In Proceedings of the 2nd Int'l Workshop on Service-Oriented Computing: Consequences for Engineering Requirements (SOCCER'06),
14th International Requirements Engineering Conference (RE'06),
September 11-15, 2006 Minneapolis/St. Paul, Minnesota, USA.

Process Artifacts Defined as an Aspectual Service to System Models

Robin A. Gandhi, Siddharth J. Wagle, Seok-Won Lee
Knowledge-intensive Software Engineering Research Group
Dept. of Software and Information Systems, The University of North Carolina at Charlotte
Charlotte, NC 28223-0001, USA. {rgandhi, sjwagle, seoklee}@uncc.edu

Abstract

Process artifacts identified from a process description often implicitly bias and cross-cut the definition of generic services from various tools that assist/automate process activities. The resulting tool-support is tightly coupled with the process definition it supports, leading to poor adaptability when the required artifacts or process activities evolve/change. This issue is of further concern while providing tool-support for assisting knowledge-intensive process activities through an interactive exploration of related knowledge-bases. Therefore, our focus is on early separation of process related cross-cutting concerns from generic tool-support services for creating, browsing, accessing, querying, inferencing, and visualizing associated knowledge-bases. We discuss our approach in the context of designing tool support for a system security Certification and Accreditation (C&A) process automation based on service-oriented and aspect-oriented design paradigms.

1. Introduction

Process is an abstract description of a series of activities that produce artifacts for satisfying real-world goals/objectives. As the service-oriented design paradigm gains momentum, process activities are being increasingly supported through combinations of services that support common operations on related information repositories. However, the activities and related artifacts of a process description implicitly bias and cross-cut the definition of generic support services, tightly coupling them to the given process. These issues are further complicated while defining services for assisting knowledge-intensive activities that require significant interaction of human experts with the process artifacts available from diverse dimensions at different levels of abstraction in associated knowledge-bases.

In the context of knowledge-intensive C&A process activities for establishing software assurance, each target information system requires careful adaptation and tailoring of the assurance process and related artifacts based on non-trivial analysis and negotiations among related stakeholders. Most often C&A process activities for software assurance are tailored according to the organization; agency/site specific policies; system lifecycle status; program strategy; and information classification [3]. In turn, each C&A process activity requires artifacts that are relevant to, and reveal the behavior of the target system in its problem domain. C&A process evolution/improvement is also continuously motivated by factors such as the ever increasing complexities of target systems; changes in the perceived types and levels of threats; or the stakeholders understanding of various threats improves over time.

In our efforts [8] for supporting activities of the Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP) [3], we have applied the Ontology-based Active Requirements Engineering (Onto-ActRE) [6] framework to elicit, model and analyze the artifacts related to the target system and its problem domain. The framework combines the strengths of multiple complementary Requirements Engineering (RE) modeling philosophies in a unifying ontological frame representation. The operations for editing, browsing, accessing, querying, inferring, and visualizing the ontological system models in associated knowledge-bases are defined as generic *Knowledge Services* following the service-oriented design paradigm.

We intend to combine these knowledge services to support the knowledge-intensive activities of the DITSCAP through an integrated automation tool. However, the need to adapt DITSCAP according to the characteristics of each target system and problem domain poses unique challenges in defining the related tool-support workflow and its integration with required knowledge services. Additionally, given the nature of security problem domain, the DITSCAP activities require continuous evolution/changes to keep up with the ever-changing and emerging forms of threats to complex software systems.

To address these concerns, we present our position on developing a human and machine understandable definition of process activities and their required artifacts and expose it an *aspectual service* that guides the composition of generic knowledge services for establishing tool-support workflow. In essence, the process definition along with other related cross-cutting concerns are separated as ontological representations during the RE process for providing tool-support. Such ontological process definition will provide the ability to dynamically tailor tool-support workflow and increase process comprehension.

The rest of the paper is organized as follows. Section 2 provides a brief introduction to our DITSCAP automation efforts and the related DITSCAP-Automation Tool (DITSCAP-AT). In Section 3 we elaborate on our approach to elicit and represent requirements for a process as an ontological definition exposed through an aspectual service. Section 4 presents a conceptual DITSCAP-AT architecture that combines service-oriented and aspect-oriented design paradigms. Finally, we conclude in Section 5 with some future research directions.

