

Capítulo

2

Sistemas de Sistemas de Informação e Ecossistemas de Software: Conceitos e Aplicações

Valdemar Vicente Graciano Neto, Rodrigo Santos, Renata Araujo

Abstract

Information systems (IS) have been combined to deliver new functionalities by means of interoperability among distinct IS, forming new large-scale systems called Systems of Information Systems (SoIS). SoIS are composed by pre-existing IS, with potential for innovation and business value creation. Understanding their particularities is essential to the development of the next generation of IS, which will be required to operate in distributed and open environments, to dynamically adapt to changes and to provide reliability. At the same time, each IS of a SoIS forms a software ecosystem (SECO), which comprises technical, business and social relationships that affect the way IS influences the business it supports. Understanding the intrinsic relationships between IS, SoIS and SECO becomes essential for progress in IS design and research in the forthcoming years. The aim of this work is to present the foundations on SoIS and SECO in order to enhance a common understanding of the business supported by IS, its effects on the SoIS development process, and a foreshadowing on how relations between such topics can be established.

Resumo

Softwares de sistemas de informação (SI) têm sido associados por meio da interoperabilidade para criar novas funcionalidades, formando sistemas de larga escala denominados Sistemas de Sistemas de Informação (SdSI). SdSI são compostos por SI pré-existentes, com grande potencial para inovação e criação de novos negócios. Compreender suas peculiaridades é essencial para desenvolver a próxima geração de SI, para que operem em ambientes distribuídos e abertos, adaptem-se dinamicamente a mudanças e atuem de modo confiável. Ao mesmo tempo, cada SI de um SdSI forma um ecossistema de software (ECOS), que compreende relações técnicas, de negócio e sociais que afetam a forma como o SI influencia no negócio que apoia. O entendimento das relações intrínsecas entre SI, SdSI e ECOS torna-se essencial para o progresso na concepção de SI nos próximos

anos. O objetivo deste trabalho é apresentar os fundamentos de SdSI e ECOS, de modo a potencializar a compreensão do negócio apoiado pelo SI, seus impactos no processo de desenvolvimento de SdSI, e um prenúncio sobre as relações entre ambos os tópicos.

2.1. Introdução

Nossa sociedade está experimentando uma tendência de alta conectividade, com elevada taxa de interoperabilidade, que é a capacidade de diferentes sistemas e tecnologias se comunicarem, trocarem e usarem informação. Esse fenômeno criou uma nova maneira de estabelecer relações entre pessoas, empresas e sistemas, impondo novas exigências tecnológicas.

Os Sistemas de Informação (SI) estão profundamente envolvidos com esta tendência. Eles são o elemento central de vários empreendimentos bem-sucedidos e apoiam atividades sociais e industriais essenciais. SI estão presentes na forma de Sistemas de Apoio à Decisão, Sistemas de Informação Empresariais, Redes Sociais e Sistemas de Informação Meteorológica, oferecendo serviços valiosos para beneficiar a sociedade [Graciano Neto 2016] [Graciano Neto et al. 2017a; 2017b]. À medida que as tecnologias evoluem, o SI também evolui, sendo pressionado a realizar tarefas mais complexas, estabelecendo novas relações para oferecer funcionalidades não triviais. Isso é possível, dependendo das capacidades conjuntas oferecidas por um arranjo de SI distintos. Nesta perspectiva, está surgindo uma classe distinta de sistemas conhecidos como Sistemas de Sistemas de Informação (SdSI)¹.

Um SdSI resulta de vários SI intensivos em software, operacionalmente e gerencialmente independentes que trabalham juntos para atingir objetivos de negócio comuns [Majd et al. 2015] [Saleh e Abel 2015]. Como tal, as relações entre estes SI interoperáveis são de extrema importância, uma vez que as empresas podem ser sustentadas e gerar mais valor se essas relações são melhor investigadas e compreendidas [Jansen et al. 2009]. Novas relações também podem ser estabelecidas para aumentar ganhos em produtividade, considerando dimensões técnica, de negócio e social. Por sua vez, um domínio de negócio inteiro pode ser prejudicado ou eliminado devido a relações comerciais prejudiciais entre SI. Este cenário requer novas visões sobre como compreender, descrever e analisar as relações entre esses sistemas, bem como dos novos ecossistemas que os rodeiam.

Neste contexto, o conceito de Ecossistema de Software (ECOS) tem ajudado pesquisadores e profissionais a modelar e analisar as relações existentes entre sistemas de software que compõem uma plataforma tecnológica, e entre seus atores internos e externos. Eles são importantes pois permitem prever como obter valor, retorno sobre o investimento e como as relações entre produtos distintos podem ser benéficas ou prejudiciais para o progresso do negócio no desenvolvimento de SI [Bosch 2009].

ECOS podem promover a compreensão sobre o SdSI explorando as relações existentes entre os SI constituintes de um SdSI, bem como a natureza dessas relações. Esta compreensão faz nascer o conceito de Ecossistema de Sistemas de Informação (EcoSdSI), isto é, um ECOS que envolve o desenvolvimento, integração e interoperabilidade de um

¹Por simplicidade, SdSI é usado de forma intercambiável para expressar singular e plural. A mesma regra é aplicada a SoS e SI.

conjunto de SI trabalhando em conjunto, a fim de apoiar as metas de negócios. É imperativo que este novo cenário seja investigado, uma vez que as novas relações entre SI podem beneficiar/prejudicar uma organização que oferece produtos e serviços, e cujos clientes podem combinar esses ativos para cooperar em conjunto, a fim de alcançar um maior valor de negócio.

O objetivo deste trabalho é apresentar os fundamentos e conceitos de dois tópicos emergentes e estratégicos para a área de SI: os Sistemas de Sistemas de Informação (SdSI) e os Ecossistemas de Software (ECOSs). Dessa forma, visa esclarecer os elementos que envolvem SdSI (múltiplos SI e múltiplas organizações colaborando para explorar novas oportunidades de negócio) e ECOS, bem como o potencial de associação entre os tópicos, sem perder de vista a influência destas novas tendências na Engenharia de Software (ES) para a área de SI.

O trabalho está estruturado da seguinte forma: a Seção 2.2 apresenta as bases de SI; a Seção 2.3 apresenta as bases de ecossistemas digitais; a Seção 2.4 apresenta as bases de ECOS; a Seção 2.5 apresenta as bases de SoS; a Seção 2.6 discute as relações de ECOS e SoS; a Seção 2.7 discute o prenúncio dos SdSI; a Seção 2.8 discute o advento das relações entre ECOS e SdSI (EcoSdSI); por fim, a Seção 2.9 conclui o trabalho.

2.2. Sistemas de Informação

De acordo com a Teoria Geral de Sistemas (TGS) [Bertalanffy 2015], sistemas são um conjunto de elementos dinamicamente inter-relacionados para realizar atividades visando atingir uma meta específica, consumindo energia, materiais ou dados (entrada) e produzindo novas formas de energia, materiais ou dados (saída). O conceito de sistema tem sido útil para descrever e compreender o comportamento de estruturas complexas em muitos domínios de conhecimento diferentes – da Biologia às Ciências Sociais, e particularmente na área de SI.

Com base nesse conceito, uma definição técnica muito simples de SI é descrita por [Laudon e Laudon 2016] como um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informações. O uso de um mural em um corredor de uma organização onde diversas pessoas divulgam informação pode ser compreendido como um SI composto por humanos e objetos (mural, papéis etc.) onde informação pode ser publicada, recuperada, processada e eliminada, de forma a apoiar a organização no alcance de seus objetivos e na comunicação de informação relevante.

Conceitualmente, SI também pode incluir software ou outra tecnologia como um dos seus elementos (SI intensivos em software). Em um contexto de uso de um portal de informação organizacional, por exemplo, onde profissionais relatam e compartilham notícias e informação usando seus celulares também pode ser considerado um SI. Muito frequentemente, o termo *sistemas de informação* é utilizado como referência ao software que processa informação para um conjunto de usuários. Isto é perfeitamente correto se compreendemos que um software, por sua vez, pode ser composto de partes inter-relacionadas (e.g., módulos, funções) com o objetivo de processar informação. Doravante neste texto, consideramos SI intensivo em software e SI como termos intercambiáveis.

