ITIL Metamodel

Nelson Gama[1,2,3,*], Marco Vicente[2], and Miguel Mira da Silva[1,2]

[1]Instituto Superior Tecnico, Lisboa, Portugal
[2]INOV, Lisboa, Portugal
{nelsongama,marco.vicente,mms}@ist.utl.pt
[3]CINAV-PT Navy Research Center, Escola Naval, Portugal
nelson.gama@defesa.pt

Abstract. IT Infrastructure Library (ITIL) has become the *de facto* standard for IT Service Management (ITSM). Despite the advantages in the adoption of ITIL's best practices, some problems have been identified: different interpretations due to the complexity of concepts with poor specification and formalization; different approaches to the same problems; difficulties exchanging process models in different process model languages. Besides all published work, is still missing a metamodel expressing the core concepts, their relationship, and constraints. In this paper, we propose an ITIL metamodel to reduce conceptual and terminological ambiguity, addressing the identified problems, namely: (1) describing the core concepts of ITIL to be used by other approaches; (2) allowing the integration, exchange, sharing and reutilization of models; and (3) the use of different modelling languages following the defined principles.

Keywords: ITIL, metamodel, modelling.

1 Introduction

Abstract. IT Infrastructure Library (ITIL) has becoming the de facto standard, currently the most widely accepted framework in the world, for implementing IT Service Management (ITSM) [1-3].

Despite the advantages in the adoption of ITIL's best practices, many organizations follow ITIL's best practices without a reference model and some problems have been identified: (1) the complexity of ITIL concepts with poor specification and formalization, which leads to misunderstandings about these concepts; (2) different tools and methodologies that are not harmonized or grounded in a shared reference model leading to different approaches to the same problems; (3) the exchange of process models in different process model languages still remains a challenge [4].

Besides all published work and books about ITIL, a metamodel expressing the core concepts, the relation between them, their constraints and limitations is still missing, especially with academic support.

Once a model is an instance of a metamodel, an ITIL metamodel will be a model to shape ITIL. A metamodel of ITIL as an explicit model of constructs and rules,

* Corresponding Author.

X. Franch et al. (Eds.): ICSOC 2014, LNCS 8831, pp. 486–493, 2014.

defining logical structures and generating semantics is warranted to specify models within this defined domain of interest.

In this paper, we propose an ITIL metamodel addressing the identified problems, namely: (1) describing the core concepts of ITIL so as to be used by other approaches; (2) allowing the integration, exchange, and reutilization of models developed based on the proposed metamodel; and (3) the use of different modelling languages following the defined principles supported by the metamodel.

2 Related Work

There are a few academic or professional publications concerning the conceptual modelling or metamodels of IT services. They are mostly process-oriented describing how to generalize service processes into universal patterns or conceptual models [5].

To define an ITIL metamodel, firstly we analysed the ontological constructs of ITIL's model definition from ITIL books, identifying modelling limitations. In general, the concepts and processes, as well as necessary databases and interfaces, are defined at a high abstraction level [5]. Only a linguistic description of concepts is provided [1], while its processes are usually depicted as sequences of activities.

We evaluated several ITIL graphical representations searching for the best approach to represent and relate concepts. We found disparate ad-hoc diagrams, different approaches and notations from distinct organizations. These were mainly in-house sketches, diagrams and flowcharts expressing the ITIL views of its authors. ITIL representations are often depicted using BPMN [6] but some have also come across proprietary commercial solutions. In common, all ITIL representations come from one author's interpretation of concepts and relationships from a linguistic description in ITIL books. We acknowledge the added value of these models and tools and we are not claiming they are incorrect, but instead we pointing out their lack of completeness and the absence of a common referential – a metamodel providing a uniform basis in terms of concepts and their relationships.

Afterwards we evaluated some architectural frameworks (MDA core standards [7], and MOF specification [8]) to develop a model and metamodels. Despite valuable, these approaches are software development oriented. A rigid perception of "layered metamodel architecture" may be confused and limitative when considering metamodels to a widespread use [5].

There are many publications and research strands on process metamodel and we also made an overview of published work in this area [4, 5, 9, 10]. However, research work or professional publications regarding the definition of an ITIL metamodel is quite limited. Most of them focused on the previous ITIL version (v2) and are mostly process-oriented, stressing few processes. On the other hand, we did not find a holistic description on how to generalize service lifecycle and neither defining universal patterns nor conceptual models for ITIL's concepts.

