts in Service Engineering Analysis and Design

Purnomo Yustianto, Robin Doss

Abstract—As a process of developing a service system, the term ‘service engineering’ evolves in scope and definition. To achieve an integrated understanding of the process, a general framework and an ontology are required. This paper extends a previously built service engineering framework by exploring metamodels for the framework artefacts based on a foundational ontology and a metamodel landscape. The first part of this paper presents a correlation map between the proposed framework with the ontology as a form of evaluation for the conceptual coverage of the framework. The mapping also serves to characterize the artefacts to be produced for each activity in the framework. The second part describes potential metamodels to be used, from the metamodel landscape, as alternative formats of the framework artefacts. The results suggest that the framework sufficiently covers the ontological concepts, both from general service context and software service context. The metamodel exploration enriches the suggested artefact format from the original eighteen formats to thirty metamodel alternatives.

Keywords—Artefact, framework, service, metamodel.

I. INTRODUCTION

AN umbrella research of this paper produces service engineering framework labelled as the General Service Engineering Framework (GSEF) which covers service aspects from the business side to the informatics side, i.e. Business Capability, Business Model, Service Value, Interaction Model, Process Model and, Software-Service Model [1].

The framework is presented in a three layers structure of: Activity, Artefact, and Modelling (Fig. 1) [2]. The activity specifies the steps of analysis and design process. The artefact defines the product of each step while also implying the dependency and flow of the produced artefact. The modelling layer is a container of artefact format.

The paper is divided into two parts. The first part examines the framework in terms of artefacts coverage to the produced ontology to assess the completeness of the framework and characterize the metamodel suitable for the artefacts. Two service ontologies are used in this paper: general service ontology (Fig. 2), and software service ontology (Fig. 3) [3].

II. ONTOLOGY AND ARTEFACT MAPPING

To verify the completeness of framework coverage on the aspects of service system, a comparative triangulation is made between the produced service ontology with artefacts defined

Purnomo Yustianto is with the School of Information Technology - Deakin University, Geelong, VIC, Australia (corresponding author; e-mail: purnostian@deakin.edu.au).

Robin Doss is with the School of Information Technology Deakin University, Geelong, VIC, Australia (e-mail: robin.doss@deakin.edu.au).

in the framework. This cross-referencing into the proposed metamodels also serves as a bridge to characterize the artefacts form.

The assessment is performed in four parts divided by sub-stage, i.e. activity, in the latest iteration of proposed framework (Fig. 1): (1) *Understanding Service Context*, coded activity 1, (2) *Defining Service Concept*, coded as activity 2, (3) *Business Service Design*, coded as activity 3, and (4) *Software Service Design*, coded as activity 4.

A. Activity 1: Understanding Service Context

Before proposing new or improved services, an understanding toward the context is required. The activity in the first part of the identification stage capture and analyse the existing situation of the environment.

The activity covers foundational aspects of an organization. As illustrated in Fig. 4, four existing-situation aspects are captured as artefacts in this sub-stage: (1) business directives, (2) business model, (3) business process, and (4) business capability. The framework proposes to capture the guiding business directives as a list of narratives, which can be presented in tabular format. The current business model is visualized with Business Model Canvas (BMC) format [4], and the business process with Business Process Modeling and Notation (BPMN) metamodel [5].

In parallel, owned (and potential) capabilities of the organization are also examined, and presented in a capability diagram. The analysis and modelling could be based on Component Business Model (CBM) [6], which is based on organization structure, or based on SoaML capability diagram [7].

The result of this sub-stage should be an identification of opportunities to be pursued in provisioning a business service. The opportunity could be numerous therefore, a ranked list should be made based on combination of various factors such as feasibility, prospective gain and cost, or resources required.

In identifying the opportunity, external perspectives must also be incorporated. These external perspectives should capture the market opportunity and business partnership. In the framework, the combination of outward and inward-looking perspective is accommodated in the last artefact, the opportunity list. The framework does not specify the standard format for the list, but it is usually in a narrative format produced from business analysis techniques, such as a Strengths, Weakness, Opportunities, Threats (SWOT) or Value Chain Analysis. Any format should be acceptable as long as it helps the management to decide a specific opportunity to pursue.

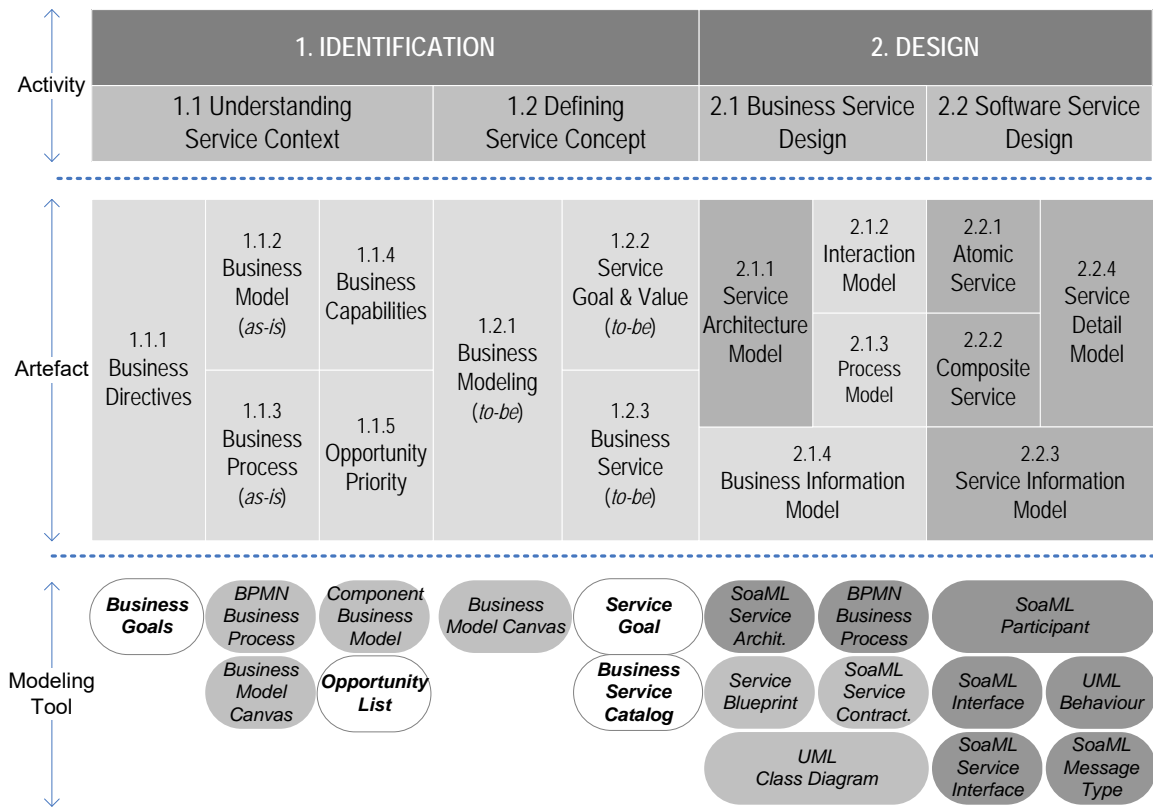


Fig. 1 Service Engineering Framework [2]