No entanto, não podemos restringir o entendimento de SI ao elemento de software. De volta à TGS [Bertalanffy 2015], sistemas (i) podem existir dentro de outros sistemas (que variam em hierarquia e complexidade), (ii) são usualmente abertos (interagem com o ambiente em que estão inseridos e aprendem com esta interação), (iii) seu funcionamento depende de sua estrutura interna (as relações entre seus elementos), e (iv) possuem regras que ajudam a manter seu equilíbrio e regulam sua operação (tentam evitar variações que podem danificar sua operação, ou porque precisam ser capazes de se adaptar a novas situações, reequilibrando-se). O comportamento de sistemas pode ser previsível e descritivo (e.g., computadores) ou imprevisíveis, complexos e difíceis de serem descritos (e.g., sistemas sociais ou econômicos).

Uma visão abrangente de SI voltada ao entendimento da complexidade das estruturas sistêmicas do mundo moderno é o conceito de ecossistemas, usualmente utilizado em áreas como Biologia [Dhungana et al. 2010], adaptado para comportar a presença do componente tecnológico, denominados Ecossistemas Digitais (ECODigs).

2.3. Ecossistemas Digitais

Um ecossistema, na Biologia, é uma comunidade de organismos vivos (i.e., plantas, animais e microrganismos) em conjunto com componentes não vivos (e.g., ar, água e solo), bem como suas relações entre si e com o ambiente, interagindo como um sistema [Smith e Smith 2012]. Muitas das atuais características atribuídas aos ECODigs foram inspiradas na observação dos ecossistemas da natureza [Boley e Chang 2007].

Ecossistema digital (do inglês, *digital ecosystem*) é um paradigma emergente para inovação tecnológica. Consiste de uma infraestrutura digital auto organizável com o intuito de criar um ambiente digital para as organizações (ou agentes) conectadas em rede, provendo apoio à cooperação, compartilhamento de conhecimento e desenvolvimento de tecnologias adaptativas e abertas [IEEE 2007].

Um ECODig consiste em uma comunidade aberta, onde não há nenhuma necessidade permanente de controle centralizado ou distribuído. Uma estrutura de liderança pode ser formada (e dissolvida), em resposta às necessidades dinâmicas do ambiente [Boley e Chang 2007]. Assim, os ECODigs promovem mudanças também nas tradicionais formas de comunicação, pois as organizações (públicas ou privadas) deixam de atuar como ilhas isoladas para compor um ecossistema interconectado por meio de técnicas de engajamento alavancadas pelas TICs [Araújo e Magdaleno 2015].

Boley e Chang (2007) resumem as características essenciais dos ECODigs baseadas nos conceitos de agentes (entidades que se integram a um ambiente ou comunidade com base em seus próprios interesses) e espécies (tipos de agentes) e inspiradas na analogia com a biologia:

Abertura, Interação e Engajamento: A abertura se refere a um ambiente virtual transparente, onde existe interação entre os agentes do ECODig visando o bem-estar social e o engajamento com outros para obter oportunidades e compartilhar recursos. Às vezes, a comunidade precisa se unir para se defender contra ameaças externas. Os agentes não serão capazes de sobreviver a menos que reconheçam que são inter-

dependentes em relação a outras espécies deste ecossistema e que estejam dispostos a cooperar com elas;

Balanceamento: O balanceamento significa a harmonia, estabilidade e sustentabilidade dentro de um ECODig. Se alguma espécie está ficando desproporcionalmente tensionada ou dividida, todo o ecossistema pode entrar em colapso. No entanto, um único ponto de falha não precisa levar a um desastre, mas pode dar origem a um novo equilíbrio do ecossistema como um todo;

Agrupamento por domínio e acoplamento fraco: As espécies compõem um ecossistema por escolha própria. Seus membros compartilham uma cultura, hábitos sociais, interesses e objetivos semelhantes. Cada espécie preserva o meio ambiente comum e é proativo(a) e receptivo(a) para o seu próprio benefício. Ao mesmo tempo, os agentes percebem as vantagens da colaboração e existe um interesse mútuo comum entre as partes. Assim, possuem entusiasmo em participar do trabalho da comunidade. Eles são, portanto, capazes de viver juntos em comunidade e apoiar uns aos outros para a sustentabilidade do ECODig como um todo;

Auto-organizável: Cada espécie é independente, auto-habilitada, auto-preparada, capaz de se defender e sobreviver através da auto-coordenação. Os agentes de um ecossistema digital são capazes de agir de forma autônoma, tomar decisões e cumprir responsabilidades.

2.4. Ecossistemas de Software (ECOS)

ECOS podem ser vistos como uma classe particular de ECODig centrados intensamente em software. O ECOS é uma forma eficaz de construir software em cima de uma plataforma tecnológica comum (por exemplo, sistema operacional, base de ativos de software etc.), compondo aplicações e tecnologias desenvolvidas por múltiplos atores (desenvolvedores de terceiros, comunidades e organizações) [Santos et al. 2012]. Além disso, um ECOS compreende uma tecnologia de fundação ou um conjunto de componentes utilizados para além de uma única empresa que reúne várias partes para um objetivo comum de negócio ou de desenvolvimento, ou para resolver um problema comum.

Esta plataforma tecnológica comum pode originar produtos de software que podem cooperar e/ou competir no mercado, ou mesmo outras relações podem ser traçadas entre eles. ECOS também é caracterizado tanto pela produção de software quanto pelas relações de consumo. Essas relações podem ser estabelecidas com desenvolvedores de terceiros, comunidades e/ou outras organizações para promover o desenvolvimento, fornecimento e evolução de componentes em um grande ecossistema criado sobre a plataforma tecnológica comum. Exemplos de ECOS incluem o ECOS Microsoft, o ECOS iPhone e o ECOS Drupal [Santos et al. 2012]. Adicionalmente, um ECOS pode fazer parte de outro ECOS, por exemplo, o ECOS Microsoft CRM faz parte do ECOS Microsoft.

Os fornecedores de software co-desenvolvem capacidades em torno da inovação: trabalham de forma cooperativa e competitiva para apoiar e desenvolver novos produtos, satisfazer as necessidades dos clientes e inovar continuamente [Moore 1996]. Essas redes

apertadas de fornecedores, distribuidores, empresas de terceirização, desenvolvedores de produtos ou serviços relacionados, provedores de tecnologia e uma série de outras organizações afetam e são afetadas pela criação e entrega de produtos e serviços de um fornecedor de software. Alinhados a este ponto de vista, os investigadores propuseram a perspectiva de ECOS para analisar a indústria de software, [Jansen et al. 2009] [Manikas 2016]. Para isso, as dimensões técnica, de negócio e social precisam ser levadas em consideração para entender como modelar e analisar um ECOS.

Nas perspectivas social e de negócio, um ECOS fornece visão organizacional complementar ao desenvolvimento de SoS, que define papéis, regras de interação, colaboração e capacidades sinérgicas para seus sistemas constituintes. Existem muitas semelhanças entre as características de SoS [Santos et al. 2014] e os desafios técnicos ECOS [Bosch 2009], por exemplo, como garantir a estabilidade da plataforma, simplicidade, segurança, confiabilidade e evolução. Nesse sentido, podemos conjecturar que a ECOS e SoS (incluindo SdSI) também podem manter relações intrínsecas e sinérgicas que podem ser exploradas. A próxima seção discute essa perspectiva.

2.5. Sistemas de Sistemas (SoS)

SoS são um arranjo de sistemas interoperáveis, chamados *constituintes*, concebidos para realizar um conjunto de missões. SoS surgiram como resultado da necessidade de interoperabilidade entre sistemas de software individuais para criar aplicações mais robustas, oferecendo funcionalidades que não poderiam ser entregues por nenhum destes sistemas de forma isolada [Maier 1998]. Primeiramente presentes no domínio militar, estes sistemas têm migrado para o domínio civil, com advento das cidades inteligentes e dos sistemas de monitoramento urbano de enchentes e de gerenciamento de crises e emergências [Santos et al. 2014] [Paes et al. 2016].

