SoSML: Rumo a uma Linguagem de Modelagem Arquitetural para Sistemas-de-Sistemas Intensivos em Software

Carlos Eduardo de Barros Paes[1], **Valdemar Vicente Graciano Neto**[2]

[1]Pontifícia Universidade Católica de São Paulo (PUC-SP) – São Paulo, Brasil

[2]Universidade Federal de Goiás (UFG) – Goiânia, Brasil

carlospaes@pucsp.br, valdemarneto@ufg.br

Abstract. *Systems-of-Systems (SoS) offer innovative services that could not be offered by individual systems. Since SoS often support critical domains, it is imperative to guarantee, still at design-time, their quality. In that sense, models can contribute in such an endeavor, since they can capture the essence of the SoS architecture (structure and behavior) at design-time. However, regular languages available and frequently adopted in software engineering practice do not offer enough semantic expressiveness to precisely capture SoS idiossincrasies. In this direction, this paper discusses the main challenges of SoS modeling and introduces an excerpt of a Architecture Description Language (ADL) for specifying SoS architectures, so-named SoSML (System-of-System Modeling Language). SoSML is a semiformal ADL that is being developed as an extension of the UML, SysML and SoaML.*

Resumo. *Sistemas de Sistemas (SoS) oferecem serviços inovadores que não poderiam ser oferecidos por sistemas individuais. Como os SoS geralmente apoiam domínios críticos, é imperativo garantir, ainda em tempo de projeto, sua qualidade. Nesse sentido, os modelos podem contribuir nessa iniciativa, pois podem capturar a essência da arquitetura do SoS (estrutura e comportamento) ainda em tempo de projeto. No entanto, as linguagens disponíveis e frequentemente adotadas na prática da engenharia de software (tais como UML) não oferecem expressividade semântica suficiente para capturar com precisão as idiossincrasias das arquiteturas de SoS. Nesse sentido, este artigo discute os principais desafios da modelagem SoS e apresenta a concepção de uma Linguagem de Descrição Arquitetural (ADL) para especificar arquiteturas de SoS, denominada SoSML (System-of-System Modeling Language), uma ADL semiformal que está sendo desenvolvida pelo enriquecimento de linguagens como UML, SysML e SoaML.*

1. Introdução e Background

Sistemas-intensivos em Software (SiS) são estratégicos para vários domínios. Esses sistemas foram interconectados para trabalharem juntos e alcançarem objetivos comuns. Tais sistemas foram então denominados de Sistemas-de-Sistemas (SoS[1]). SoS oferece novas funcionalidades que não podem ser fornecidas por nenhum desses sistemas constituintes (SC) trabalhando sozinhos [Maier 1998]. Um exemplo clássico são as cidades

[1]Os acrônimos SiS e SoS serão usados para expressar suas formas plural e singular: Sistema-intensivo em software e Sistemas-intensivos em software; e Sistema-de-Sistemas e Sistemas-de-Sistemas.

inteligentes. Os SC carros autônomos e sistema de tráfego inteligente, juntos, podem oferecer, por exemplo, um comportamento emergente de **tráfego fluido** como resultado da cooperação entre eles e usando suas funcionalidades individuais (f1) regular suas próprias velocidades, (f2) se comunicar com carros próximos para regular a distância entre eles e (f3) comunicar-se com o sistema de tráfego inteligente.

Cada arranjo arquitetural que resulta de diferentes combinações de SC é denominado uma *coalizão*. Tradicionalmente, SoS são construídos usando práticas e princípios da Engenharia de Sistemas (ES). Nos últimos anos, a área de Engenharia de Sistemas vem passando por uma transformação importante que consiste basicamente na adoção de práticas consolidadas advindas da Engenharia de Software moderna. Dentre essas práticas, pode-se destacar o uso de modelos como uma forma de especificação e representação arquitetural do sistema [de Barros Paes et al. 2019]. Esse novo paradigma de ES é conhecido como Engenharia de Sistemas Baseada em Modelos (*Model Based System Engineering* - MBSE) [Friedenthal et al. 2015].

Por outro lado, a engenharia de SoS enfrenta desafios importantes devido às suas características e complexidades inerentes [Axelband et al. 2007]. O principal problema é que a Engenharia de Sistemas Baseada em Modelos utiliza como referência ADLs tradicionais (UML e SysML) concebidas para descrever a arquitetura de sistemas de software isolados (não múltiplos). Essas ADLs não dão apoio à especificação precisa para a descrição das arquiteturas de SoS de [Guessi et al. 2015].