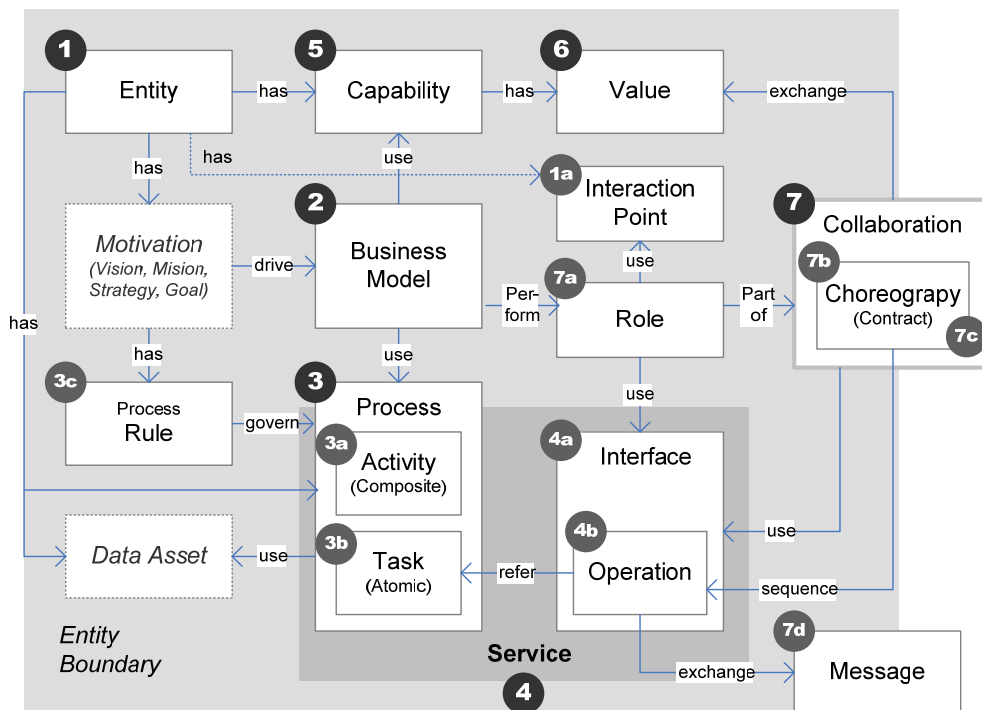


Fig. 2 General Service Engineering Ontology

[10]-[12] as part of (service) ‘product model’ [13]. The ontology artefact should cover ontology components related to business models and service system, as a part of the whole business domain ontology. The artefact can be presented in UML class diagram [2].

D. Activity 4: Software Service Design

The fourth activity mirrors the activity in the ‘Business service design’ sub-stage. The difference is that the service elaborated in an IT context, i.e. software context, rather than in a business context as in the previous sub-stage. Fig. 7 lists four aspects to be covered from the ontology at the software-service level: (1) Atomic service, (2) Composite service, (3) Service Detail, and (4) Service Information.

The *atomic service* specifies the design of self-sufficient software in terms of the service interface, contained operations and its underlying behaviour. The artefact is presented as diagrams of SoaML Interface [7].

The *composite service* describes the combined use of the atomic service in the form SoaML Service Interface which includes the behaviour in the form of a sequential arrangement of operations invocation. In the case of a composite service, it contains invocations to external services, i.e. services provided by other participants, the behaviour specification represents a software level collaboration-interaction with an external software component

The *service detail* aspect gathers the software services into an abstracted form of software components with a service port, invoked and invocable services into a SoaML Participant. All service behaviours are to be detailed in this container, to be implemented later as software components. These components serve as a representation of an interacting party, either as a consumer, provider or both.

Finally, the service information collects all of the information exchanged between services as operands and return values of invoked operations. SoaML message type diagram is defined for each service interaction transaction and cross-referenced with business information artefacts from previous activity to be standardized in maintaining consistency while at the same time facilitate message type reusability.

In general, the described triangulation between ontology and framework artefacts demonstrates a sufficiency of framework coverage in assessing service aspects and components, both in business and software perspective. In this stage, the prescribed metamodel of targeted artefacts is an open specification. The suggested metamodels are demonstrated to be sufficient for the case studies. But the dynamic nature of the metamodel landscape may offer alternative formats that might be better in capturing the modelling needs.

III. ALTERNATIVE METAMODELS

In another part of the research an emerging structure of a metamodel is defined. While a clear differentiation between metamodel groups is not claimed, seven stereotypes of metamodel are offered: (1) goal, (2) enterprise, (3) business model, (4) service, (5) process, (6) software, and (7) system.

The “service” perspective emerges as an alternative integrative approach, as in “enterprise” perspective, in traversing the context between “business” and “software” aspects. The “service” perspective covers the aspects of “business model”, “business capability”, “business interaction”, “value proposition”, “value exchange”, “customer interaction”, and “software service”, which is consistent with the produced framework.

Open Science Index, Computer and Systems Engineering Vol:12, No:9, 2018 publications.waset.org/10009469.pdf

