Serviços em Redes Futuras: Objetivos, Tecnologias e Comparação de Iniciativas

Rodrigo Carneiro Brandão[1]
Agostinho Manuel Vaz[2]
Antônio Marcos Alberti[2]

Resumo: O setor de serviços, também chamado em economia como, terceiro setor, é o novo motor da economia mundial. Estima-se que mais de 50% do produto interno bruto dos países seja derivado desse setor. A evolução tecnológica do final dos anos 90 levou o conceito de serviços para a web, através dos chamados *Web Services*. Com o uso cada vez maior da Internet nos mais diversos ramos, houve a necessidade de se ampliar o conceito de serviço na rede para um nível superior, não apenas técnico ou restrito a tecnologia de informação, mas sim trazer para a rede, também, os serviços do mundo real ou serviços do dia-a-dia. A Internet de Serviços visa o desenvolvimento de novas arquiteturas para o mercado atual e futuro de serviços na Internet. Assim, este artigo discute o papel dos serviços nas redes futuras, apresentando as tecnologias chaves para esse fim, bem como comparando algumas iniciativas que tem se destacado neste cenário.

Palavras chave — Serviços, Internet do Futuro, Arquiteturas Orientadas a Serviços, Composição Dinâmica, Semântica, Internet de Serviços, Computação Autonômica.

Abstract*: The service sector, also known in economy as the third sector, is the new engine of the world's economy. It is estimated that over 50% of gross domestic product of countries is derived from this sector. The technological developments of the late '90s led to the concept of Internet services through the so-called Web Services. With the increasing use of the Internet in various branches of the economy, there was a need to extend the concept of service on the network to a higher level, not just restricted to information technology experts, but to people in general. The Internet of Services aims to develop new architectures for current and future market of networked services. Thus, this paper discusses the role of services in future networks, presenting the key technologies for this purpose and comparing some of the initiatives that have been highlighted in this scenario.*

Keywords — Services, Future Internet, Service Oriented Architectures, Dynamic Composition, Semantics, Internet of Services, Autonomic Computing.

[1] Instituto Federal de Educação, Ciência e Tecnologia do Norte de Minas Gerais – IFNMG. Rodovia BR 367, km 07, S/N – Almenara, MG, Brasil. 39900-000 {rodrigo.brandao@ifnmg.edu.br}.

[2] Instituto Nacional de Telecomunicações - Inatel. Av. João de Camargo, 510 - Santa Rita do Sapucaí, MG, Brasil. 37540-000 {agostinhov28@hotmail.com; alberti@inatel.br}.

1 Introdução

Desde o surgimento da Internet, esta tem crescido e se popularizado rapidamente. No ano 2000, a Internet contava com aproximadamente 361 milhões de usuários e em junho de 2012 com pouco mais de 2,4 bilhões de usuários, que correspondem a aproximadamente 34,3% da população mundial [1].

Este crescimento se deve ao fato de que o modelo da Internet atual foi projetado utilizando o paradigma *host-centric,* sendo o núcleo da arquitetura composto por terminais identificados com endereço IP (*Internet Protocol*). Este modelo facilitou e acelerou o crescimento da Internet, uma vez que não é necessário modificar o núcleo da rede para a criação de novas aplicações. Em contrapartida, esta simplicidade praticamente "engessa" a arquitetura, tornando difícil resolver problemas estruturais como escalabilidade, mobilidade, flexibilidade, confiabilidade e gerenciamento. Habitualmente, problemas deste tipo contam com soluções paliativas que são "remendadas" a arquitetura atual, dificultando ou até mesmo impossibilitando soluções mais significativas.

A Internet afetou e tem afetado a sociedade de forma significativa e irreversível, provocando mudanças na forma como as pessoas trabalham, estudam e se divertem. Mudanças que podem ser percebidas quando observamos a grande variedade de serviços oferecidos, tais como vídeos interativos, redes sociais, jogos 3D, transmissão de voz e TV sobre redes IP denominados *Voice over IP* (VoIP) e *Television over IP* (TVoIP), entre muitos outros.

É perceptível a transição de uma economia baseada em produtos para uma economia baseada em serviços. Todos os países desenvolvidos apresentam o setor de serviços como um setor em evidência, seja na geração de empregos, atuando como diferencial competitivo ou suporte às atividades de manufatura.

A tendência é que a Internet atenda as demandas desta economia, exercendo um papel cada vez mais decisivo em todos os processos de negócio e se torne uma ferramenta de produtividade por excelência [2]. No entanto, o modelo atual não é flexível a ponto de permitir a criação e adaptação de serviços para atender os processos de negócios de forma satisfatória, rentável e com baixo custo.