Nesta direção, este artigo tem como objetivo apresentar os principais desafios da modelagem SoS e a proposta de ADL para a modelagem de arquiteturas SoS, denominada SoSML (*System-of-System Modeling Language*). A SoSML está sendo desenvolvida como ADL semi-formal construída a partir da extensão da linguagem UML e suas derivações SysML e SoaML, além de SoSADL[Oquendo 2016]. O restante deste artigo está organizado da seguinte forma. A seção 2 apresenta os principais desafios da modelagem SoS. A seção 3 apresenta uma visão geral da proposta SoSML. Por fim, a seção 4 apresenta as conclusões e considerações finais.

2. Desafios da Modelagem SoS

Nos últimos anos, foram propostas várias ADLs para modelagem de arquiteturas de SiS, incluindo linguagens formais e semi-formais. As ADLs semi-formais UML, SysML e SoaML foram concebidas para descrever a arquitetura de software e de sistemas embarcados. Elas **não possuem expressividade sintática suficiente para a descrição precisa das arquiteturas de SoS** [Guessi et al. 2015]. Por outro lado, surgiram ADLs formais para descrição de arquitetura de software e sistema comunicantes (comunicação entre processos) tais como Darwin (base formal FSP), Wright (base formal CSP) e pi-ADL (base formal pi-Calculus). Em particular, a combinação de ADLs formais e semi-formais foi a direção seguida pelas recentes pesquisas em ADLs para descrever SoS. Por exemplo, Bryans et al. [Bryans et al. 2014] propõem o projeto COMPASS CML, uma ADL que utiliza a SysML para especificar interfaces entre os constituintes de um SoS e a linguagem formal CML para a representação do SoS. Oquendo [Oquendo 2016] propôs uma ADL formal denominada pi-*Calculus for SoS* e sua extensão SosADL especialmente concebidas para descrever arquiteturas de SoS intensivos em software. Entretanto, o uso de linguagem formal pode representar um desafio para os engenheiros SoS, uma vez que **a**

complexidade sintática exige um conhecimento especializado com curva de aprendizado bastante elevada. Ademais, o esforço necessário para a representação de todo SoS, a **complexidade de manutenção e evolução da arquitetura** e as **dificuldades de comunicação inerentes a representação formal** tornam-se fatores importantes e impactantes na decisão do uso de ADL formal para SoS [Abrial 2006].

As características e complexidades específicas dos SoS podem ser consideradas um grande desafio para modelagem arquitetural SoS devido às suas características intrínsecas, que são [Maier 1998]: independência operacional, independência de gestão, distribuição inerente dos sistemas constituintes, desenvolvimento evolucionário e comportamentos emergentes. No caso de um ambiente de mundo aberto, pode-se adicionar o **ambiente e constituintes imprevisíveis como características e desafios no desenvolvimento deste tipo de sistema** [Lana et al. 2016]. Em essência, os desafios existentes na modelagem de SoS demandam uma mudança de paradigma de arquitetura: SoS apontam para um novo paradigma, complementando o tradicional uso de modelos em tempo de design normalmente aplicado aos sistemas do mundo fechado por novas abordagens confiáveis diminuindo assim a fronteira entre o tempo de design e o tempo de execução apoiando a correção contínua dos sistemas de mundo aberto.

Nesse sentido, modelos que representam a arquitetura e o comportamentos emergentes do SoS são necessários para melhorar o entendimento sobre eles. Por exemplo, tais ADLs são baseadas na noção de que a operação e o gerenciamento de componentes são totalmente controlados pelo sistema, o que não é o caso de constituintes em um SoS. Além disso, os componentes concretos da visão de sistema único são conhecidos em tempo de projeto, o que não é necessariamente o caso de constituintes. Enquanto as ADLs tradicionais semi-formais suportam a noção de componentes logicamente distribuídos, nenhuma delas suporta a noção de mobilidade, em particular no que se refere a interações locais inesperadas entre componentes que se movem fisicamente perto uns dos outros, como é o caso de SoS. Além disso, essas ADLs assumem que componentes concretos de uma arquitetura são conhecidos em tempo de projeto arquitetural e que podem entrar ou sair do sistema em tempo de execução sob o controle do próprio sistema. Finalmente, as ADLs tradicionais também não suportam a noção de comportamentos emergentes e a formação de coalizões em tempo de execução (desenvolvimento evolucionário do SoS), características importantes inerentes a qualquer SoS [Guessi et al. 2015].