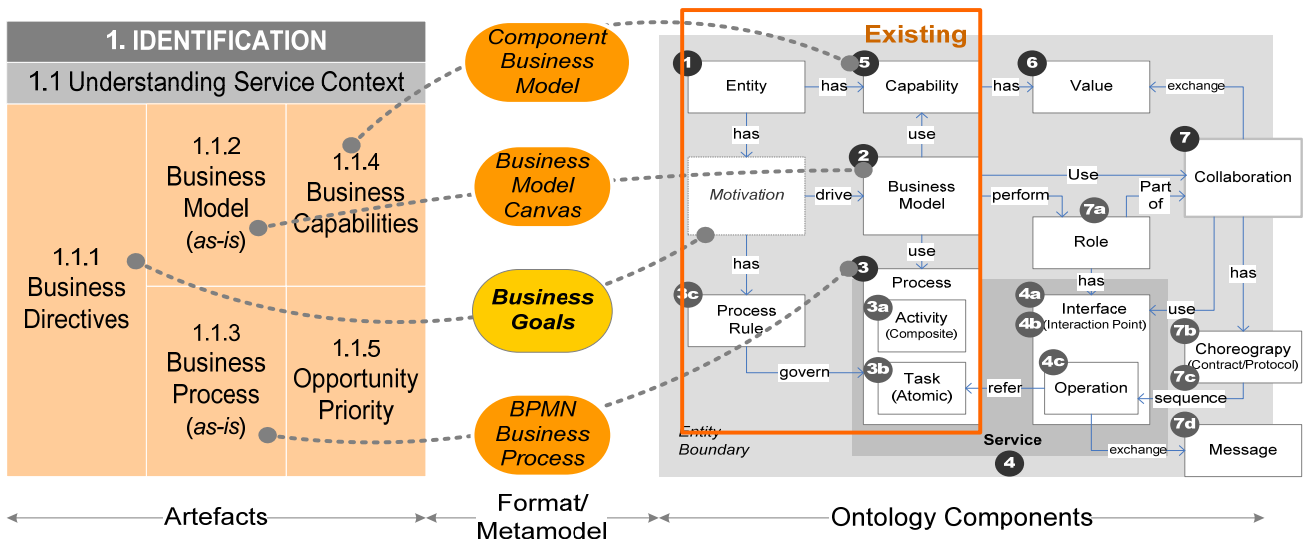


Fig. 4 Artefact Ontological Position in Activity 1

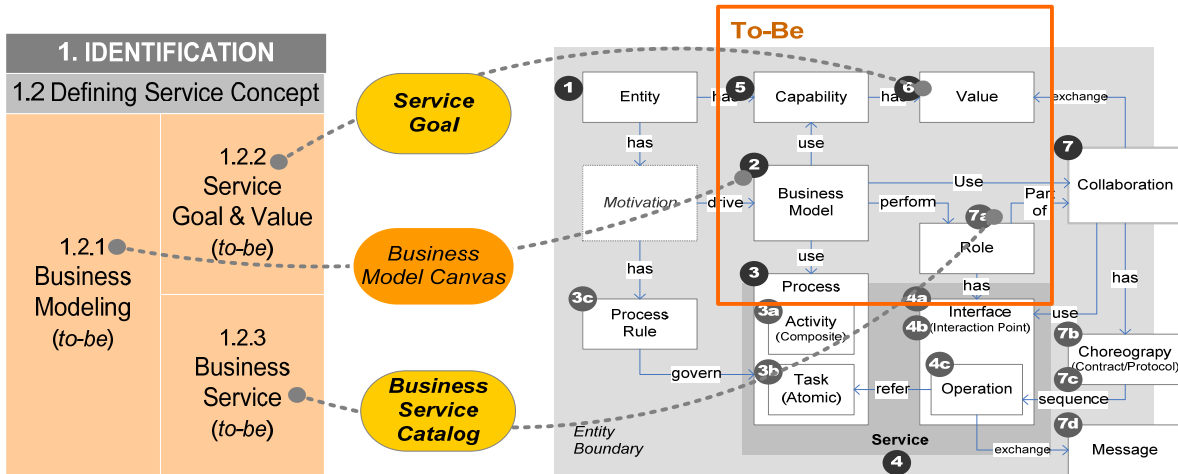


Fig. 5 Artefact Ontological Position in Activity 2

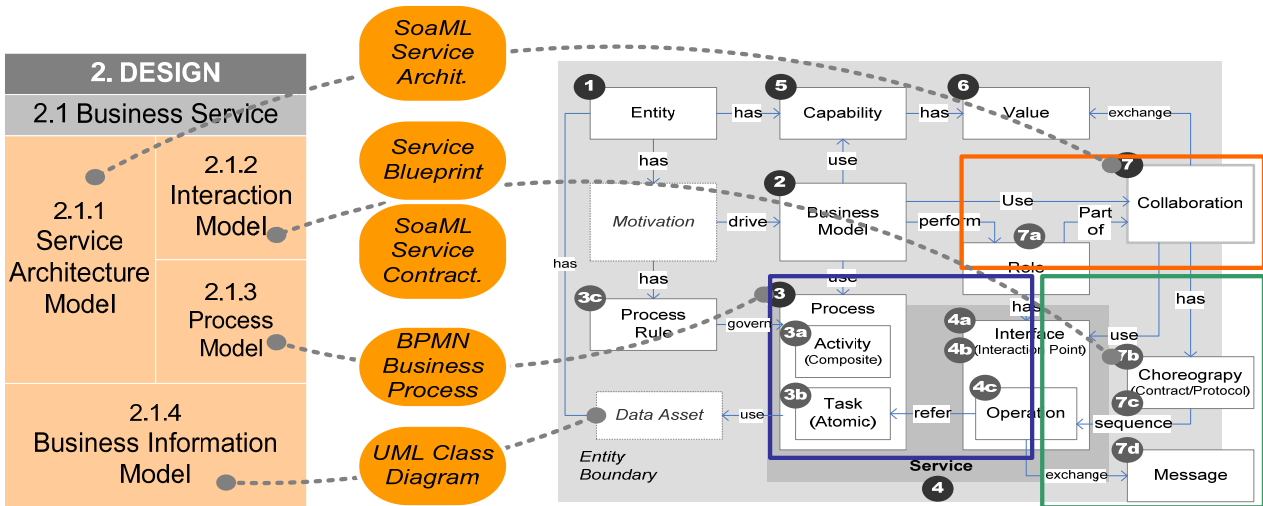


Fig. 6 Artefact Ontological Position in Activity 3

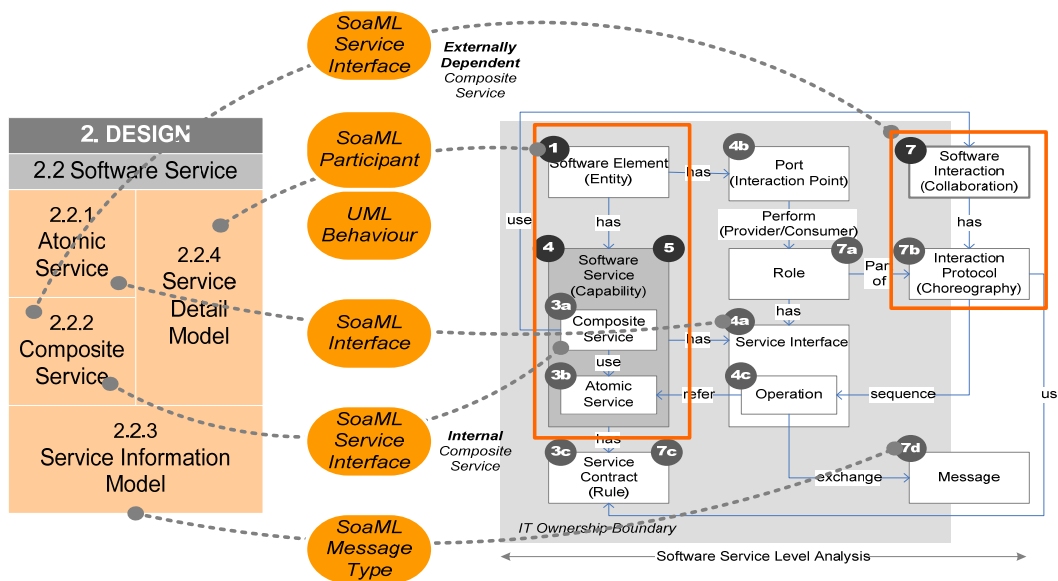


Fig. 7 Artefact Ontological Position in Activity 4

The observed proliferation pattern of metamodel projects its nature as an ever-dynamic landscape. Some cross-disciplinary initiatives might pursue an integrative universal metamodel. But a more pragmatic and feasible option is available in the form of specifying a metamodel stack [14], [15], such as adopted by the framework. As a set of originally unrelated metamodels, special care must be taken to ensure the traceability and translatability of the artefacts between stage and activity.

Examining the metamodel landscape, several newer metamodel propositions are worthy to be proposed as metamodel alternatives. The following sections presents these potential metamodel alternatives divided into four item: (1) OMG's Business Motivation Model (BMM) [16], (2) OMG's Value Definition Modelling Language (VDML) [17], (3) alternative Business Modelling, and (4) alternative Interaction Modelling.

A. Business Motivation Model

As the name implied, Business Motivation Model (BMM) [16] covers motivational aspects of a business case. It is situated on the strategic level of a business model by defining the drivers, the element and its interrelations for a business plan, without elaborating the detailed aspects of business process and business structure.

From the illustration in Fig. 8, BMM can be seen as a ontological structure for business motivation, which relates to two aspects: (1) *Ends*, defined as situations to be achieved, i.e. goals and objective, and (2) *Means*, as concepts adopted to achieve the ends, i.e. strategies, tactics, policies, and rules.

Not many published articles are found documenting BMM adoption for real world cases, but the recent update to the specification introduces metamodel notations for modelling purposes [18]. BMM metamodel can be useful for structuring a business goal artefact in the framework. If needed, BMM is useful to document the traceability between the components of goals, objectives, strategies, tactics, policies, and rules, as a directive context for a service system.

B. Value Definition Modelling Language

Value Definition Modelling Language (VDML) [17] is a relatively recent metamodel to be introduced by OMG. It covers business concepts in terms of activities, roles, flows, participants and capabilities in a higher abstraction compared to BPMN. VDML is proposed as a modelling language for business analysis with focus on value creation and exchange, by combining external perspective on market opportunities with extended organisational capability structure.

Like UML and SoaML, VDML is actually a family of diagrams, which contains eight type of diagrams: (1) Role Collaboration, (2) Value Proposition Exchange, (3) Activity Network, (4) Collaboration Structure, (5) Capability Library, (6) Capability Heat Map, (7) Capability Management, (8) Measurement Dependency. The detail specification of these diagrams is considered to be out of scope for this paper, but it suffices to identify the components relevant to the framework, with comparison to existing metamodels.