SoS compartilham cinco características postuladas por Maier [Maier 1998]: (i) independência operacional dos constituintes, uma vez que os constituintes operam de forma autônoma, contribuindo ocasional e oportunamente para cumprir missões no contexto SoS; (ii) independência gerencial dos constituintes, isto é, uma diversidade de stakeholders e organizações podem possuir e/ou gerenciar um ou mais constituintes que fazem parte de um SoS; (iii) comportamento emergente, uma funcionalidade mais complexa decorrente da interoperabilidade entre constituintes, deliberadamente planejada para cumprir uma ou mais missões ou surgindo de modo não previsto devido à dinâmica do SoS; (iv) desenvolvimento evolucionário, uma vez que o SoS como um todo evolui como resultado da evolução de seus constituintes; e (v) distribuição, uma vez que sua comunicação se baseia na tecnologia de rede. Além disso, SoS apresentam frequentemente uma natureza oportunista, isto é, um sistema é capaz de unir outros sistemas para formar um SoS para realizar uma missão e deixar o SoS quando a missão termina [Graciano Neto et al. 2014]. A arquitetura dinâmica também foi considerada uma característica notável de SoS e que surge como consequência das outras características que definem um SoS [Oquendo 2016]. Por fim, outras características como autonomia, heterogeneidade, e autoadaptação têm sido também atribuídas a SoS ao longo dos anos [Boardman e Sauser 2006].

SoS são desenvolvidos orientados a missões. Tais missões constituem objetivos de alto nível que, para serem atingidos, valem-se da interoperabilidade entre os constituintes

e das suas funcionalidades individuais. Missões são especificadas em nível de requisitos, enquanto comportamentos emergentes correspondentes a estas missões são planejados em tempo de projeto do SoS [Oquendo 2016]. Missões estão relacionadas tanto com as capacidades dos sistemas constituintes como com as interações entre esses sistemas que contribuem para a realização dos objetivos globais dos SoS. A atividade de concretização de um modelo de missão consiste no seu refinamento para um modelo de arquitetura que possua constituintes para os quais seja possível atribuir partes das missões estabelecidas de modo a formar comportamentos emergentes em conformidade com estas missões e para atingir os objetivos estabelecidos por elas [Silva et al. 2016].

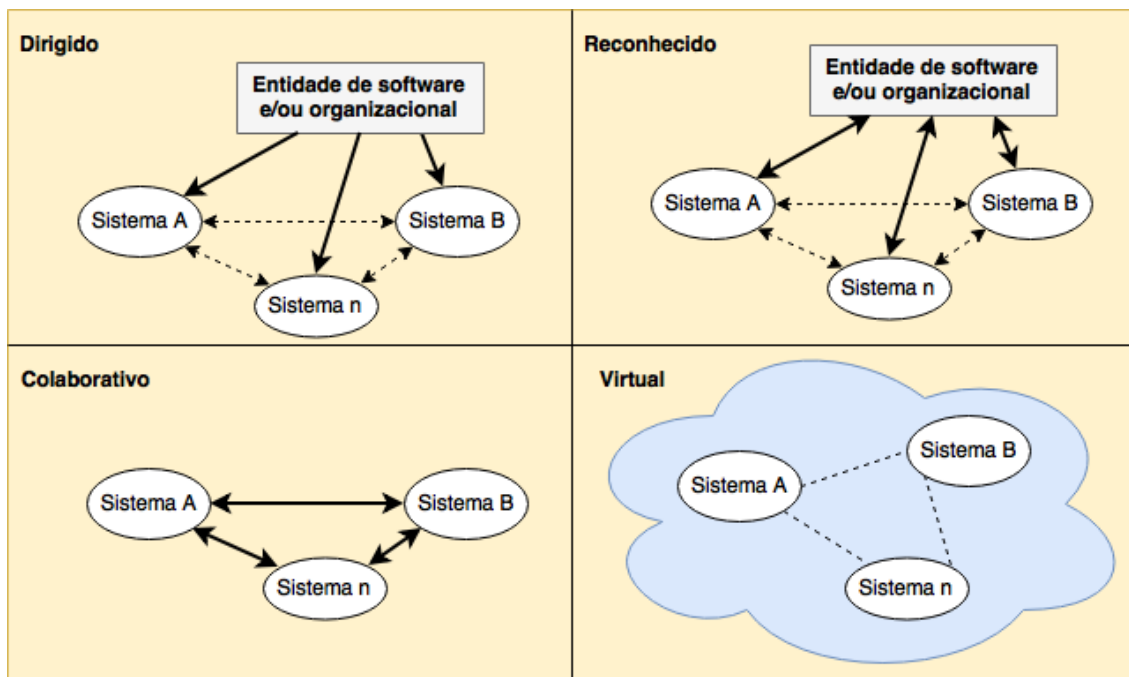


Figura 2.1. Concepção visual da taxonomia de SoS vigente na literatura especializada.
Fonte: [Lane 2013]

SoS têm sido amplamente distinguidos pelo nível de controle gerencial [Dahmann et al. 2008], por sua autoridade central e nível de colaboração [Bryans et al. 2013] [Pérez et al. 2013]. Esta classificação estabelece quatro tipos de SoS [Maier, 1998] [Dahmann et al. 2009] [Pérez et al. 2013]: direcionado, colaborativo, reconhecido e virtual. De acordo com a taxonomia de Maier, abaixo encontram-se os tipos essenciais de SoS e suas respectivas características [Maier 1998] [Dahmann et al. 2009] [Lane 2013]. Em paralelo, a Figura 2.1 ilustra uma concepção de como os sistemas constituintes interagem entre eles de acordo com os tipos de SoS discutidos [Lane 2013]. Sob essa perspectiva, um SoS pode ser:

Dirigido: este tipo de SoS possui uma autoridade central (integrador², *broker*, ou *glue code*), que pode ser uma entidade de software ou um grupo de gestores. Ele es-

²Integração não têm sido vista como uma palavra adequada para designar o modo como constituintes são associados para formar um SoS uma vez que integrar passa a ideia de um todo indivisível e permanente, o que não é o caso do SoS, que pode ser um fenômeno breve, transiente, que dure de horas a anos, mas não permanente.

tabelece uma comunicação unilateral com os constituintes, orquestrando a forma (sequência de tarefas e atribuições) como eles vão trabalhar para cumprir missões. Nesta modalidade, os constituintes podem não estar cientes de que fazem parte de um SoS, e suas ações são dirigidas pela entidade centralizadora. Seus sistemas operam subordinados à finalidade gerida centralmente [Pérez et al. 2013];

Reconhecido: neste tipo, uma entidade de software estabelece uma via bidirecional de comunicação, onde esses sistemas consultam essa entidade de software para descobrir outros constituintes disponíveis e serviços prestados. A autoridade central oferece uma resposta, permitindo que os constituintes se comuniquem, organizando-se para cumprir uma ou mais missões como resultado;

Colaborativo: SoS colaborativos são distintos dos sistemas direcionados no sentido de que a entidade central gestora ou de software não possui poder coercitivo para executar o sistema. Os sistemas constituintes devem colaborar voluntariamente para cumprir os objetivos centrais estabelecidos. A Internet é um sistema colaborativo. Embora os padrões sejam definidos para ela, não há um poder absoluto para impô-los. Os acordos entre os atores centrais sobre a prestação de serviços e a rejeição fornecem o mecanismo de execução que existe para manter as normas. A cooperação entre os constituintes surge espontaneamente, mas com uma intenção prevista de interação dentro de um propósito pré-definido e embutido. A Internet começou como um sistema direcionado, controlado pela Agência de Projetos de Pesquisa Avançada dos EUA, para compartilhar recursos de computadores. Ao longo do tempo, evoluiu através de mecanismos colaborativos não planejados [Maier 1998];

Virtual: Nesta categoria, não há autoridade de gestão centralizada. O comportamento em grande escala surge e pode ser desejável, mas todo o sistema deve depender de mecanismos relativamente invisíveis para mantê-lo. Os constituintes não estão sujeitos a uma autoridade de gestão central e não existe um objetivo claro para o SoS. Os sistemas constituintes não necessariamente conhecem uns aos outros. Eles executam suas funcionalidades individuais e o resultado final acaba por ser a soma dos resultados parciais, sem uma intenção clara ou colaboração explícita entre eles. Exemplos notáveis citados na literatura incluem a forma atual da World Wide Web e economias nacionais. Ambos os sistemas são distribuídos fisicamente e gerencialmente. A World Wide Web é ainda mais distribuída do que a Internet, já que ninguém desempenha um controle central, exceto nas primeiras etapas [Maier 1998] [Lane 2013].