Despite all valuable work regarding the definition of an ITIL metamodel we did not identify a proposal focused on the current version of ITIL neither covering the principal ITIL concepts that allows defining a metamodel. An ITIL metamodel as a

description of a language's abstract syntax in order to define a set of constructs that allows the creation of grammatically valid models [11].

3 Research Problem

ITIL processes, concepts, and relationships are specified in natural language. Without a formal and commonly accepted semantics, modelling graphical representation is complex [10]. In addition to the aforementioned problems in ITIL adoption, we identified some weaknesses in ITIL representation: (1) unclear concepts definition leading to different interpretations; (2) models developed from a language description and not from a universal referential; (3) lack in formal notation and representation leading to loosely depicted graphical diagrams; (4) focus on logical description of processes; (5) different approaches and methodologies to the same problems, making exchange and knowledge sharing difficult; (6) lack of holistic visibility and traceability from the theory; and (7) different approaches to implementations and tools development.

A metamodel of ITIL, as an explicit model of constructs and rules, is still needed to specify models within this defined domain of interest. The most important contribution of an ITIL metamodel would be the convergence of approaches and applications of leading vendors and motion towards the ITIL compliant solutions [5].

Metamodels are also closely related to ontologies. Both are often used to describe relations between concepts [12], allowing us to understand the logical structures and generating semantics for best practice frameworks [13]. We acknowledge the difference between ontologies and metamodels, once their characteristics and goals are different. However, without an ontology, different knowledge representations of the same domain can be incompatible even when using the same metamodel for their implementation [14]. While an ontology is descriptive and belongs to the domain of the problem, a metamodel is prescriptive and belongs to the domain of the solution [15]. Ontologies provide the semantics, while metamodels provide a visual interpretation of the syntax of specific languages that approximate as closely as possible to the ideal representation of the domain [16]. As a semantic model, the relation to the reality and the interrelation of concepts are true if they obey to a mathematical structure for all axioms and derivation rules of the structure [17]. To the best of our knowledge, there is no universally accepted ITIL metamodel as a reference that allows the modelling development and a language basis for graphical representation.

4 Proposal

Following some previous published work [4, 5, 13, 18, 19] we considered two separate orthogonal dimensions of metamodelling: one dimension concerned with language definition and the other with representation. A language can be defined as a set of valid combinations of symbols and rules that constitute the language's syntax [11].

Firstly, we identified the core concepts from the ITIL glossary [20], by reducing the concepts to the fundamental ones, with representation needs, that should be part of the metamodel. To that aim we followed an ontology engineering process [19]

and analysed ITIL domain, clarifying abstract concepts from ITIL' books specifications developing the proposed metamodel. Secondly, we defined linguistically all concepts for an ontological common understanding adding a mathematic representation to concepts and relationships. This clarification provided design rules and a modelling process, decreasing the concept's abstraction but allowing the fundamental distinction of the concepts relation and interoperability. In a next step, we defined the notation, clarifying the ontological metamodel and, thus, the metamodel. The metamodel's quality was validated through the guidelines of modelling defined by Schütte [17].

4.1 Concepts Identification and Characterization

As stated before, we used an ontology-engineering methodology [19] to identify concepts and relationships from ITIL's five books and through the collaboration of ITIL practitioners. We started by defining the domain, the terms, their properties, and purposes. Having identified terms, we created the concepts and determined their relationships, providing the ontological vocabulary and symbols used to define a domain's problems and solutions [21, 22].

Since the subjectivity of modelling cannot be eliminated (only managed), a demand for rules that have to be carried out in the modelling process should be found. Therefore, before developing the metamodel we defined a basis for the development of a language (metamodelling language capability), identifying the core concepts by description and, especially, by definition in order to outline the abstract syntax of a modelling language as follows in Table 1. The language definition (Table 1) clarifies concepts and provides an ontological clarification through the definition of linguistic concepts. This proposed metamodel allows a hierarchy of metamodel levels where each level is "an instance" of the level above. We may have concepts' extensions that allow multilevel instantiations establishing its own kind of metalevel definitions.