Role collaboration and *Value proposition exchange* describes service architecture, as a network of providers and consumers, in term of participant role and value (potentially) exchanged. For this purpose, SoaML's Service Architecture is decided to be sufficient to present the similar abstraction, with a compact abstraction for participants, roles and interactions.

On the other hand, VDML is quite attractive to represent the concept of capability in the framework. Among three of its capability related diagrams, two are identified to have potential use in the framework: (1) capability Library and (2) capability management.

As can be seen in Fig. 9, the capability library provides a hierarchical structure of capabilities in an organization, which could be useful to replace the use of a Component Business Model (CBM) in the identification stage.

The capability management (Fig. 10) provides a graphical abstraction for ownership, dependency and exposition of capability within an organization or an extended organization. It has similarity with SoaML's participant diagram in software-services but resides in the 'capability' context. VDML capability management diagram is also identified to be a potential metamodel for a capability model artefact, as it has the features to accommodate an extended business model.

C. Business Modelling

Business modelling is a growing research area which flourished since the introduction of BMC in 2010. In the metamodel exploration, three business modelling languages are identified to be potentially relevant for the framework: (1) Service Dominant Business Model (SDBM) [19], and (2) Service BMC (S-BMC) [20].

SDBM offers a simple view of service business model with only four components [19]: (1) Service as the core element, (2) Management, representing the 'how' aspect of service access, analogous with relationship and channel components in BMC, (3) Cost-benefit, characterize the service value for specific participant in mostly financial context, and (4) Actor as providing or consuming participant of the service. All of the components are visualized as layers circling a specific service in the centre (Fig. 11).

SDBM strength lies in its simplicity in abstracting a service with multiple participants. But compared to the standard BMC, SDBM lacks the specification of activities and resources involved in the service provision. Despite its limitation, SDBM is still an attractive format as an early form of the service business model, describing a preliminary architecture of service participants.

Service BMC (S-BMC) is another reformulation of the BMC format by extending its usage for multi-party business models. In S-BMC, seven BMC components are spread vertically, and each participant's perspective are specified as layers of these component.

Three perspectives are offered in its basic format: (1) customer perspective, (2) internal organization perspective, and (3) partner perspective (Fig. 12). Additional layers might be added for other business participant, as intermediaries either toward the customer or partner side (Fig. 13).

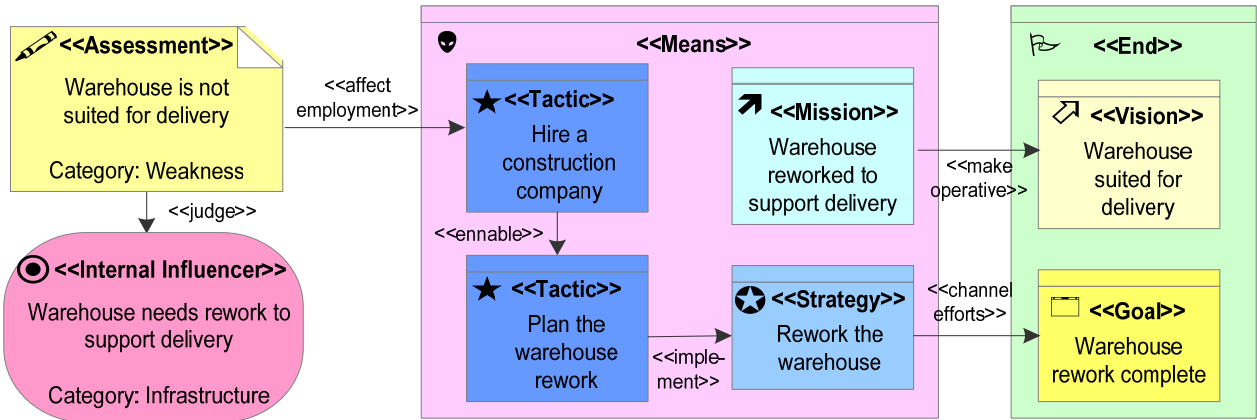


Fig. 8 Example of Business Motivation Model Artefact [18]

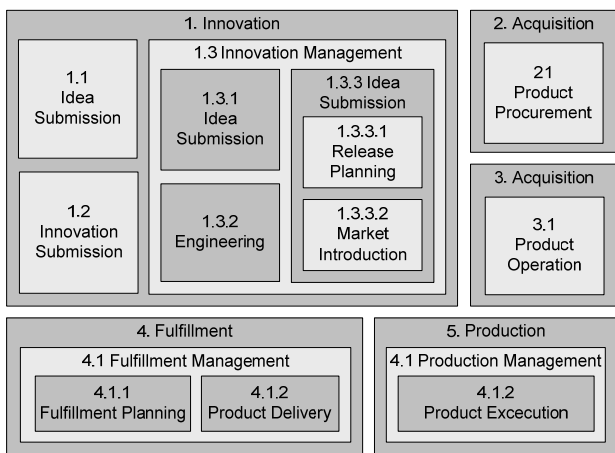


Fig. 9 Example of VDML Capability Library [17]

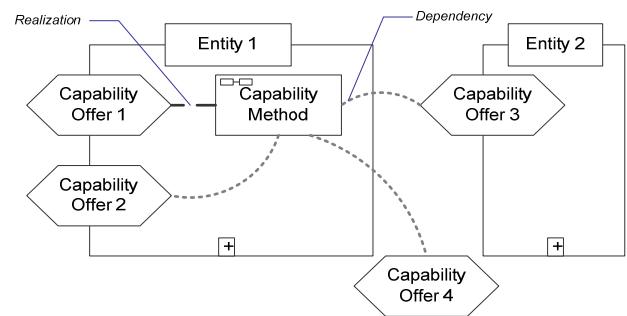


Fig. 10 Example of VDML Capability Management [17]

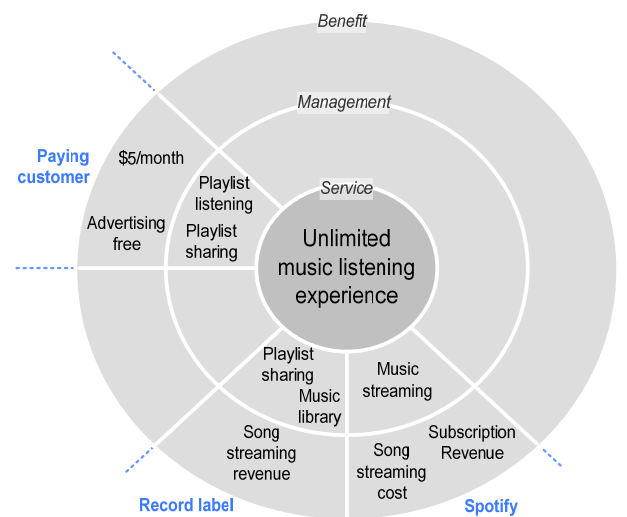


Fig. 11 Example of Service Dominant Business Model (SDBM) [19]

It can be observed that the three alternatives business model formats try to address BMC limitation in representing an extended organization, as a multi-participant forming a service system. BMC or BMI format is fairly sufficient for simple business cases with one dominant providing participant. But to

represent complex business model architecture, SDBM or S-BMC might be required.

D. Interaction Modelling

Three interaction modelling formats are identified in

metamodel exploration: (1) Process Chain Network (PCN) [21], (2) Service Modelling Language (Service ML) [22], and (3) Service Journey Modelling Language (SJML) [23].