É importante destacar que pesquisadores especialistas em SoS têm questionado a taxonomia vigente proposta por Maier em seu artigo seminal de 1998. A existência do SoS *virtual* tem sido contestada, e a taxonomia estaria reduzida apenas aos três primeiros tipos. O tipo virtual seria apenas uma variação um pouco mais desacoplada do tipo *colaborativo*. Ainda não há consenso ou proposta de nova taxonomia. Não obstante, é possível inferir que à medida que o nível de controle gerencial vai diminuindo entre os tipos de SoS, a autonomia dos constituintes para interoperar sem intermédio de uma entidade reguladora aumenta.

Durante muito tempo, a modelagem de SoS foi realizada sob uma perspectiva tradicional centrada em documentos, sofrendo com (i) replicação de informações, (ii) falta de rastreabilidade entre documentos, (iii) inconsistências de informações e regras de negócios e (iv) dificuldades para manusear e pesquisar informações em tais documentos [Paes et al. 2016]. No entanto, ao longo da última década, os engenheiros da SoS aumentaram significativamente a adoção da Engenharia de Sistemas Baseados em Modelos (MBSE), uma abordagem de engenharia de SoS que muda de uma perspectiva documental para uma realidade baseada em modelos, enfatizando o desenvolvimento e adoção de modelos para engenharia de SoS [Ramos et al. 2012]. O MBSE promete (i) uma gestão mais eficaz do conhecimento que pode aumentar a capacidade das partes interessadas para compreender o sistema, bem como seu comportamento e desempenho, (ii) melhor comunicação da equipe, (iii) processos explícitos de decisão sobre questões do sistema, (iv) detecção precoce de erros e omissões, (v) arquitetura de sistemas melhorada, (vi) integridade detalhada do projeto e (vii) rastreabilidade efetiva do projeto. SysML (*System Engineering Modeling Language*), uma extensão da UML (*Unified Modeling Language*), é considerada uma notação padrão central no MBSE [Graciano Neto et al. 2014] [Paes et al. 2016].

SoS frequentemente dão suporte a domínios críticos, ou seja, aplicações em que erros podem causar ameaças à integridade humana, perdas de vidas, ambientais, e financeiras [Graciano Neto 2017]. Neste sentido, é essencial garantir que SoS sejam aplicações baseadas em software com operação confiável e robusta. Em virtude disso, os modelos de desenvolvimento de SoS têm sido intensamente voltados às atividades de verificação e validação (V&V) (formal inclusive, se possível), uma vez que V&V pode reduzir o número de falhas esperadas, proporcionando um nível de confiança suficientemente elevado para ser considerado aceitável para estar disponível no mercado [Graciano Neto et al. 2017b]. Sabe-se que toda atividade de verificação e validação exige algum nível de especificação que possibilite a execução destas atividades. Os modelos de especificação de candidatos para suportar V&V incluem modelos de exigência (tais como modelos de sequência UML, requisitos textuais, diagramas de casos de uso ou diagramas de blocos SysML) ou modelos de arquitetura de software. Qualquer um destes modelos deve ser desenvolvido inicialmente no ciclo de vida SoS para evitar custos elevados devido a retrabalho ou especificação ou implementação errada. Para esse contexto, os modelos arquiteturais parecem ser uma opção melhor para apoiar as atividades V&V, uma vez que inerentemente mantêm a estrutura do SoS, incluindo elementos constituintes, suas relações e dinâmica. Além disso, os modelos arquiteturais são mais ricos em detalhes do que os modelos de requisitos, que podem ser incompletos, imprecisos e rudimentares para suportar um processo V&V confiável.

No que se refere a SoS, uma arquitetura de software engloba a estrutura de software fundamental do SoS, incluindo seus constituintes e conexões entre eles, bem como propriedades sobre seus constituintes e do ambiente [Nielsen, 2015]. As descrições de arquitetura são artefatos tangíveis que expressam essa arquitetura de software [ISO 42010]. Assim, é imprescindível adotar modelos que capturam precisamente tais arquiteturas [Nielsen 2015]. Arquiteturas de software são frequentemente descritas por meio de linguagens de descrição arquitetural (ADL), ou seja, linguagens específicas de domínio especialmente concebidas para a representação de arquiteturas de software [ISO 42010]. A ADL

oferece construções canônicas para especificar adequadamente arquiteturas de software, e muitas vezes são categorizadas em diferentes níveis de formalismo, ou seja, informal (linguagens baseadas em representações de caixa e linhas), semiformal (linguagens com sintaxe bem definida e uma semântica completa) e formal (linguagens com sintaxe e semântica formalmente definidas) [Oquendo 2016].

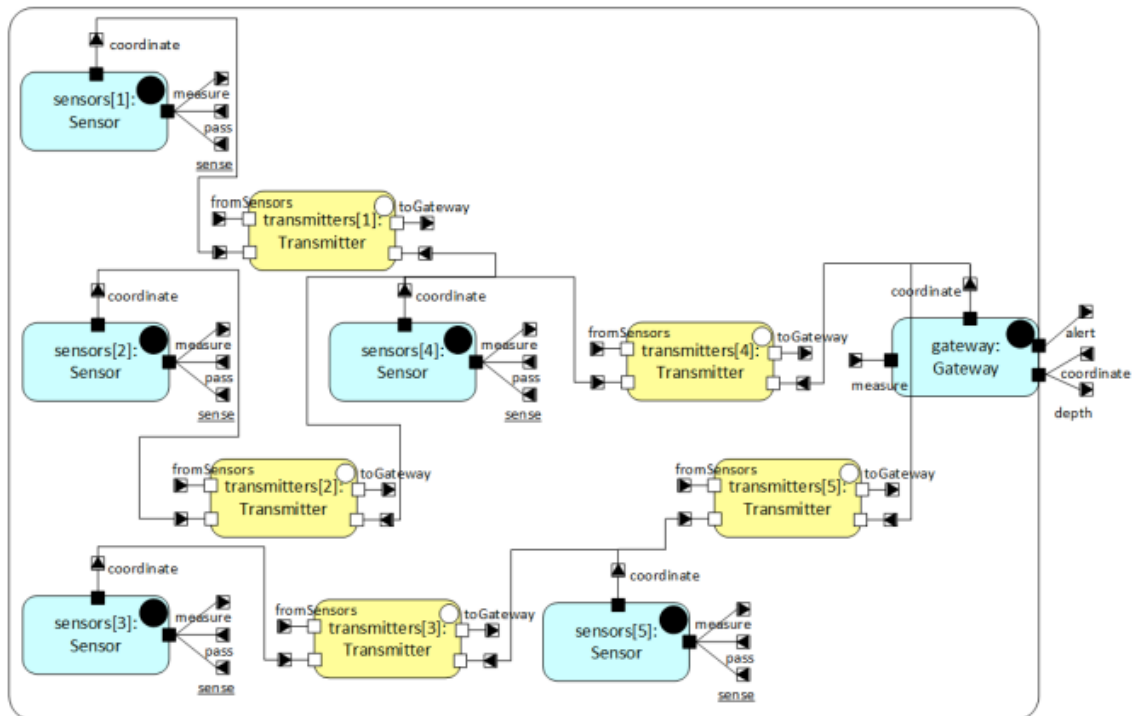


Figura 2.2. Arquitetura concreta de um SoS de monitoramento de enchentes: um exemplo de uma coalizão que satisfaz restrições de uma arquitetura abstrata.
 Fonte: [Oquendo 2016]

Uma nova ADL, chamada SosADL, foi introduzida para descrever arquiteturas de SoS. Ela fornece sintaxe e semântica formalmente definidas para elementos de arquitetura de SoS, incluindo comportamentos emergentes [Oquendo 2016]. SosADL descreve SoS como uma combinação de arquitetura, sistemas e mediadores. Mediadores são elementos da arquitetura que estabelecem comunicação entre dois ou mais constituintes [Wiederhold 1992]. Cada declaração de arquitetura é expressa em termos de seu comportamento intrínseco, tipos de dados e portas, ou seja, abstrações que permitem o estabelecimento de conexões. Uma conexão é estabelecida para receber estímulos ou agir sobre o ambiente, ou simplesmente comunicar com outros constituintes. Mediadores e sistemas, bem como a arquitetura SoS em si também têm portas, tipos de dados e comportamentos. Os sistemas desempenham o papel de constituintes, e são mediados por mediadores. SosADL dá suporte à representação de comportamento emergente por meio de uma coalizão, isto é, uma aliança temporária que permite que os constituintes executem uma ação combinada. Estes comportamentos emergentes são especificados como parte do comportamento da

coalizão, documentando como os constituintes devem interagir para realizar um determinado conjunto de missões³ [Oquendo 2016].