3. Construindo a Linguagem SoSML

Com o propósito de superar os desafios da representação arquitetural de SoS, este artigo apresenta os primeiros passos rumo ao estabelecimento da SoSML (*System-of-System Modeling Language*), uma ADL para a modelagem de arquiteturas SoS. A SoSML está sendo desenvolvida como uma ADL semi-formal construída a partir da extensão da linguagem UML e suas derivações SysML e SoaML. Na definição da SoSML, serão utilizados os conceitos e blocos de construção propostos pela SosADL e os estereótipos e diagramas propostos pela SysML e SoaML. No SoSML os modelos arquiteturais serão descritos em termos de especificações abstratas de possíveis participantes SoS e seus contratos de interação e especificações concretas em termos de sistemas constituintes, mediadores e coalizões. A Figura 1 detalha os estereótipos definidos no perfil SoSML. A SoSML apoiará a representação de duas visões arquiteturais SoS. A visão arquitetural abstrata consiste na descrição em alto nível de como os participantes SoS colaboram com

o propósito de fornecer e usar interfaces expressas como contratos de interação. Cada participante do SoS consiste na representação lógica de sistema(s) constituinte(s) que participa(m) da interação (cooperação) necessária para o cumprimento das missões estabelecidas pelo SoS. Os contratos de interação representam as especificações das interações implementadas pelos participantes SoS que desempenham um papel dentro do SoS.

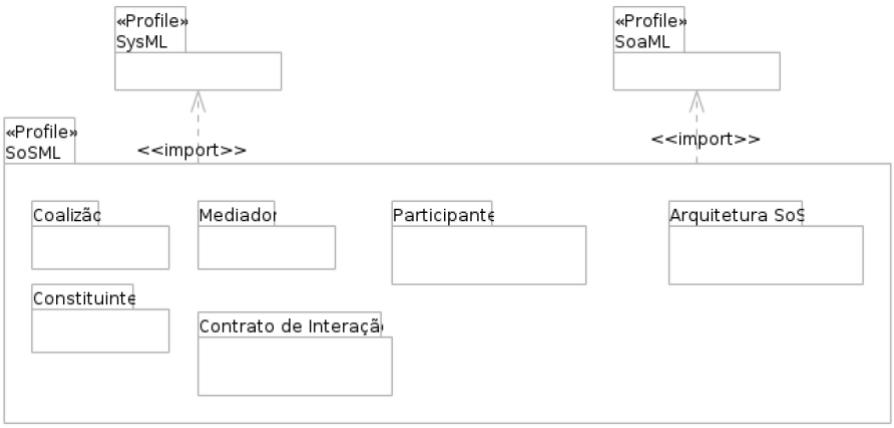

Figure 1. Estereótipos da SoSML.

Por sua vez, a arquitetura concreta representa uma coalização em tempo de execução, formada a partir da visão abstrata da arquitetura, que descreve o SoS em termos de constituintes e mediadores. Mais precisamente, para a modelagem arquitetural concreta do SoS serão utilizados os seguintes blocos de construção da SosADL estendidos a partir dos estereótipos e diagramas do SysML: (i) constituintes são elementos arquitetônicos definidos pela intenção (declarativamente em termos de sistemas) e identificados em tempo de execução (concretizados); (ii) mediadores são elementos arquitetônicos definidos pela intenção (declarativamente em termos de mediadores) e criados em tempo de execução (concretizados pelo SoS) para atingir um objetivo, parte de uma missão abrangente do SoS; (iii) coalizões são composições arquiteturais concretas de sistemas constituintes mediados, definidas por intenção (declarativamente em termos de possíveis sistemas e mediadores e políticas para suas composições dinâmicas) e criadas evolutivamente em tempo de execução (concretizadas) para alcançar uma missão SoS em um ambiente operacional.