PCN is proposed in service operation management field as an attempt to improve Service Blueprinting (SBP) [21]. PCN focuses on the touch point by introducing three degree of interaction layer for each party, i.e. provider and consumer: (1) direct interaction, (2) surrogate interaction, and (3) independent processing.

The presentation has similar feature with SBP by defining participant activities in the interaction, but the elaboration is not only in the provider side but also accommodated in the consumer side (Fig. 14). PCN presentation also provides an abstraction of business-process networks, by aligning series of interactions for multiple service participants (Fig. 15). In this sense, PCN can be seen as an alternative improvement of SBP for multiple participants' interaction.

Customer perspective	Customer (Customers in the business model)						
	<i>(Cost borne by customers)</i>	<i>(Resources provided by customers)</i>	<i>(Activities carried out by customers)</i>	<i>(Value proposition for customers)</i>	<i>(Customers contribution in relationship)</i>	<i>(Channel provided by customers)</i>	<i>(Revenues captured by customers)</i>
Company perspective	Cost Structure <i>(Cost borne by local company)</i>	Key Resources <i>(Resources provided by local company)</i>	Key Activities <i>(Activities carried out by the local company)</i>	Value Proposition <i>(Value proposed by the local company)</i>	Relationship <i>(Contribution of local company in relationship)</i>	Channels <i>(Channels provided by the local company)</i>	Revenue Streams <i>(Revenues captured by local company)</i>
Partner perspective	<i>(Cost borne by partners)</i>	<i>(Resources provided by partners)</i>	<i>(Activities carried out by partners)</i>	<i>(Value proposition for partners)</i>	<i>(Partner contribution in relationship)</i>	<i>(Channel provided by partners)</i>	<i>(Revenues captured by partners)</i>
	Key Partner (Partners in the business model)						

Fig. 12 Structure of Service BMC (S-BMC) [20]

	Customer: - Differentiated Business-to-Consumer Segments - Low number of transaction / Low or no willingness to pay	CUSTOMER					Merchant: - Undifferentiated Business-to-Business Segments - Low number of transaction / Low or no willingness to pay -
	Cost Structure	Key Resources	Key Activities	Value Proposition	Relationship	Channels	Revenue Streams
Customer	- Debit card - Compatible mobile device	- Compatible mobile device - Right of debit card disposal	- Registration - Configuration - Authentication - Account mgt.	- Payment safety - Payment speed - Saving by m-marketing.	- Registration for automated or self service	- Smartphone with installed mobile app	- Saving (e.g. monetary and time)
Merchant	- Setup cost & operation - Infrastructure (terminal etc.)	- System, terminal - Right of merchant account disposal - Staff	- Configuration - Awareness - Confirmation - Settlement.	- Payment safety - Payment speed - Cost saving - Customer data	- Automated or self service - Personal store assistant	- Web - After sales - Store	- Customer per sale or purchase volume
Company	- Setup cost & operation (e.g. staff, negotiation) - Infrastructure	- Payment infrastructure, system - Customer and merchant basis - Staff	- App development, & operation - Transmission of payment	- Increased platform sales - Operational cost reduction	- Integrated into service provision	- Mobile payment system - Mobile app - After sales serv.	- Flat fee (membership) - Transaction fee (discount)
Payment provider	- Setup cost & operation - Infrastructure	- Payment Infrastructure	- Risk mgmt. - Settlement (direct debit)	- Sales increase - Cost saving	- Automated service (settlement)	- Web - Sales force	- Merchant usage fee (transaction based)
Technology provider	- Setup cost & operation - Infrastructure	- Checkout system, plugin, terminal		- Sales increase	- Personal assistance (distribution)	- Sales force	- Per sale from merchant
	Payment Provider (e.g. bank)	KEY PARTNER					Technology Provider (e.g. ISP)

Fig. 13 Example of S-BMC Artefact with five party layers [20]

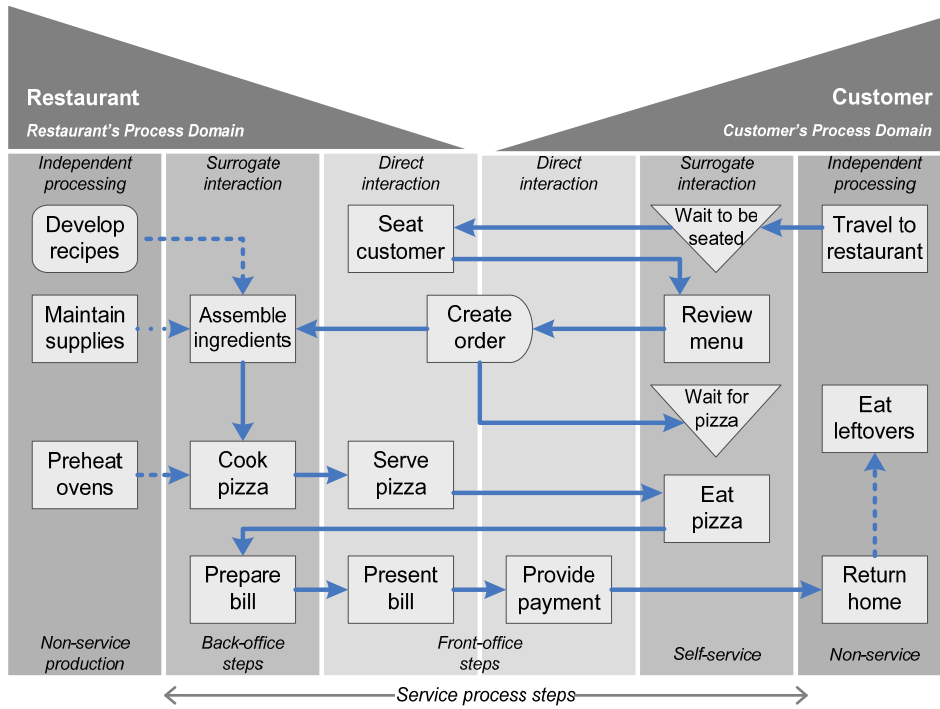


Fig. 14 Example of Process Chain Network (PCN) Artefactn [21]