Table 1. ITIL metamodel core concepts - modelling language

Concept	Description	Definition
Service Portfolio	The service portfolio S represents the current complete set of services. S is a key element and it is used to manage the entire lifecycle of each service $s_i \in S$. It includes three categories: (i) Service Pipeline P with $P \subset S$ (proposed or in development); (ii) Service Catalog C with $C \subset S$ (live or available for deployment); and (iii) Retired Services R with $R \subset S$.	$S = \{s_1, s_2, \ldots, s_n\}$ $S = P \cup C \cup R$ $\exists\, s_i \in S : s_i \in P \vee s_i \in C \vee s_i \in R$
Service	A Service s_i definition covers from an ITIL book to an IT service. Both cases can share the same metamodel despite having different levels. A Service s_i is a vector performed by a Role ρ as defined as a Contract c_i and in function of a Process Π	$s_i \in C$ $s_i = (\rho, f(\Pi), c_i)$
Process	A Process π_i from the set of Processes Π is a structured set of Activities Δ designed to conduct a Service s_i under a defined Role ρ and triggered by an Event ε^i and delivering an Event ε^o.	$\pi_i \in \Pi$ $\pi_i = (f(\Delta), \varepsilon^i, \varepsilon^o, \rho)$
Activity	A set of actions a_{ij} designed to achieve a particular result in a Process π_i. An Activity Δ can have one or "n" actions a_{ij}.	$\Delta = \sum_{i=1}^{n} \alpha_{ij}$ $1 \leq \Delta \leq n$

Table 1. (*Continued*)

Action	*An atom activity α_{ij} with defined procedure from a set A_{ij} of possible actions. Actions can be performed by a Stakeholder or automatically by an Application Service as_i*	$A_i = \{\alpha_{i1}, \alpha_{i2}, \ldots \alpha_{in}\}$ $\alpha_{i1} = f(as_i)$
Role	A role ρ from a set of roles R is defined as a specific behavior of a stakeholder σ (a business actor) participating in a defining a set of responsibilities in a Process π or in a Service s_i	$\rho_i = f(\sigma_i)$ $\forall x \in R \; \exists \sigma \in \Sigma$: $\rho(x) \in R$
Stakeholder	A stakeholder σ represents a person in a set of Stakeholders Σ with a defined Role ρ at a given moment t.	$\sigma_t = (\sigma_{1t}, \sigma_{2t}, \ldots, \sigma_{nt}) \in \Sigma_t$ $\forall x$: $\sigma(x) \rightarrow \rho(x)$
Contract	A compromise c_i between two or more parties.	c_i = {OLA, SLA, Agreement, UC}
Event	A set ξ of events e_i triggering a Process π_i as input of a Process or e_o produced as output, referring any change of state or anything that happens (internally or externally).	$\xi = \{e_{i1}, e_{i2}, \ldots e_{in}\} \cup \{e_{o1}, e_{o2}, \ldots e_{on}\}$ $\forall e_{i,o}: \; e_{i,o} \in \xi$
Application Service	An externally visible unit of software functionality as_i, provided by one or more components, exposed through well-defined interfaces, and meaningful to the environment [23]	$as_i = f(af_i)$ $as_i \in AS_i$
Application Function	A software component afi that provides Functions through an Application Service as_i required by an Action α_i. Each Application may be part of more than one IT Service. An Application runs on one or more Infrastructure Service $f(\sigma_{ij})$	$af_i = f(\sigma_{ij})$ $af_i \in AF_i$
Infrastructure Service	Externally visible unit of Infrastructure functionality σ_{ij}, from the overall organization's technologic infrastructure I_{ij} provided by one or more Infrastructure Function, exposed as service to the Application Function.	$I_{ij} = \sum_{i,j}^{n} \sigma_{ij}$, $\sigma ij \in Iij$

4.2 Metamodel Representation

We mapped the ITIL concepts in the language's metamodel. The proposed ITIL metamodel formalizes expressiveness through the definition of concepts and corresponding visual representation as the following graphical representation (Fig. 1). The proposed ITIL metamodel is based on the structure illustrated in Fig. 1, which relies on concepts presented in Table 1.

5 Demonstration and Evaluation

To demonstrate the widespread use of the proposed ITIL metamodel, we modelled several models of ITIL [24, 25] (not included due to paper size restrictions) and used the ITIL metamodel to model ITIL with ArchiMate [26] in the datacentre organization of the Portuguese Defence Ministry.