Além das dificuldades deparadas, a utilização da Internet pelos usuários está relacionada em sua maioria a processos de descoberta, negociação, contratação e utilização de serviços e conteúdos. Fatores que contribuem para que outras propostas com abordagens projetadas a partir do zero e que não são obrigatoriamente acopladas às redes IP atuais ganhem força. Estas arquiteturas são baseadas na abordagem *Clean Slate*.

De acordo com a Universidade de Stanford [3], a abordagem *Clean Slate* para a Internet é o mesmo que "reinventar a Internet". Esta "Nova Internet" ou "Internet do Futuro" deverá superar as limitações da Internet atual, incorporar novas tecnologias, permitir a criação de novas classes de serviços e aplicações, bem como, ser uma plataforma que suporte inovações e atenda as demandas atuais e futuras [41].

Em tempos de mudanças rápidas e um ambiente de negócios extremamente competitivo é preciso desenvolver alternativas que possibilitem prover serviços adequados e adaptativos. Desta forma, a abordagem *clean slate* propõe substituir a atual arquitetura por uma mais adequada, que possibilite modelar e representar os serviços através da Internet, a chamada Internet de Serviços. Ela tem como objetivo pesquisar e desenvolver novas teorias, modelos e tecnologias para prover soluções eficientes e eficazes que permitam a todo tipo de usuário, comercializar e consumir serviços disponíveis na Internet [4]. Por meio da Internet de Serviços as organizações e usuários poderão facilmente encontrar sistemas expostos como serviços, combiná-los e adaptá-los a seu contexto específico.

A estrutura orientada a serviços remete ao paradigma de Computação Orientada a Serviços, que utiliza os serviços como elemento fundamental para o desenvolvimento de aplicações de forma rápida, com baixo custo e em ambientes distribuídos [5]. A Arquitetura Orientada a Serviços utiliza essa computação para desenvolver sistemas e aplicações com base em serviços independentes, reutilizáveis e fracamente acoplados, a fim de orquestrar serviços que sejam capazes de atender processos de negócios específicos e se adaptar a um mercado de negócios dinâmico, complexo e competitivo.

Este artigo faz uma revisão dos principais conceitos, objetivos e tecnologias que fundamentam a Internet de Serviços (Seção 2). Ele também apresenta e discute algumas das iniciativas que tem se destacado nesse ambiente (Seção 3), bem como alguns dos requisitos e desafios por de trás da Internet de Serviços (Seção 4). Por fim, apresenta na Seção 4 algumas considerações finais.

2 Ambientes Orientados a Serviços: Conceitos, Objetivos e Tecnologias

2.1 Alguns Conceitos

Um serviço é um recurso abstrato que representa uma capacidade de realizar tarefas específicas. É normalmente executado quando necessário, mas permanece inativo até que seja solicitado. Novos serviços podem ser oferecidos através da combinação de serviços existentes [6]. Nos correios, por exemplo, pode ser utilizado o serviço de transporte oferecido por outras empresas para enviar documentos e encomendas entre um remetente e um destinatário.

Os termos mais relevantes associados com o conceito de serviços e introduzidos ao longo do tempo pela comunidade científica e industrial [4][37] são: Serviços do Mundo Real, *E-Service*, *Web Service* e Internet de Serviços. A Figura 1 adaptada de [4] ilustra a trajetória histórica desses conceitos, que podem ser definidos como:

- **Serviços do Mundo Real:** É o termo usado para se referir a qualquer tipo de serviço que pode ser encontrado na sociedade. Devido sua diversidade e heterogeneidade, tem sido tradicionalmente difícil definir seu significado. Um serviço do mundo real pode ser defi-

nido como qualquer atividade ou benefício que uma parte possa oferecer a outra, e que seja basicamente intangível, podendo sua produção estar ou não vinculada a um produto físico [7]. Por exemplo, qualquer pessoa que exerça uma tarefa distinta no apoio a outra, está provendo um serviço. Da mesma forma, uma organização que realiza tarefas associadas com seu negócio, também está provendo um serviço.

- ***E-Service*:** O termo "*E-Service"* (abreviação de *Electronic Service*) é usado para definir serviços que utilizam redes de dados como um canal de comunicação, permitindo a interação entre consumidores e serviços remotos. Praticamente, qualquer serviço pode ser transformado em um *e-service,* desde que ele possa ser invocado através de uma rede de dados. *E-services* são independentes da linguagem de especificação usada para definir a sua funcionalidade, interface ou propriedades não funcionais, e, tal como acontece com os serviços do mundo real, sua definição é bastante ampla [4].