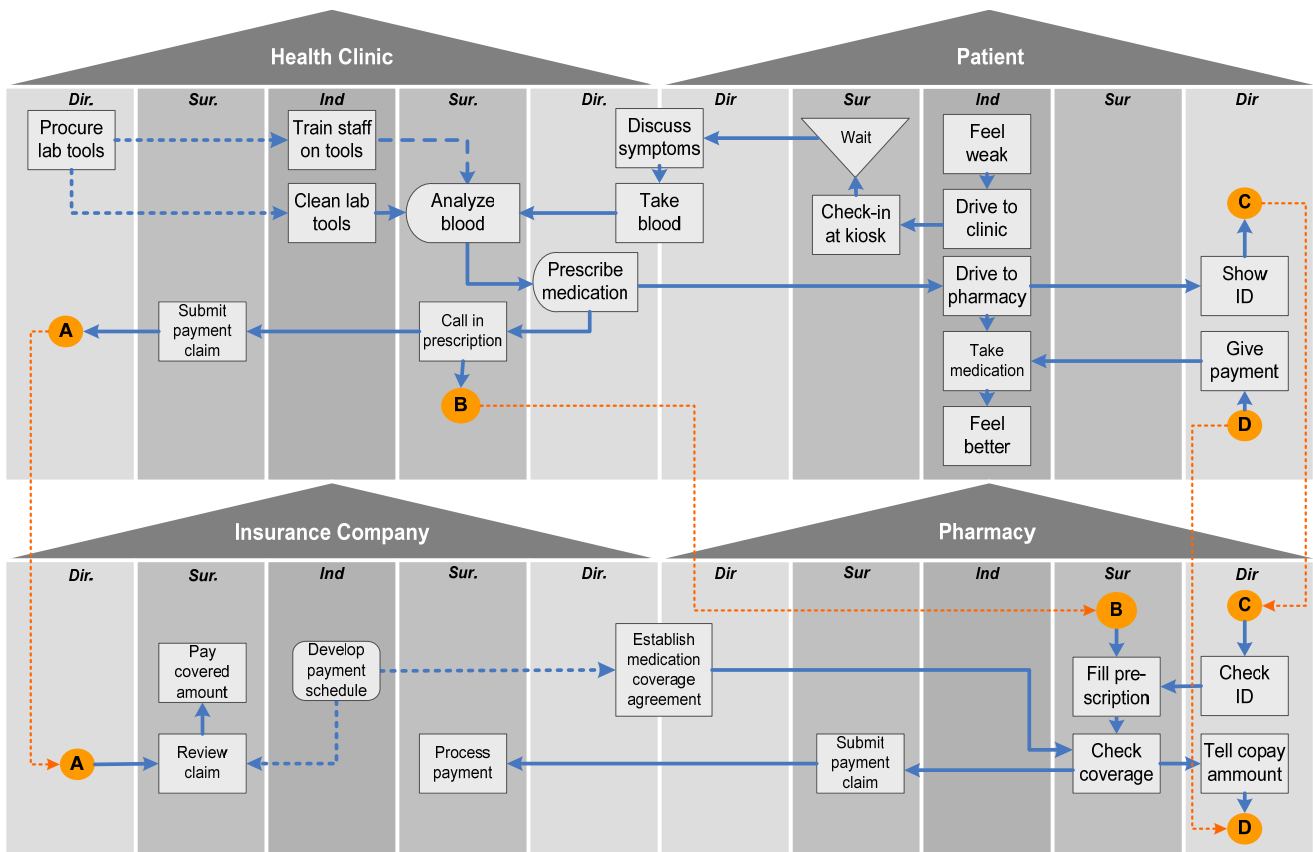
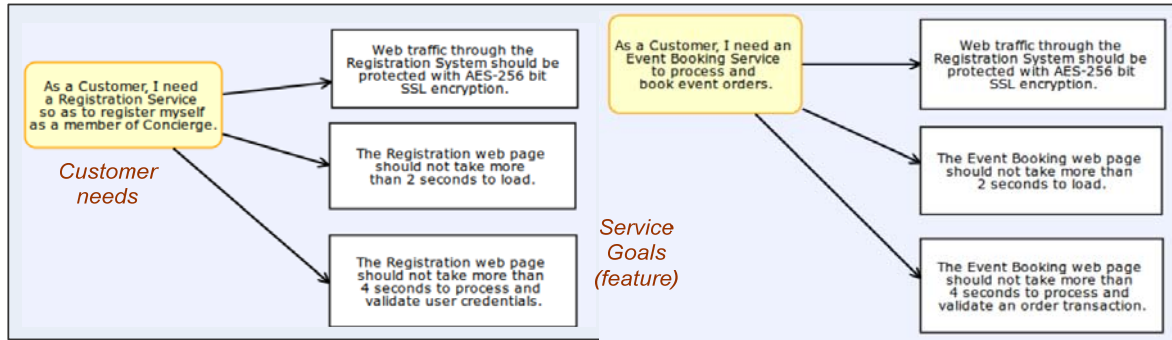
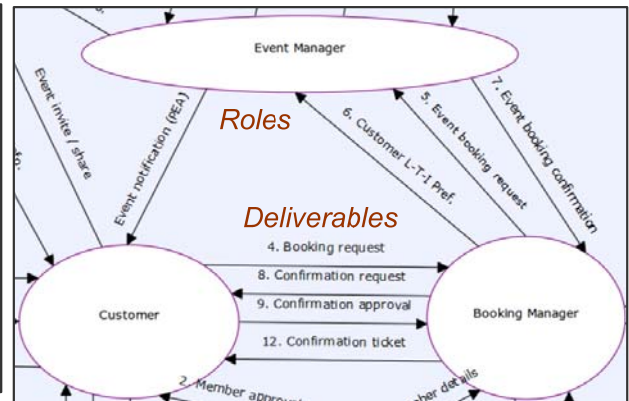
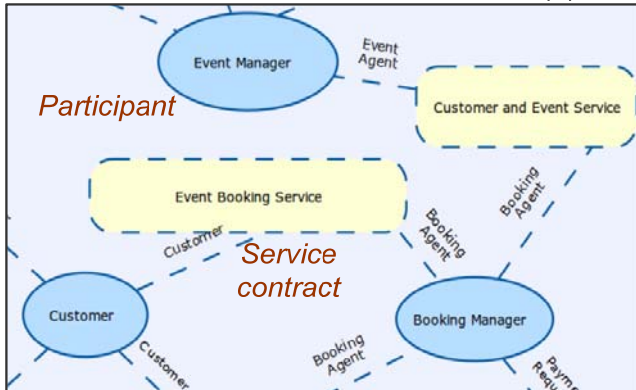


Fig. 15 Network of Interactions in PCN [21]

ServiceML [22] was proposed in a similar manner with 'service' concepts. Five types of diagram are defined, as summarized in Fig. 16:

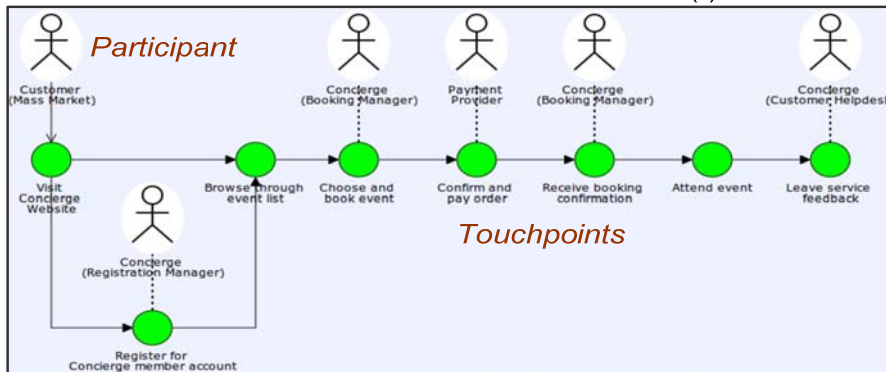


(a) Needs Model

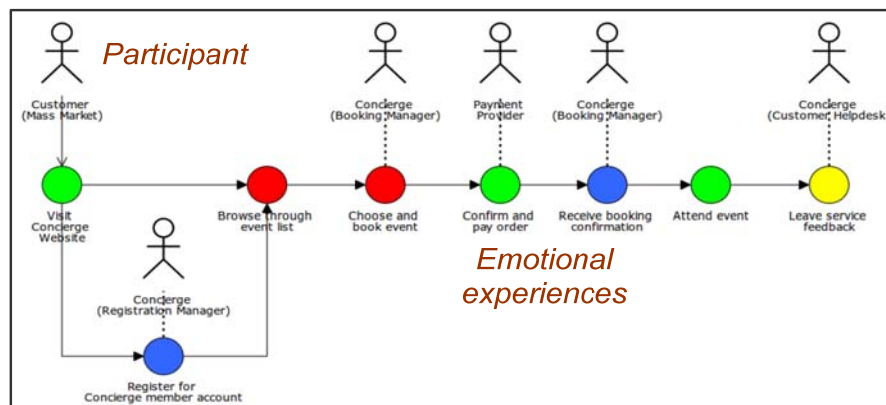


(b) Service Architecture

(c) Actor Network



(d) Service Journey Map



(e) Service Experience Journey Map

Fig. 16 Diagrams in ServiceML [22]

- *Needs model*, as a diagram relating customer needs with required service features.
- *Service Architecture*, as a simplification of SoaML's Service Architecture connecting participant and service.
- *Actor Network*, as a detailed version of Service Architecture with participant role and individual flow of sequential interaction.
- *Service Journey Map (SJM)*, as a graphical representation of a series of touch-points experienced by customer throughout the cycle of service provision.
- *Service Experience Journey Map*, a similar form of SJM with emotional colour-code representing expected customer experience.