Esse novo perfil da UML terá como objetivo aderir à expressividade necessária para a descrição precisa de arquitetura SoS. Como uma ADL semiformal, a SoSML permitirá ao arquiteto do SoS elaborar os modelos arquiteturais SoS com uma sintaxe gráfica (padrão OMG[2]), facilitando assim a comunicação entre todos os envolvidos com o projeto. Portanto, como complemento à SoSML, está sendo implementado um ambiente de desenvolvimento de arquitetura SoS para apoiar descrição e representação arquitetural de SoS usando a SoSML. O ambiente e seu conjunto de ferramentas, chamado SoSMLTools, será construído a partir do *Eclipe Modeling Framework*[3] (EMF). Além disso, será incorporado ao ambiente o recurso de transformação dos modelos SoSML

[2]OMG, Object Management Group https://www.omg.org/

[3]`https://www.eclipse.org/modeling/emf/`

para modelos descritos na linguagem formal SosADL. Com a transformação o arquiteto SoS poderá complementar os modelos SosADL para analisar o comportamento emergente e as propriedades do SoS (tempo de execução) usando o ambiente de simulação baseado em transformação de modelos, *Dynamic-SoS*, proposto por Graciano Neto [Graciano Neto et al. 2022] [Manzano et al. 2019].

4. Considerações Finais

Neste artigo foram apresentados os principais desafios da modelagem SoS e a SoSML, e as principais características de uma *Architectural Description Language* (ADL) chamada SoSML (SoS *Modeling Language*). SoSML é um perfil SysML e SoaML que incorpora conceitos consolidados da SosADL, permitindo a descrição de alto nível da arquitetura de SoS. Como SoSML é uma extensão semi-formal de um padrão existente, pode ser adotada pela comunidade de engenharia de software, com potencial para envolver tanto a academia quanto a indústria, que pode usar uma linguagem única com suporte de ferramenta para explorar o poder de duas linguagens de modelagem complementares. Trabalhos futuros incluem o desenvolvimento de uma plataforma de execução orientada a modelos arquiteturais de SoS, usando a integração com a plataforma *Dynamic-SoS* para permitir a análise, em tempo de projeto, das propriedades do SoS.

References

Abrial, J.-R. (2006). Formal methods in industry. page 761. Association for Computing Machinery (ACM).

Axelband, E., Valerdi, R., Baehren, T., Boehm, B., Dorenbos, D., Jackson, S., Madni, A., Nadler, G., Robitaille, P., and Settles, S. (2007). A research agenda for systems of systems architecting. pages 1892–1908.

Bryans, J., Fitzgerald, J., Payne, R., and Kristensen, K. (2014). Maintaining emergence in systems of systems integration: a contractual approach using sysml. *INCOSE Symposium*, 24:166–181.

de Barros Paes, C. E., Neto, V. V. G., Moreira, T., and Nakagawa, E. Y. (2019). Conceptualization of a System-of-Systems in the Defense Domain: An Experience Report in the Brazilian Scenario. *IEEE Syst. J.*, 13(3):2098–2107.

Friedenthal, S., Moore, A., and Steiner, R. (2015). *A Practical Guide to SysML*.

Graciano Neto, V. V., Manzano, W., Antonino, P. O., and Nakagawa, E. Y. (2022). Foundations and research agenda for simulation of smart ecosystems architectures. In Scandurra, P., Galster, M., Mirandola, R., and Weyns, D., editors, *Software Architecture*, pages 333–352, Cham. Springer.

Guessi, M., Cavalcante, E., and Oliveira, L. B. (2015). Characterizing Architecture Description Languages for Software-Intensive Systems-of-Systems. pages 12–18. IEEE.

Lana, C. A., Souza, N. M., Delamaro, M. E., Nakagawa, E. Y., Oquendo, F., and Maldonado, J. C. (2016). Systems-of-Systems Development: Initiatives, Trends, and Challenges. In *CLEI '16*, pages 1–10, Valparaiso, Chile. IEEE.

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, 1(4):267–284.

Manzano, W., Neto, V. V. G., and Nakagawa, E. Y. (2019). Dynamic-sos: An approach for the simulation of systems-of-systems dynamic architectures. *The Computer Journal*.

Oquendo, F. (2016). Formally Describing the Software Architecture of Systems-of-Systems with SosADL. In *11th SoSE*, pages 1–6, Kongsberg, Norway. IEEE.