2. DITSCAP Automation Background

The key roles involved in the DITSCAP are the program manager, DAA, certifier, and the user representative that tailor and scope the C&A efforts to the particular mission, environment, system architecture, threats, funding and schedule of the system through negotiations. Once the system definition has been agreed upon by the key roles, it is documented and becomes the Software Security Authorization Agreement (SSAA). The SSAA aggregates all process artifacts produced by performing DITSCAP activities for the target system.

Nevertheless, DITSCAP is a knowledge-intensive process requiring information from large and diverse sources to be interpreted, recalled, and analyzed. DITSCAP being a primarily requirements-driven approach, the need to systematically capture and organize problem domain concepts related to DITSCAP security requirements is apparent for efficiently analyzing its applicability, and implementation effectiveness. Through our DITSCAP automation efforts we have produced hierarchical models of ontological concepts that capture well-defined dimensions of the problem domain with related properties and non-taxonomic dependencies among them [5]. Theoretical foundations behind our approach are based on the Onto-ActRE framework [6].

The resulting DITSCAP Problem Domain Ontology (PDO) includes structured and well defined representations of: 1) A Requirements Domain Model that hierarchically organizes requirements categories with leaf-node security requirements extracted from DITSCAP-oriented regulatory documents; 2) A viewpoints hierarchy that captures different perspectives and related stakeholders of a security requirement; 3) A risk assessment taxonomy that gathers risk factors from a broad spectrum of perceived risk sources in the DITSCAP domain; 4) A DITSCAP Goal hierarchy that captures the rationale behind enforced security requirements; 5) Meta-knowledge about information learned from network discovery/monitoring tools; and 6) Interdependencies between various concepts in the PDO. Further details about these models can be found in [4].

Throughout the DITSCAP, the hierarchical organization of PDO concepts provide placeholders to capture information regarding the target system from users, operating manuals, plans, architecture diagrams, and automated network-based information discovery toolkits at various levels of abstraction. Essentially, a comprehensive collection of well-categorized process artifacts gathered from multiple dimensions, at different levels of abstraction and different stages of the target system lifecycle becomes available which can be reused, independent of DITSCAP, across other software assurance initiatives.

To support the representation of rich knowledge structures required by the PDO, various ontological engineering processes are provided by the GENeric Object Model (GenOM) [7] toolkit. GenOM inherits the theoretical foundation of the frame representation and is compatible with the Open Knowledge Based Connectivity (OKBC) specification [1] as well as the Web Ontology Language (OWL) representation [9] format. Building upon on generic APIs offered by GenOM, we have developed specialized *service facades* to define knowledge services for editing, browsing, accessing, querying, inferencing, and visualizing a specific knowledge model in the PDO. A conceptual overview of these knowledge services is shown in Figure 1.

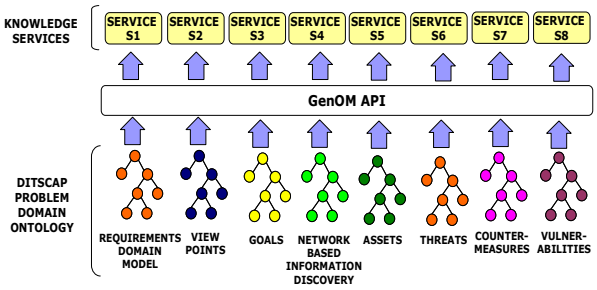

Figure 1: DITSCAP-AT Knowledge Services

3. The DITSCAP-AT Process Definition

As various process artifacts become available in the DITSCAP PDO, the goal of DITSCAP-AT (tool-support) is to expose relevant artifacts during target system data collection and analysis based on the DITSCAP activity definitions. We now discuss how the requirements elicitation activities for the DITSCAP-AT workflow lead to the creation of hierarchical ontological models. Exposing such ontological process definition as a service will guide the composition of generic knowledge services for establishing the DITSCAP-AT workflow.

As a first step, we identify core components of the DITSCAP to produce a process-driven workflow that is applicable across all types of target systems. These process components provide a level of abstraction to aggregate activities for tailoring the DITSCAP-AT based on unique characteristics of each target system. A representative set of these process components is depicted in Figure 2.