Em SosADL, a descrição da arquitetura de um SoS é especificada em dois níveis: a abstrato (define os possíveis tipos de sistemas constituintes que podem participar do SoS) e concreto (instancia a arquitetura abstrata de acordo com os sistemas constituintes selecionados). A Figura 2.2 traz uma concepção de uma arquitetura de SoS concreta para um SoS de monitoramento de enchentes via sensores, instalado no Rio Monjolinho, em São Carlos, Brasil [Oquendo 2016].

```

architecture WnsMonitoringSosArchitecture() is {...
  behavior coalition is compose {
    sensors is sequence{Sensor}
    gateway is Gateway
    transmitters is sequence{Transmitter}
  } binding {...
    forall{isensor1 in sensors, isensor2 in sensors
      suchthat
        exists{itransmitter in transmitters
          suchthat
            (isensor1 <> isensor2) implies
              unify one{itransmitter::fromSensors}
                to one{isensor1::measurement::measure}
              and unify one{itransmitter::towardsGateway}
                to (one{isensor2::measurement::pass}
                  xor unify one{itransmitter::towardsGateway}
                    to one{gateway::notification::measure}
                )
            }
          }
        // multiplicities are 'one', 'none',
        // 'lone' (none or one),
        // 'any' (none or more),
        // 'some' (one or more), 'all'
      }
    }
  } guarantee {...}
}

```

Figura 2.3. Arquitetura de SoS abstrata descrita em SosADL: descrição da coalizão em termos de ligações intencionais.

Fonte: [Oquendo 2016]

Por sua vez, a Figura 2.3 mostra como uma arquitetura abstrata do mesmo SoS pode ser especificada em SoSADL. Conforme ilustrado na Figura 2.3, uma coalizão pode envolver possivelmente muitos constituintes, exatamente um constituinte do tipo gateway e possivelmente muitos mediadores. Esta coalizão não especifica quais sistemas constituintes existirão em tempo de execução, mas sim quais são os possíveis sistemas que podem existir e quais são as condições necessárias para formar uma coalizão entre os sistemas identificados em tempo de execução para participar do SoS. Da arquitetura abstrata com ligações intencionais descritas acima, podem ser criadas diferentes arquiteturas SoS concretas. Estas arquiteturas SoS concretas resultam da seleção em tempo de execução de possíveis sistemas que podem participar do SoS. No caso do rio Monjolinho, mostrado na Figura 2.2, existem cinco sensores instalados e um *gateway*. Alguns sensores estão muito

³Detalhes adicionais sobre a sintaxe de descrições de arquitetura em SosADL podem ser encontrados em [Oquendo 2016].

longe do *gateway* para transmitir medidas diretamente, precisando executar um *multihop* usando sensores intermediários [Oquendo 2016].

SoS têm sido implementados utilizando uma grande diversidade de plataformas subjacentes. Algumas incluem Sistemas Distribuídos (utilizando plataformas de *middleware* para interoperar seus constituintes), Sistemas de Barramento Empresarial (*Enterprise Bus Systems – EBS*), e Arquitetura Orientada a Serviços (*Service Oriented Architecture – SOA*). Outras plataformas potenciais são as plataformas de *Cloud* e Internet das Coisas (*Internet of Things – IoT*), de modo que a última oferece abstrações e serviços alinhados ao que se imagina para SoS. Independentemente da plataforma subjacente escolhida para implementação do SoS, é importante destacar que a simulação é uma etapa essencial na maioria dos processos de desenvolvimento de SoS. As abordagens baseadas em simulação tem auxiliado principalmente no tratamento das propriedades dinâmicas de SoS, tais como arquitetura dinâmica e comportamento emergente [Mittal e Rainey 2015] [Nielsen, 2015], uma vez que simulações (i) apoiam a validação de comportamentos emergentes previstos, (ii) possibilitam a observação de comportamentos emergentes inesperados; (iii) possibilitam a previsão de erros, seu diagnóstico e correção; e (iv) fornecem uma abordagem visual e dinâmica, reproduzindo estímulos que o SoS pode receber do ambiente sem necessariamente estar em operação, poupando custos e reduzindo riscos [Graciano Neto 2016].

Vários projetos têm sido concebidos para a concepção de SoS. Na Europa, os projetos DANSE e COMPASS são expressivos no projeto e desenvolvimento de SoS. DANSE adota SysML para semi-formalmente descrever arquiteturas executáveis e testáveis. COMPASS desenvolveu uma abordagem chamada *Compass Modeling Language (CML)* para modelagem e análise de SoS [Oquendo 2016]. Os grupos de pesquisa START (*Software Architecture Team, ICMC/USP*) e ArchWare (*IRISA/UBSA/França*) são exemplos de grupos de pesquisa expressivos em desenvolvimento e produção de linguagens, ferramentas, e processos para a concepção de SoS na Europa e no Brasil, respectivamente. Investimentos substanciais foram feitos para apoiar a engenharia SoS na Arábia Saudita, que investiu 70 bilhões de dólares em cidades inteligentes, e na África do Sul, que iniciou um projeto de 7,4 bilhões de dólares em cidades inteligentes⁴. No Brasil, o projeto *Rede de Cidades Inteligentes e Humanizadas*⁵ ganha destaque, com objetivo de apresentar um conceito brasileiro do que vem a ser uma cidade inteligente e humana, e também uma proposta de metodologia de trabalho para que, em conjunto com outras instituições, possa organizar a captação de recursos e iniciar um trabalho para efetivar a existência de cidades inteligentes até o ano de 2030.

Além das cidades inteligentes, outros exemplos de SoS incluem o SoS de Observação Global (GEOSS⁶ - *Global Earth Observation System of Systems*), O GEOSS é um conjunto de sistemas coordenados e independentes de observação, informação e processamento que interagem e fornecem acesso a diversas informações para um amplo leque

⁴Keeping Smart Cities Smart: Preempting Emerging Cyber Attacks in U.S. Cities: <http://icitech.org/wp-content/uploads/2015/06/ICIT-Smart-Cities-Brief1.pdf> (2015). Último acesso: Maio de 2017.

⁵Rede Brasileira de Cidades Inteligentes: <http://redebrasileira.org/brasil-2030> (2016). Último acesso: Maio de 2017.

⁶<http://www.earthobservations.org/geoss.php>.

de usuários nos setores público e privado. GEOSS liga esses sistemas para fortalecer o monitoramento do estado da Terra, facilita a partilha de dados e informações ambientais recolhidos a partir da grande variedade de sistemas de observação contribuídos por países e organizações. Esta infraestrutura compreende sistemas de observação, que incluem sensores terrestres, aéreos, aquáticos e espaciais, pesquisas de campo e observatórios de cidadãos; e SI e processamento, que incluem ferramentas de hardware e software necessárias para o manuseio, processamento e fornecimento de dados dos sistemas de observação para fornecer informação, conhecimento, serviços e produtos.