Each of these diagrams has usage potential for the framework. The needs model may be used to relate service goals (artefact 1.2.2 in Fig. 1) and service features in business catalogue (artefact 1.2.3 in Fig. 1). Service architecture simplification, also introduced as part of b-SoaML [24], might replace SoaML's format for artefact 2.1.1 in Fig. 1, while the

details of Actor Network is more appropriate in the later stage, such as accompanying the Interaction model (artefact 2.1.2 in Fig. 1).

The actual interaction model is offered in the form of SJM. The diagram focuses on the touch-points from the consumer side and a suitable format for the interaction model (artefact 2.1.2). This interaction model describes a 'service path' [25], as a series of service encounters while at the same time reflect service states. The omission of supporting back end activities avoids a coverage redundancy with the process model.

SJML [23] share a similar abstraction with ServiceML's SJM, representing a series of touch points experienced by the customer. An example of SJML diagram is provided in Fig. 17. SJML enhances the touch point visualization by providing notations characterized by type of interaction, e.g. via telephone, email, website. A more recent version of SJML adopts the multi-participant feature of service interactions, as demonstrated in Fig. 18 [26]. SJML is therefore an important alternative to be adopted for interaction modelling.

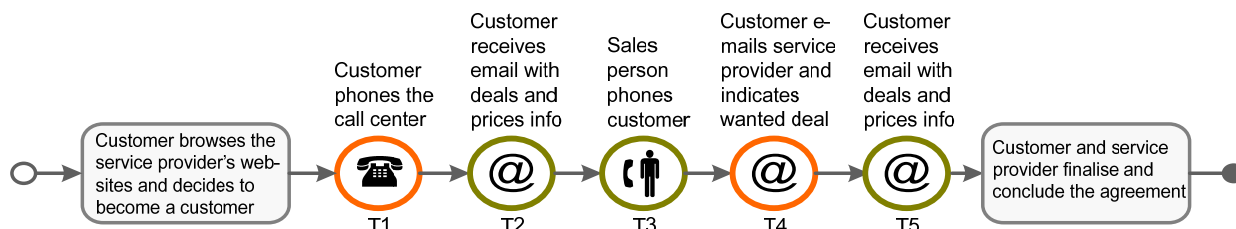


Fig. 17 Example of Service Journey Modelling Language (SJML) [23]

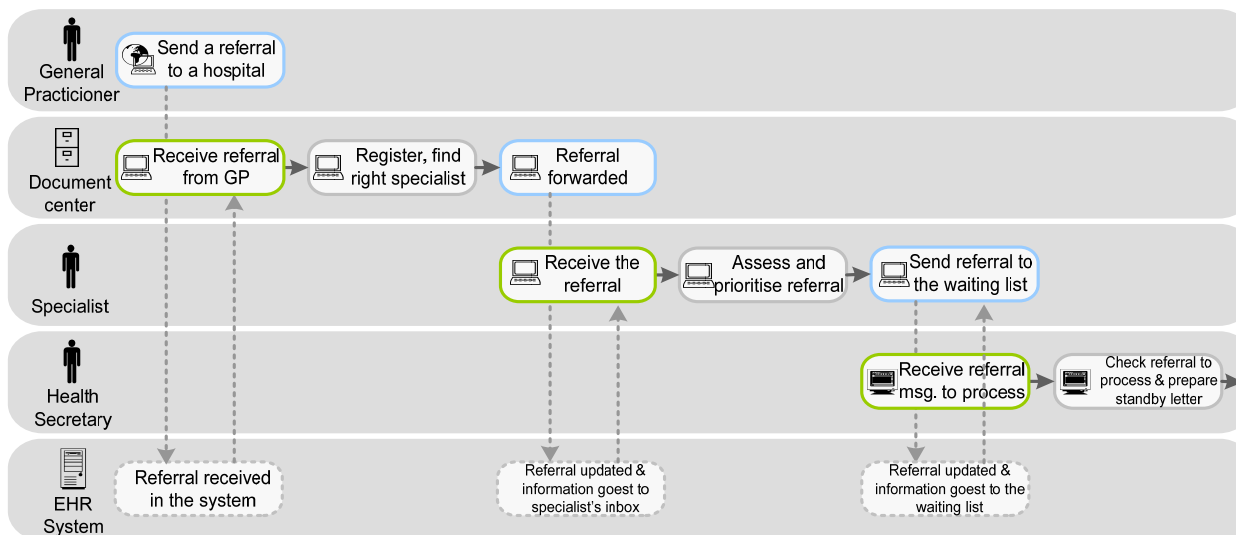


Fig. 18 SJML abstraction for multi-participant interaction [26]

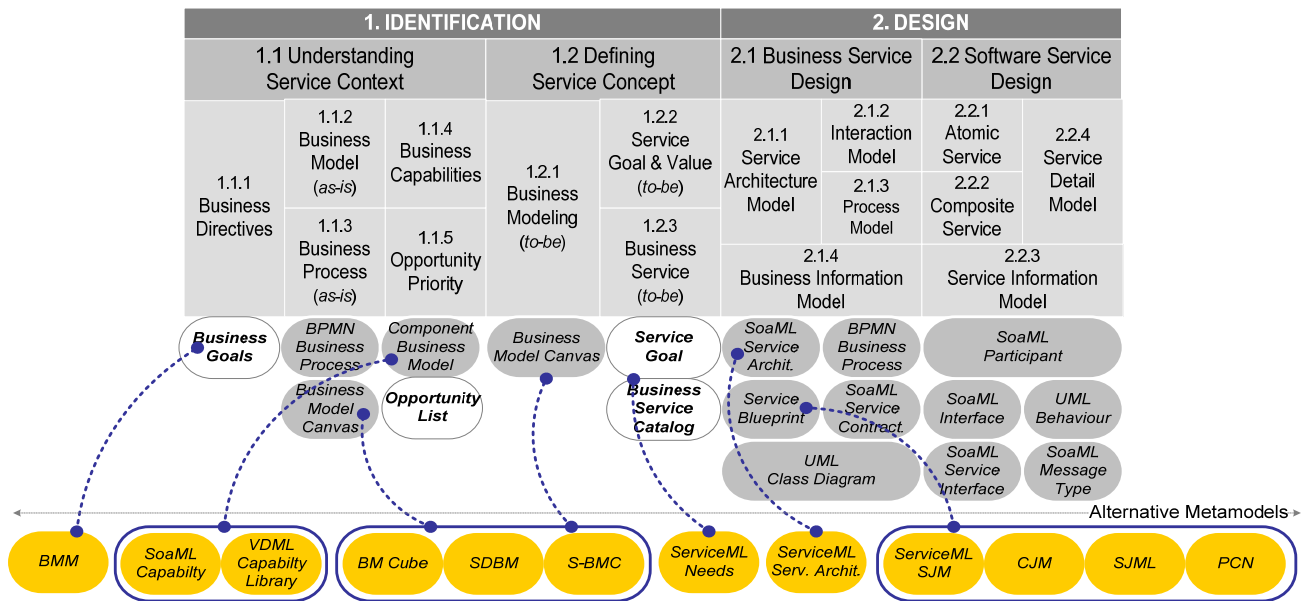


Fig. 19 Metamodel Options for Service Engineering Framework

An important feature of these new streams of interaction modelling is in its specification of touch point type, i.e. manual or software. It therefore might serve an important role in the framework; to differentiate between manual services and software services. Only services identified to be of the software type are required to be processed toward the software-service design (activity 2.2 in Fig. 1).