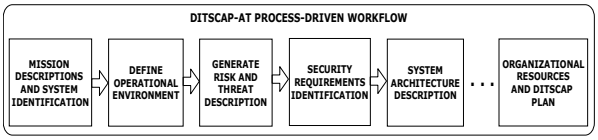

Figure 2: Process-Driven Workflow Components

The second step involves a goal-driven decomposition of each abstract process-driven workflow component in to specific activities that satisfice the higher level goals of DITSCAP. For each specific activity we identify the required process artifacts along with the corresponding knowledge services that make them accessible for analysis.

3.1. Elicitation of Process Requirements

The DITSCAP application manual [2]; Knowledge acquisition sessions with Subject Matter Experts (SME) who perform knowledge-intensive activities over C&A data; and stakeholder negotiations for establishing the certification effort are the primary sources to tailor the DITSACP-AT process workflow. Following RE techniques for goal-driven requirements elaboration [12], we elaborate on our approach using an example of the "Generate Risk and Threat Description" DITSCAP-AT process component as shown in Figure 3.

Based on our approach, the high level process components are gradually operationalized through specific activities that utilize knowledge services shown in Figure 1. Goals of the high level process components are refined based on the agreed upon definition among stakeholders of the DITSCAP. As more refined goals are available, knowledge-intensive activities that operationalize these leaf-node goals are elicited from the DITSCAP application manual in collaboration with the field practices of SMEs. Finally, each activity based on its needs, is assigned the required combinations of *knowledge services* for creating, browsing, accessing, querying, inferring, and visualizing associated system models. Annotation of each activity with required *knowledge services* in Figure 3 clearly depicts the interactions of DITSCAP-AT workflow with multiple system models to satisfy the higher level C&A goals. These *knowledge services* correspond to those shown in Figure 1.

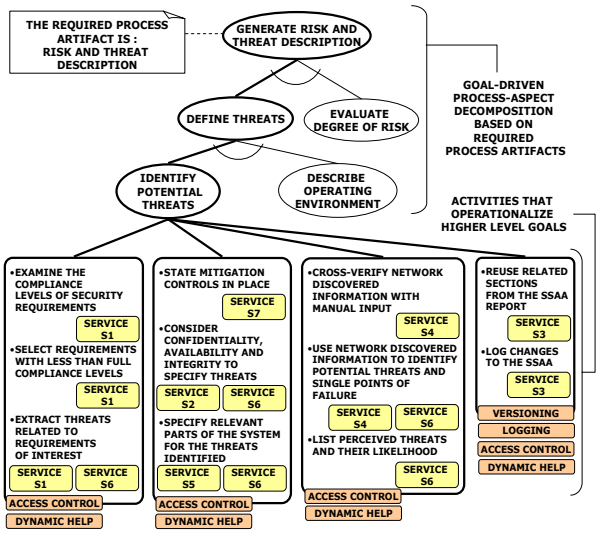

Figure 3: Partial Goal-driven Process Component Operationalization

Goal-driven process component operationalization also facilitates the discovery of leverage points in the process definition for common operations such as access control; search for related concepts; logging and recovery of changes; and versioning. For example, context-sensitive user guidance is a feature that will be frequently required for various the knowledge-intensive activities supported by DITSCAP-AT. The annotation of specific activities with such cross-cutting concerns, as shown in Figure 3, helps to identify other candidate aspectual services.

3.2. Representation of Process Requirements

We use GenOM to represent the goal-driven process decomposition models that results from elicitation efforts in the previous sub-section based on ontological engineering techniques. An ontology-based process definition will allow for process tailoring using high-level semantics that are closer to the real-world goals and objectives of the C&A process. Building upon on APIs offered by GenOM, the service facade that exposes the ontological process definition will make available the necessary information to guide the composition of DITSACAP-AT knowledge services.

4. DITSCAP-AT Conceptual Architecture

As we move towards more intelligent backend systems, the need to support reuse of knowledge services across multiple knowledge-intensive activities requires the corresponding processes and other cross-cutting concerns (functional and non-functional) to be modularized and easily managed as an aspectual service. Our vision for such separation and modularity leads to the conceptual architecture for DITSCAP-AT as shown in Figure 4. Through this architecture we envision a dynamic composition of *knowledge services* and *aspectual services* to satisfy the goals of the DITSCAP process components.