Outro projeto notável é o projeto Rescuer, um SoS para Gerenciamento de Crises e Emergências⁷. O Rescuer desenvolveu uma solução de *crowdsourcing* para apoiar centros emergências e gerenciamento de crises. O foco especial é sobre os incidentes em áreas industriais e em eventos de grande porte. Eventos de grande escala – como a Copa do Mundo de Futebol e os Jogos Olímpicos – exigem preocupação em relação à segurança. Além da enorme infraestrutura de organização de tais eventos, a segurança dos visitantes, em particular, é uma questão central. Um protótipo experimental do Rescuer foi testado na Copa do Mundo de 2014 e usado para os Jogos Olímpicos em 2016. Esta parceria euro-brasileira inclui nove instituições de quatro países (Brasil, Alemanha, Áustria e Espanha), com a coordenação brasileira da Universidade Federal da Bahia (UFBA) e europeia do Instituto Fraunhofer para Engenharia de Software Experimental IESE. O Rescuer é financiado pela União Europeia e pelo Ministério da Ciência, Tecnologia e Inovação. Os parceiros do consórcio Rescuer do lado europeu são: (1) Instituto Fraunhofer para Engenharia de Software Experimental IESE, (2) Centro de Pesquisa Alemão para a Inteligência Artificial (DFKI), (3) Universidad Politécnica de Madrid, (4) VOMATEC International GMBH e (5) FireServ EU. Do lado brasileiro, os parceiros são: (6) Universidade Federal de Bahia (UFBA), (7) Universidade de São Paulo (USP), (8) MTM Informática Serviços Ltda e (9) Comitê de Fomento Industrial de Camaçari (COFIC).

2.6. Relações entre SoS e ECOS

SoS e ECOS são tópicos emergentes de pesquisa com interseção em vários pontos. O primeiro é a preocupação com arquitetura. Outro elemento é a utilização de SoS como a *plataforma* sobre a qual edifica-se um ECOS. Este tipo de plataforma fornece serviços específicos de domínio e gerais a um conjunto de sistemas que precisam interagir para formar um SoS [Maier 1998]. Do ponto de vista do ECOS, uma plataforma SoS pode existir em um ambiente de diferentes níveis de atores, artefatos e relacionamentos para o desenvolvimento de produtos e serviços globalizados, de larga escala e de longo prazo [Santos et al. 2012]. Esses produtos e serviços podem ser desenvolvidos com tecnologias diferentes, onde a integração e a comunicação são cruciais, dado que são sistemas intensivos em software. Além disso, SoS é uma estratégia para alcançar objetivos, oferecendo funcionalidades exclusivas que são resultado de um trabalho colaborativo de um conjunto dinâmico de sistemas existentes [Santos et al. 2014].

Existem algumas semelhanças entre as características de SoS [Maier 1998] e os desafios técnicos do ECOS [Bosch 2009]. Por exemplo, a estabilidade arquitetural necessária para as plataformas ECOS em relação aos seus componentes, serviços e aplicações

⁷<http://www.rescuer-project.org/?lang=pt>.

pode ser comparada com a independência operacional de sistemas constituintes de um SoS. Nesse caso, a integração de sistemas de software e o desenvolvimento baseado em componentes podem ser combinados para oferecer suporte a estratégias para lidar com problemas de interface de programação de aplicativos. Por outro lado, a evolução da plataforma depende diretamente das necessidades e contribuições emergentes da comunidade ECOS, bem como dos ajustes dos modelos de negócios híbridos subjacentes. Como tal, o desenvolvimento evolutivo de SoS deve ter em conta não só as questões técnicas do ambiente, mas também as questões empresariais e sociais. Nesse caso, os modelos de arquitetura SoS devem ser estendidos para lidar com variáveis de contexto baseadas em cadeias de valor e redes sociais. Por fim, o comportamento emergente produzido por sistemas constituintes de um SoS trabalhando em conjunto pode estar relacionado à segurança e confiabilidade do ECOS. A teoria da dinâmica de sistemas pode ser um instrumento útil para simular componentes para melhorar o design da arquitetura do SoS [Santos et al. 2014].

As relações entre ECOS e SoS estão associadas a questões específicas das três dimensões de um ECOS. Um mapeamento realizado sobre o assunto [Jeronimo Junior e Werner 2015] identificou as principais áreas estudadas neste contexto, os possíveis benefícios que um SoS pode obter de empresas e redes sociais e os principais desafios e limitações. O estudo apontou que a maioria das áreas estudadas neste contexto estão ligadas a aspectos técnicos. Como tal, o conceito de arquitetura foi ampliado para o chamado SoS ou plataforma da indústria, a fim de ajudar a compreensão da arquitetura em ECOS. Este tipo de plataforma fornece suporte a um conjunto de sistemas que precisam interagir para formar um SoS [Maier 1998]. Trata-se de sistemas complexos e interdisciplinares cujas funcionalidades e propósitos podem evoluir dinamicamente, englobando vários novos desafios a serem desenvolvidos.

Nesse sentido, os conceitos de SoS virtual e colaborativo têm sido discutidos no contexto da ECOS, permitindo a colaboração de diferentes sistemas e organizações para produzir funcionalidades emergentes. Em SoS colaborativo e virtual, o ECOS é mais valioso porque nessas categorias não há controle estrito sobre os sistemas constituintes. Numa perspectiva de questões sociais e de negócios, um ECOS fornece uma visão organizacional complementar ao desenvolvimento de SoS, que introduz papéis e regras de interação, colaboração e capacidades sinérgicas para seus sistemas constituintes. A partir desta discussão é possível confirmar a existência de muitas semelhanças entre as características e desafios de SoS [Maier 1998] e ECOS [Bosch 2009], que foram levantados na pesquisa realizada por Santos et al. (2014).

Conforme mencionado, a evolução da plataforma depende diretamente das necessidades e contribuições emergentes da comunidade ECOS, bem como dos ajustes dos modelos de negócios híbridos subjacentes. Exige uma modelagem explícita dos papéis em diferentes organizações e das regras que regem suas interações internas e externas em relação a cada organização, por exemplo, quando uma organização colabora com terceiros independentes [Jeronimo Junior & Werner 2015]. Assim, o desenvolvimento evolutivo do SoS também deve levar em conta questões sociais, e não apenas as questões técnicas do ambiente. Nesse sentido, os modelos de arquitetura SoS devem ser estendidos para lidar com variáveis de contexto baseadas em cadeias de valor e redes sociais.

2.7. O Prenúncio dos Sistemas de Sistemas de Informação (SdSI)

Sistemas de Sistemas de Informação (SdSI) são um tipo particular de SoS formados por SIs, que foram formados devido à crescente tendência de cooperação entre empresas distintas, combinando esforços para oferecer funcionalidades mais complexas. Eles são uma especialização da ideia de Sistemas de Sistemas (SoS). Um SdSI pode ser criado dinamicamente através de alianças entre vários software de SI, interoperando para criar valor para seus proprietários e para clientes que se beneficiam da parceria resultante. Exemplos dessa tendência são as Organizações Virtuais, que compreendem várias organizações distintas que se reúnem espontaneamente, trabalhando em cooperação (incluindo combinando seus SI) no contexto de um projeto executado por tempo pré-determinado. Movimentos como Clean Web, em que software de rede social e tecnologia da informação são articulados para resolver problemas relacionados com restrições de recursos naturais, também representam tendências em SdSI [Graciano Neto 2016]. Processos de negócio flexíveis e inter-organizacionais podem ser estabelecidos entre SI constituintes de um SdSI [Graciano Neto et al. 2017a].

O termo SdSI apareceu na década de 1990 [Carlsson e Stankiewicz 1991], sendo lembrado recentemente [Majd et al. 2015] [Saleh e Abel 2015]. Sob essa perspectiva, a SdSI exibe uma forte natureza empresarial, Saleh e Abel resumem que o SdSI deveria [Saleh e Abel 2015]: (i) estar preocupado com o fluxo de informação e conhecimento entre diferentes IS; (ii) abordar o impacto das inter-relações entre diferentes SdSI (SdSI como constituintes); (iii) ser responsável pela geração de informações do SdSI emergente; e (iv) abordar a interoperabilidade da informação como uma questão-chave. SdSI são formados por vários IS que combinam suas capacidades. As organizações virtuais são uma potencial instância de negócios suportada pelo SdSI. Eles compreendem várias organizações distintas que se reúnem espontaneamente, trabalhando cooperativamente (incluindo seus sistemas) no contexto de um projeto específico por um período de tempo bem definido, como seis meses ou dois anos, deixando o SdSI depois disso [Graciano Neto et al. 2017b].