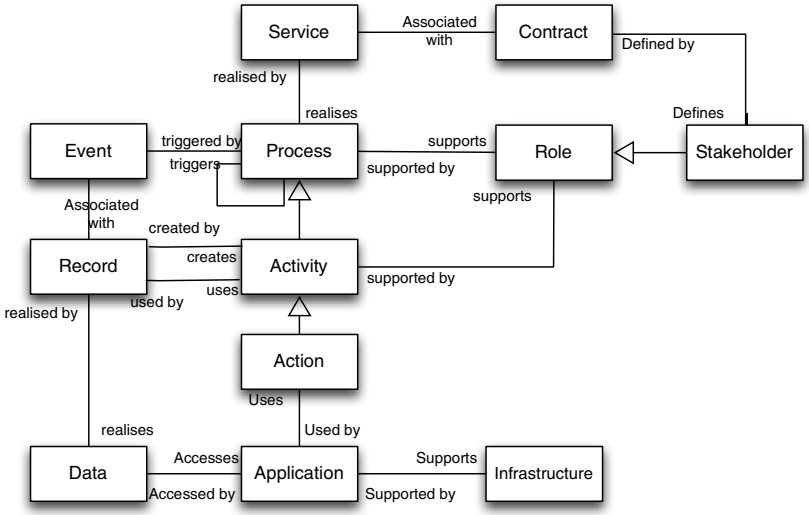

Fig. 1. Proposed ITIL Metamodel

We used the ArchiMate's [26] notation to graphically represent the metamodel for no other reason than making it easy to use, but we may represent the metamodel in any other notation. This generalization makes it possible to model in different languages, and the integration and reuse of models.

We have modelled an overview of all ITIL's five books [24, 25] to understand which services (and from which books) ITIL can provide to its external environment. We have also modelled each ITIL book, showing which are the applications ITIL uses to support its processes, and also the infrastructure components that support those applications. It provides a top view, having ITIL core processes as a black box system providing services to the environment while using all the ITIL processes. We modelled each one of ITIL's processes showing a deeper fine-grained representation, which allow us to look inside the ITIL's processes and see all of its individual activities. These models are consistent, since the processes' inputs and outputs, business, application and infrastructure services are the same, although at different granularity levels. We also mapped the activity sequence of ITIL's Incident Management process from two different notations (ArchiMate [26] and BPMN [6]), which matched almost completely. We realized that it would be harder to integrate two approaches if they did not speak the same language. Therefore, a common frame of reference provided by the ITIL metamodel is warranted. Even in the absence of a formal graphical language we are able to model ITIL using the proposed metamodel.

For the purpose of our research, a high-level checking of utility, correctness, consistency and completeness of ITIL metamodel has been performed. Schu□tte [17] defines guidelines to metamodel's quality evaluation, which are very similar to the set of design criteria for ontologies [15]: clarity, coherence, extendibility, minimal encoding bias, and minimal ontological commitment.

To evaluation purposes, we interviewed practitioners from different areas, skills and nationalities, all with a strong ITIL background. This evaluation allow us to ask open-ended questions exploring emotions, experiences or feelings that cannot be easily observed or described via pre-defined questionnaire responses [27]. We concluded that the models from ITIL metamodel would benefit ITIL implementation.

6 Conclusion

The understanding of ITIL's concepts and relationships from ITIL referential books is hard and requires a lot of time and effort. Different organizations and service providers develop their own models regarding ITIL adoption without a metamodel or a common referential, making difficult to share and communicate ITIL models between different stakeholders. We developed an ITIL metamodel providing an academic contribution to this area, which was not available by the time we started this research.

An ITIL metamodel is per se a valuable contribution. However, the main contribution of this proposal lies in defining a metamodel to help the ITIL adopters with a universal identification of concepts and relationships among them, independently of approach, language or tool used. We identified the core concepts of ITIL's service lifecycle and the relationship among them, proposing an ITIL metamodel. Our approach keeps the semantics of the core concepts intact and thus allows for the reuse of models and reasoning over the customized metamodel.

Our proposed metamodel might represent a basis to model and to implement ITIL. Moreover, it provides the sharing and the reutilization of the models from one approach to another, even with different modelling languages, to improve the representation of ITIL concepts, and to help promote ITIL discussion and validation within the ITIL community itself.