A summary of the additional format for artefacts is presented in Fig. 19. The proposed alternative metamodels are offered as a palette of options. The actual usefulness and usability of these alternatives still required further examination.

IV. CONCLUSION

This paper examines the GSEF in terms of the ontological coverage and the alternative metamodels for the artefacts. The ontological examination demonstrates a sufficiency of the framework in covering aspect of general service and software service.

This paper also describes an exploration toward metamodels as an artefact format in a service engineering analysis and design. Alternative metamodels are proposed as potential options. Further case studies are required to verify the usability of the metamodels in an actual project use.

In this sense, some metamodels can be considered to be a 'better' metamodel than others. The measuring criteria should be defined, which among others will be a trade-off between the expressive power (i.e. accuracy) and simplicity (i.e. comprehensibility) [27]. As a stack of metamodels used in a sequential process, the traceability and cross-translatibility features will also be important criterias.

Despite the recent demand for agility in the business landscape, a model is still expected to hold an important role in the analysis and design process [28]. Its usage pattern suggests the existence of a communication and collaboration

space where models hold a central role, such as envisioned by the Model Driven Engineering approach [29].

REFERENCES

- [1] Suhardi, R. Doss, and P. Yustianto, "Service Engineering Based on Service Oriented Architecture Methodology," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 13, no. 4, pp. 1466–1477, 2015.
- [2] P. Yustianto, R. Doss, and Suhardi, "Activities and Artifacts in Service Engineering Case Studies Report of Service Engineering Framework," *Inf. Technol. Syst. Innov. (ICITSI), 2017 Int. Conf.*, no. 1, pp. 373–377, 2017.
- [3] P. Yustianto, R. Doss, and Suhardi, "Consolidating service engineering perspectives," in *Information Technology Systems and Innovation (ICITSI), 2015 International Conference on*, 2015, pp. 1–7.
- [4] A. Osterwalder, Y. Pigneur, A. Smith, and T. Movement, "Business model canvas," *Self Publ. Last*, 2010.
- [5] OMG, "Business Process Model And Notation (BPMN) Specification," *Object Management Group (OMG) Specification*, 2014. (Online). Available: <https://www.omg.org/spec/BPMN>. (Accessed: 10-Apr-2018).
- [6] G. Pohle, P. Korsten, and S. Ramamurthy, "Component business models: Making specialization real," *IBM Inst. Bus. Value. August*, vol. 19, 2005.
- [7] OMG, "Service oriented architecture Modeling Language (SoaML) Specification," *Object Management Group (OMG) Specification*, 2012. (Online). Available: <http://www.omg.org/spec/SoaML/>. (Accessed: 07-Jun-2017).
- [8] M. J. Bitner, A. L. Ostrom, and F. N. Morgan, "Service blueprinting: a practical technique for service innovation," *Calif. Manage. Rev.*, vol. 50, no. 3, pp. 66–94, 2008.
- [9] J. Amsden, "Modeling with soaml, the service-oriented architecture modeling language," *IBM, Tech. Artic. Ser. January-February*, pp. 1–22, 2010.
- [10] T. Kohlbom, A. Korthaus, T. Chan, M. Rosemann., and M. Rosemann, "Identification and analysis of business and software services-a consolidated approach," *IEEE Trans. Serv. Comput.*, vol. 2, no. 1, pp. 50–64, 2009.
- [11] V. Bicer, S. Lamparter, Y. Sure, and A. H. Dogru, "Towards an interdisciplinary methodology for service-oriented system engineering," in *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*, 2009, pp. 486–491.
- [12] H. Karhunen, M. Jantti, and A. Eerola, "Service-oriented software engineering (SOSE) framework," in *Services Systems and Services Management, 2005. Proceedings of ICSSSM'05. 2005 International Conference on*, 2005, vol. 2, pp. 1199–1204.

- [13] F. Lin and P.-S. Hsieh, "A SAT view on new service development," *Serv. Sci.*, vol. 3, no. 2, pp. 141–157, 2011.
- [14] V. Hrgovic, W. Utz, and D. Karagiannis, "Service Modeling: A Model Based Approach for Business and IT Alignment," *2011 IEEE 35th Annu. Comput. Softw. Appl. Conf. Work.*, pp. 422–427, 2011.
- [15] M. Anjum and D. Budgen, "An investigation of modelling and design for software service applications," *PLoS One*, vol. 12, no. 5, p. e0176936, 2017.
- [16] OMG, "Business Motivation Model (BMM)," *Object Management Group (OMG) Specification*, 2015. (Online). Available: <http://www.omg.org/spec/BMM/>. (Accessed: 07-Jun-2017).
- [17] OMG, "Value Delivery Modeling Language (VDML)," *Object Management Group (OMG) Specification*, 2015. (Online). Available: <http://www.omg.org/spec/VDML/>. (Accessed: 08-Jun-2017).
- [18] R. Tönisson and R. Matulevičius, "A Coarse-Grained Comparison of Modelling Languages for Business Motivation and Intentional Distribution," in *International Conference on Business Informatics Research*, 2016, pp. 80–95.
- [19] E. Lüftenegger, *Service-dominant business design*, vol. 402, no. January. Eindhoven University of Technology, 2014.
- [20] A. Zolnowski, C. Weiß, and T. Bohmann, "Representing Service Business Models with the Service Business Model Canvas--The Case of a Mobile Payment Service in the Retail Industry," in *system sciences (HICSS), 2014 47th Hawaii International Conference on*, 2014, pp. 718–727.
- [21] S. E. Sampson, "Visualizing service operations," *J. Serv. Res.*, vol. 15, no. 2, pp. 182–198, 2012.
- [22] A. J. Berre, H. de Man, Y. Lew, B. Elvesæter, and B. M. Ursin-Holm, "Open business model, process and service innovation with VDML and ServiceML," in *Enterprise Interoperability: Research and Applications in Service-oriented Ecosystem (Proceedings of the 5th International IFIP Working Conference IWIE 2013)*, 2013, vol. X, no. X, p. 127.
- [23] R. Halvorsrud, E. Lee, I. M. Haugstveit, and A. Følstad, "Components of a visual language for service design," in *ServDes. 2014 Service Future; Proceedings of the fourth Service Design and Service Innovation Conference; Lancaster University; United Kingdom; 9-11 April 2014*, 2014, no. 99, pp. 291–300.
- [24] T. Chang, A. J. Berre, C. Carrez, and B. Elvesæter, "Business-soaml: Service identification and specification from a business perspective," in *Enterprise Interoperability V*, Springer, 2012, pp. 379–389.
- [25] R. G. Qiu, *Service science: The foundations of service engineering and management*. John Wiley & Sons, 2014.
- [26] E. Lee, A. Karahasanovic, and R. Halvorsrud, "A visual language for the modelling of service delivery processes to support business processes management," *Int. J. Adv. Softw.*, vol. 8, no. 3 & 4, pp. 288–308, 2015.
- [27] B. Selic, "The pragmatics of model-driven development," *IEEE Softw.*, vol. 20, no. 5, pp. 19–25, 2003.
- [28] Z. Babar, A. Lapouchnian, and E. Yu, "Modeling DevOps deployment choices using process architecture design dimensions," in *IFIP Working Conference On The Practice of Enterprise Modeling*, 2015, vol. 2, pp. 322–337.
- [29] D. Gasevic, D. Djuric, and V. Devedzic, *Model Driven Engineering and Ontology Development*, vol. XXI. 2009.