A interconexão de SIs para conseguir uma funcionalidade mais complexa não é um tópico novo [Carlsson e Stankiewicz 1991] [Wiederhold 1992] [Breschi e Malerba 1997]. De fato, esses arranjos de SI foram construídos ao longo das últimas décadas com diferentes nomes e finalidades distintas. Sistemas Complexos, Sistemas Ultra-Grandes e Sistemas de Informação Federados [Graciano Neto et al. 2014] são alguns exemplos que emergiram ao longo das décadas para representar classes de sistemas intensivos em software formados por um conjunto de SI. A principal diferença entre um SdSI e esses outros tipos de sistemas é o nível de independência de seus sistemas constituintes. O constituinte ainda exibe uma operação independente ao não contribuir para a realização de missões globais [Falkner et al. 2016], enquanto SI Federados e outras classes que combinam SI muitas vezes têm SI exclusivamente dedicados aos objetivos dos sistemas maiores.

SdSI herda peculiaridades de SoS. SoS (e SdSI como consequência) não devem ser, necessariamente, uma estrutura permanente, mas um fenômeno desencadeado por algum estímulo. Em outras categorias de sistemas que são montados com sistemas pré-existentes, esses sistemas são projetados para serem integrados como parte de um novo

sistema complexo ou são projetados/refatorados para serem parte permanente de um sistema maior. Constituintes não são necessariamente concebidos para fazer parte de um SdSI, e eles também têm uma existência independente. Nesse sentido, há requisitos distintos que devem ser cumpridos em relação aos constituintes: eles devem ser projetados para interoperar entre eles (como outros grandes sistemas semelhantes fazem), mas eles também precisam ter uma existência independente. No entanto, SdSI deve ser confiável, bem como SoS. Eles são obrigados a lidar com a dinâmica da arquitetura, exigindo que outro constituinte trabalhe quando falhar e auto-adaptando sua estrutura para manter a realização de uma missão mesmo quando eventos externos ameaçam a estabilidade do SdSI [Graciano Neto et al. 2017b]. Abordagens dirigidas a processos de negócio (BPM) têm sido investigadas devido a seu potencial para complementar as notações de especificação em missões (tais como mKAOS) em SdSI [Graciano Neto et al. 2017].

Exemplos incluem a abordagem de Tu et al., em que os autores relatam a criação de um SdSI formado por sistemas de informação empresariais [Tu et al., 2011], como destacado em uma revisão de literatura que versa sobre o uso de modelos e abordagens dirigidas por modelos em SoS [Graciano Neto et al. 2014]. Organizações Virtuais também se valem de SdSI para possibilitar seu trabalho.

2.8. O Advento das Relações entre ECOS e SdSI (EcoSdSI)

SI são frequentemente definidos em termos de software, hardware, pessoas, processos, políticas etc. Cada SI tem esses elementos intrínsecos e os seus limites são, às vezes, difíceis de identificar. Geralmente eles funcionam apoiados em software. Por sua vez, Ecossistemas de Software (ECOS) também costumam ser definidos em termos de software (a plataforma), hardware, pessoas, políticas etc. Desta forma, vemos uma linha tênue entre os dois conceitos e conjecturamos que, inerentemente, cada SI tem em volta de si, um ECOS próprio.

Visivelmente, SdSI tem uma forte natureza empresarial e desenvolvimento dos constituintes se torna orientado a negócio. O desafio aqui é como lidar com a complexidade de descrever, desenvolver e operar o SdSI e um ECOS que emerge dele, considerando seus atributos intrínsecos – complexidade da estrutura, abertura, necessidade de equilíbrio e regulação e diferentes níveis de previsibilidade do comportamento – e alcançar seus objetivos de negócios desejados.

Devido à sua natureza, o SdSI detém um negócio completo, envolvendo fornecedores, clientes, parceiros e plataformas tecnológicas que atuam como uma tecnologia principal que apoia esse negócio. Em seguida, percebemos que a associação entre diferentes SI gera diretamente a associação entre suas respectivas ECOSs, criando um ECOS principal/novo que compreende o comportamento emergente resultante da associação de seus diferentes objetivos de negócios em um novo e comum.

As ligações de interoperabilidade entre as diferentes SI devem ser estabelecidas para criar novas funcionalidades e para explorar ou criar oportunidades de negócio. Isso acontece como resultado de alianças inter-organizacionais e cooperação. Suponha que cada uma das organizações que compõem este consórcio possua uma plataforma de SI diferente. O SdSI lida apenas com os aspectos técnicos. Outros elementos notáveis, como metas de negócios, agentes, cadeias de valor e relações produção/consumo não são

cobertos pela dimensão técnica SdSI. Como resultado, conjecturamos que, para o SdSI, emergem dimensões empresariais e sociais, criando, por definição, um ECOS que envolve cada SI que está incluído no contexto de um SdSI. De fato, estes ECOSs estão relacionados pela interoperabilidade que existe entre cada constituinte. Como resultado, esse fenômeno cria um ECOS inteiro que envolve todo o SdSI e envolve outros ECOSs internos que são inseridos nesse contexto. Assim, criamos como consequência um EcoSdSI (ou SdSI ECOS), isto é, *um ECOS formado pelos ECOSs internos e que envolvem cada constituinte que compõe um SdSI*.

Um exemplo para o exercício da visão EcoSdSI está presente no domínio do governo e da democracia eletrônica. As instituições públicas têm diferentes SI e bases de dados que precisam ser integradas para fornecer serviços eficazes aos cidadãos; por exemplo, os SI de identificação civil podem ser integradas ao SI da polícia e/ou ao sistema de saúde pública para criar um SdSI com funcionalidades inovadoras. Além disso, cada SI pode ser mantido dentro de um ECOS específico envolvendo fornecedores, desenvolvedores, legislação, auditores, cidadãos etc. Suas relações variam dependendo das necessidades de negócio e/ou de novos procedimentos ou mudanças na legislação. As exigências de transparência e responsabilização colocam um agente diferente neste ecossistema: o cidadão. Adicionalmente, companhias poderiam desenvolver novos SI capazes de acessar dados abertos fornecidos por instituições públicas para fomentar a participação cidadã na tomada de decisões pública, integrando transparência e participação efetiva da população na governança pública. Neste sentido, um SdSI formado pelos SI individuais que dão acesso a estas novas funcionalidades constitui todo um novo ECOS, formando o que concebe-se como EcoSdSI, explorando o ecossistema anterior para culminar em novos comportamentos, resultados e inovação. Olhar esta cena sob a perspectiva do EcoSdSI pode ajudar a desenvolver uma visão mais ampla desses arranjos do ecossistema, potencial de novos negócios, comportamentos emergentes e inovação.

2.9. Considerações Finais

Este trabalho apresentou os fundamentos de conceitos importantes para o desenvolvimento de SI nos próximos anos: SoS e ECOS. Adicionalmente, as discussões foram estendidas, cobrindo as potenciais relações entre ambos os tópicos, o advento dos SoS formados por SI, chamados de SdSI, bem como as possíveis relações que poderão ser estabelecidas entre os campos de investigação de ECOS e SdSI. Exemplos de cada um dos elementos principais foram apresentados e uma provocação foi feita sobre a possível caracterização das relações entre os elementos considerados.

Agradecimentos

Agradecemos em particular aos pesquisadores profa. Dra. Elisa Yumi Nakagawa, prof. Dr. Flavio Oquendo, e Ma. Lina Garcés pela colaboração e insumos provenientes de trabalhos que precedem este material. Ademais, o primeiro autor agradece pelas explanações vanguardistas dos dois primeiros pesquisadores sobre muitos dos conceitos apresentados, uma vez que estes podem ser considerados pioneiros na investigação, contribuição científica e avanços em SoS e SdSI no Brasil e no mundo.