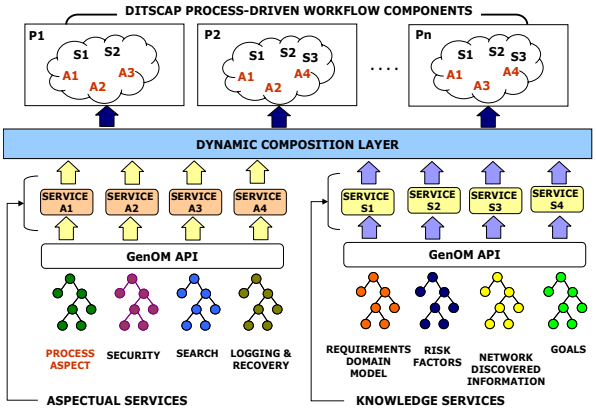

Figure 4: Conceptual DITSCAP-AT Architecture

The ideas put forth in this paper can also leverage the emerging research trends in integrating service and aspect-oriented software development paradigms [10]. Additionally, our ontology-based approach will promote uniformity, reusability, portability, and sharing of C&A artifacts which is critical for future DoD endeavors of the Global Information Grid (GIG) and net-centric dynamic C&A [11].

5. Conclusion and Future Work

In this paper we present our position on defining process activities and related artifacts as an aspectual service distinct from other generic services that assist process automation, to promote dynamic tool-support workflow composition and adaptability. Such service-oriented systems will allow the associated knowledge bases as well as the processes that utilize them, to evolve independent of each other. As part of our ongoing and future work we are exploring the possibilities of using hierarchical ontological definitions of requirements (functional and non-functional) and exposing them as aspectual services for guiding the composition of generic services based on needs of the application domain.

Acknowledgement: This work is partially supported by grant from SPAWAR Systems Center, Department of Navy, Charleston, SC, USA.

6. References

[1] Chaudhri, V. K., Farquhar, A., Fikes, R., Karp, P. D., Rice, J. P., "OKBC: a programmatic foundation for knowledge base interoperability," In Proc. of the 15th National/10th Conf. on Artificial intelligence/innovative Applications of Artificial intelligence, AAAI, CA, USA, pp: 600-607, 1998.

[2] DoD 8510.1-M, "DITSCAP Application Manual," 2000.

[3] DoD Instruction 5200.40: DITSCAP, 1997

[4] Lee, S. W., Gandhi, R. A., and Ahn, G., "Certification Process Artifacts Defined as Measurable Units for Software Assurance," To Appear in the Int'l Journal on Software Process: Improvement and Practice, Wiley, July, 2006.

[5] Lee, S. W., Muthurajan, D., Gandhi, R. A., Yavagal, D., and Ahn, G., "Building Decision Support Problem Domain Ontology from Security Requirements to Engineer Software-intensive Systems" In the International Journal on Software Engineering and Knowledge Engineering, Vol. 16(5), October, 2006

[6] Lee, S.W. Gandhi, R. A., "Ontology-based Active Requirements Engineering Framework", Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC '05), Taipei, Taiwan, Dec. 15 - 17, 2005. IEEE Computer Society Press, pp. 481 – 490.

[7] Lee, S.W. Yavagal, D., "GenOM User's Guide," Technical Report TR-SIS-NISE-04-01, Knowledge Intensive Software engineering Research Group, Dept. of Software and Information Systems, UNC Charlotte, 2004.

[8] Lee, S.W., Gandhi, R.A., Ahn, G., "Establishing Trustworthiness in Services of the Critical Infrastructure: Automating the DITSCAP. In Proc. of the 27th IEEE Int'l Conf. on Software Engineering (ICSE'05), Workshop on Software Engineering for Secure Systems (SESS), St. Louis, MO, USA, 2005, pp. 43-49 (Also appeared in ACM SIGSOFT Software Engg. Notes, 30(4), ACM Press, 2005)

[9] McGuinness, D., van Harmelen, F. (editors), "OWL Web Ontology Language Overview", W3C Recommendation, 10th February 2004

[10] Nabor C. Mendonça, Clayton F. Silva, "Aspectual Services: Unifying Service — and Aspect-Oriented Software Development" In proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05) pp. 351-356, 2005

[11] Turner, G., Holley, P., Mehan, E.J., Colon, M., "Net-Centric Assured Information Sharing – Moving Security to the Edge through dynamic certification and accreditation," IANewsletter, Vol.8(3), Winter 2005/2006

[12] van Lamsweerde, A., "Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice," In Proceedings of 12th IEEE Joint International Requirements Engineering Conference, Kyoto, 2004, pp. 4-8.