References

1. Hochstein, A., Zarnekow, R., Brenner, W.: ITIL as Common Practice Reference Model for IT Service Management: Formal Assessment and Implications for Practice. In: International Conference on e-Technology, e-Commerce and e-Service (EEE 2005), pp. 704–710. IEEE Computer Society (2005)
2. Correia, A., Abreu, F.B.E.: Integrating IT Service Management within the Enterprise Architecture. In: 4th International Conference on Software Engineering Advances (ICSEA), pp. 553–558. IEEE, Porto (2009)
3. Gama, N., Sousa, P., Mira da Silva, M.: Integrating Enterprise Architecture and IT Service Management. In: 21st International Conference on Information Systems Development (ISD 2012), Springer, Prato (2012)
4. Shen, B., Huang, X., Zhou, K., Tang, W.: Engineering Adaptive IT Service Support Processes Using Meta-modeling Technologies. In: Münch, J., Yang, Y., Schäfer, W. (eds.) ICSP 2010. LNCS, vol. 6195, pp. 200–210. Springer, Heidelberg (2010)
5. Strahonja, V.: Definition Metamodel of ITIL. Information Systems Development Challenges in Practice, Theory, and Education 2, 1081–1092 (2009)
6. Object Management Group: Business Process Model and Notation (BPMN). V 2.0 (2011)
7. OMG: MDA Guide Version 1.0. The Object Management Group (OMG) (2003)

8. OMG: MetaObject Facility (MOF) 2.0 Core Specification Version 2.4.1. OMG Adopted Specification. The Object Management Group (OMG) (2003)
9. Jantti, M., Eerola, A.: A Conceptual Model of IT Service Problem Management. In: International Conference on Service Systems and Service Management (ISSSM 2006), Troyes, France, vol. 1, pp. 798–803 (2006)
10. Valiente, M.-C., Garcia-Barriocanal, E., Sicilia, M.-A.: Applying an Ontology Approach to IT Service Management for Business-IT Integration. Knowledge-Based Systems 28, 76–87 (2012)
11. Guizzardi, G.: On Ontology, ontologies, Conceptualizations, Modeling Languages. In: Vasilecas, O., Ede, J., Caplinskas, A. (eds.) Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV, pp. 18–39. IOS Press (2007)
12. Söderström, E., Andersson, B., Johannesson, P., Perjons, E., Wangler, B.: Towards a Framework for Comparing Process Modelling Languages. In: Pidduck, A.B., Mylopoulos, J., Woo, C.C., Ozsu, M.T. (eds.) CAiSE 2002. LNCS, vol. 2348, pp. 600–611. Springer, Heidelberg (2002)
13. Neto, A.N.F., Neto, J.S.: Metamodels of Information Technology Best Practices Frameworks. Journal of Information Systems and Technology Management (JISTEM) 8, 619–640 (2011)
14. Calero, C., Ruiz, F., Piattini, M.: Ontologies for Software Engineering and Software Technology. Springer (2006)
15. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies 43, 907–928 (1995)
16. Baioco, G., Costa, A., Calvi, C., Garcia, A.: IT Service Management and Governance Modeling an ITSM Configuration Process: A Foundational Ontology Approach. In: International Symposium on Integrated Network Management-Workshops (IM 2009) IFIP/IEEE, New York, pp. 24–33 (2009)
17. Schuette, R., Rotthowe, T.: The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models. In: Ling, T.-W., Ram, S., Li Lee, M. (eds.) ER 1998. LNCS, vol. 1507, pp. 240–254. Springer, Heidelberg (1998)
18. Atkinson, C., Kühne, T.: Model-Driven Development: A Metamodeling Foundation. IEEE Software 20, 36–41 (2003)
19. Ostrowski, L., Helfert, M., Xie, S.: A Conceptual Framework to Construct an Artefact for Meta-Abstract Design. In: Sprague, R. (ed.) 45th Hawaii International Conference on System Sciences (HICSS), pp. 4074–4081. IEEE, Maui (2012)
20. OGC: ITIL Glossary of Terms, Definitions and Acronyms (2007)
21. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
22. Vaishnavi, V.K., William Kuechler, J.: Design Science Research Methods and Patterns: Innovating Information and Communication Technology. Auerbach Publications, Boston (2007)
23. Lankhorst, M.: Enterprise Architecture at Work. Springer (2009)
24. Vicente, M., Gama, N., Mira da Silva, M.: Using ArchiMate to Represent ITIL Metamodel. In: 15th IEEE Conference on Business Informatics (CBI), IEEE (2013)
25. Vicente, M., Gama, N., da Silva, M.M.: Using archiMate and TOGAF to understand the enterprise architecture and ITIL relationship. In: Franch, X., Soffer, P. (eds.) CAiSE Workshops 2013. LNBIP, vol. 148, pp. 134–145. Springer, Heidelberg (2013)
26. The Open Group: ArchiMate 2.0 Specification. Van Haren Publishing (2012)
27. Oates, B.J.: Researching Information Systems and Computing. Sage Publications (2006)