2.10. Referências

- Araújo, R., Magdaleno, A. (2015) “Ecosistemas Digitais para o Apoio a Sistemas de Governo Abertos e Colaborativos”. In: Anais do XI Simpósio Brasileiro de Sistemas de Informação, Goiânia, Brasil.
- Bertalanffy, L.V. (2015) “General System Theory: Foundations, Development, Applications”. George Braziller Inc.
- Boardman, J., Sauser B. (2006) “System of systems - the meaning of of”. In: Proceedings of the SOSE, Los Angeles, USA.
- Boley, H., Chang, E. (2007) “Digital Ecosystems: Principles and Semantics”.
- Bosch, J. (2009) “From Software Product Lines to Software Ecosystem”. In: Proceedings of 13th International Software Product Line Conference, San Francisco, USA.
- Breschi, S., Malerba, F. (1997) “Sectoral innovation systems: technological regimes, schumpeterian dynamics, and spatial boundaries”. In: Proceedings of the DRUID Conference on National Innovation Systems, Industrial Dynamics and Innovation Policy, Rebuild, Denmark, pp. 130-156.
- Bryans, J., Payne, R., Holt, J., Perry, S. (2013) “Semi-formal and formal interface specification for system of systems architecture”. In: Proceedings of the SysCon 2013, Orlando, USA, pp. 612-619.
- Carlsson, B., Stankiewicz, R. (1991) “On the nature, function and composition of technological systems”. *Journal of Evolutionary Economics* 1(2):93-118.
- Dahmann, J.S., Jr., G.R., Lane, J.A. (2008) “Systems engineering for capabilities”. *CrossTalk Journal - The Journal of Defense Software Engineering* 21(11):4-9.
- Dahmann, J., Baldwin, K.J., Jr., G.R. (2009) “Systems of systems and net-centric enterprise systems”. Technical Report, MITRE Corporation, Loughborough University (<http://www.acq.osd.mil/se/docs/SoS-and-NC-Systems-CSER2009.pdf>).
- Dhungana, D. et al. (2010) “Software Ecosystems vs. Natural Ecosystems: Learning from the Ingenious Mind of Nature”. In: Proceedings of the European Conference on Software Architecture (Companion Volume), Copenhagen, Denmark, pp. 96-102.
- Falkner, K., Szabo, C., Chiprianov, V., Puddy, G., Rieckmann, M., Fraser, D., Aston, C. (2016) “Model-driven performance prediction of systems of systems”. *Journal on Software and Systems Modeling*, Springer.
- Graciano Neto, V.V., Guessi, M., Oliveira, L.B.R., Oquendo, F., Nakagawa, E.Y. (2014) “Investigating the model-driven development for systems-of-systems”. In: Proceedings of the 2nd International Workshop on Software Engineering for Systems-of-Systems, Vienna, Austria, pp. 22:1-22:8.
- Graciano Neto, V.V. (2016) “Validating emergent behaviours in systems-of-systems through model transformations”. In: Proceedings of the ACM SRC-MODELS, St. Malo, France.

- Graciano Neto, V.V. (2017) “A model-based approach towards the building of trustworthy software-intensive systems-of-systems”. In: Proceedings of the ICSE 2017 (Companion Volume), Buenos Aires, Argentina. (To appear).
- Graciano Neto, V.V., Cavalcante, E., El-Hachem, J., Santos, D. (2017a) “On the Interplay of Business Process Modeling and Missions in Systems-of-Information Systems”. In: Proceedings of the SESoS/WDES@ICSE 2017, Buenos Aires, Argentina. (To appear).
- Graciano Neto, V.V., Paes, C.E., Garcés, L., Guessi, M., Oquendo, F., Nakagawa, E.Y. (2017b) “Stimuli-SoS: A model-based approach to derive stimuli generators in simulations of software architectures of systems-of-systems”. Journal of the Brazilian Computer Society. (Submitted).
- IEEE (2007) “Inaugural IEEE International Digital Ecosystems and Technologies Conference: 2007” (<http://www.ieee-dest.curtin.edu.au/2007/index.php>).
- Jansen, S., Finkelstein, A., Brinkkemper, S (2009) “A sense of community: A research agenda for software ecosystems”. In: Proceedings of the 31st International Conference on Software Engineering, Vancouver, Canada, pp. 187-190.
- Jeronimo Junior, H., Werner, C. (2015) “A Systematic Mapping on the Relations between Systems-of-Systems and Software Ecosystems”. In: Anais do CBSOft – WDES 2015, Belo Horizonte, Brasil, pp. 65-72.
- Lane, J.A. (2013) “What is a System of Systems and Why Should I Care?”. Technical Report, University of Southern California.
- Laudon, K.C., Laudon J.P. (2016) “Management Information Systems”. Pearson Education.
- Maier, M.W. (1998) “Architecting principles for systems-of-systems”. Systems Engineering 1(4):267-284.
- Majd, S., Marie-Hélène, A., Alok, M. (2015) “An Architectural Model for System of Information Systems”. In: OTM 2015 workshops, Rhodes, Greece, pp. 411-420.
- Manikas, K. (2016) “Revisiting Software Ecosystems Research: A Longitudinal Literature Study”. The Journal of Systems and Software 117(2016):84-103.
- Mittal S., Rainey, L. (2015) “Harnessing Emergence: The Control and Design of Emergent Behavior in System of Systems Engineering”. In: Proceedings of the SCSI, San Diego, USA, pp. 1-10.
- Moore, J.F. (1996) “The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems”. Harper Business.
- Nielsen, C.B., Larsen, P.G., Fitzgerald, J., Woodcock, J., Peleska, J. (2015) “Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions”. ACM Computing Surveys 48(2):18:1-18:41.
- Oquendo, F. (2016) “Formally Describing the Software Architecture of systems-of-systems with SosADL”. In: Proceedings of the International Conference on Systems-of-Systems Engineering (SOSE), Kongsberg, Norway, pp. 1-6.

- Paes, C.E.B., Graciano Neto, V.V., Oquendo, F., Nakagawa, E.Y. (2016) “Experience report and challenges for systems-of-systems engineering: a real case in the Brazilian defense domain”. In: Anais do CBSOft – WDES 2016, Maringá, Brasil.
- Pérez, J., Díaz, J., Garbajosa, J., Yague, A., Gonzalez, E., Lopez-Perea, M. (2013) “Large-scale smart grids as system of systems”. In: Proceedings of the First International Workshop on Software Engineering for Systems-of-Systems, New York, USA, pp. 38-42.
- Ramos, A.L., Ferreira, J.V., Barcelo, J. (2012) “Model-based Systems Engineering: An Emerging Approach for Modern systems”. IEEE Transactions On Systems Man Cybernetics Part C-applications Rev. 42(1):101-111.
- Saleh, M., Abel, M.H. (2015) “Information Systems: Towards a System of Information Systems”. In: Proceedings of the 7th International Conference on Knowledge Management and Information Sharing, Lisbon, Portugal, pp. 193-200.
- Santos, R., Werner, C., Barbosa, O., Alves, C. (2012) “Software ecosystems: trends and impacts on software engineering”. In: Proceedings of the XXVI Brazilian Symposium on Software Engineering, Natal, Brazil, pp. 206-210.
- Santos, R., Gonçalves, M., Nakagawa, E.Y., Werner, C. (2014) “On the relations between systems-of-systems and software ecosystems”. In: Anais do CBSOft – WDES 2014, Maceio, Brasil, v. 2, pp. 58-62.
- Silva, E., Batista, T., Cavalcante, E., Oquendo, F. (2016) “Bridging Missions and Architecture in Software-Intensive Systems-of-Systems”. In: Proceedings of the ICECCS 2016, pp. 201-206.
- Smith, T.M., Smith, R.L. (2012) “Elements of Ecology”. Benjamin Cummings.
- Tu, Z., Zacharewicz, G., Chen, D. (2011) “Harmonized and reversible development framework for HLA based interoperable application”. In: Symposium on Theory of Modeling and Simulation (M&S), San Diego, USA.
- Wiederhold, G. (1992) “Mediators in the architecture of future information systems”. Computer 25(3):38-49.