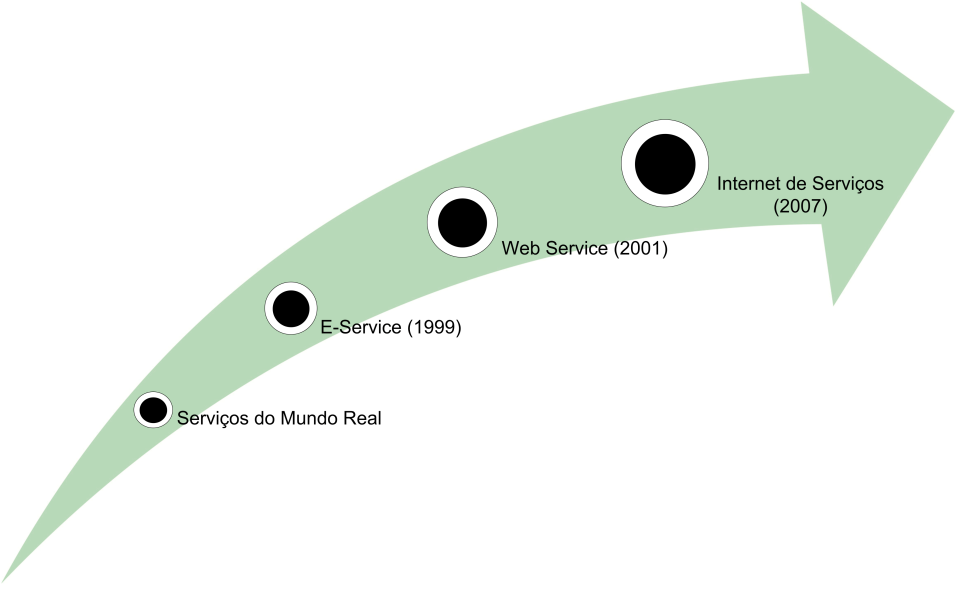

Figura 1. Termos associados com o conceito de serviços.

- ***Web Service:*** Permite que diferentes aplicativos de *software* se comuniquem facilmente, independente da linguagem de programação ou plataforma de computação. Esta tecnologia possibilita que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Cada aplicação pode ter a sua própria linguagem de programação, que é traduzida em uma linguagem universal, o formato XML (*Extensible Markup Language*). O *Web service* é o componente que permite às aplicações enviar e receber dados neste formato.

- **Internet de Serviços:** É o termo designado para identificar os serviços oferecidos através da Internet. Duas propriedades diferenciam os serviços baseados na Internet de Serviços (*IoS - Internet of Services*) dos demais tipos de serviços já apresentados [4]: a primeira é que a noção de serviço não se limita apenas aos serviços relacionados à Tecnologia de Informação (TI), mas também aos serviços do mundo real. E, a segunda, é que as partes interessadas no fornecimento e consumo de serviços não se restringem apenas aos profissionais de TI, mas a todos os usuários da rede. Desta forma, os serviços baseados na IoS podem ser utilizados diretamente pelos consumidores, e, invocados por sistemas técnicos para acessar funcionalidades de negócio oferecidas remotamente por provedores de serviços. Observe ainda que a Internet de Serviços pode ser considerada parte integrante da chamada Internet do Futuro [2][41].

2.2 Computação Orientada a Serviços

A Computação Orientada a Serviços (SOC – *Service-Oriented Computing*) é o paradigma de computação que utiliza serviços como o elemento básico para o desenvolvimento e composição de aplicações distribuídas, mesmo em ambientes heterogêneos. Serviços são autônomos, independentes de plataforma computacional, fracamente acoplados, podendo ser descritos, publicados, descobertos e dinamicamente combinados. A criação de serviços reflete uma programação orientada a serviços, baseada na ideia de compor aplicações através da descoberta e invocação de serviços disponíveis na rede, ao invés de desenvolver novas aplicações ou utilizar aplicações disponíveis para realizar alguma tarefa específica [5].

Os serviços podem ser oferecidos por diferentes provedores de serviço, se comunicar através da Internet e serem utilizados por clientes para atender desde tarefas simples a complexos processos de negócios. Para que seja possível entender exatamente o que isso significa, a Figura 2 ilustra os principais princípios e objetivos da computação orientada a serviços. São eles:

- **Flexibilidade:** O cenário de negócios tradicional exige que as empresas alterem continuamente seus sistemas para atender as mudanças do mercado, como por exemplo, novas exigências e requisitos [8]. SOC permite um melhor alinhamento entre a TI e os negócios, com a criação de novos serviços e a reutilização dos serviços já existentes, a fim de atender de forma eficiente e eficaz as mudanças do mercado [9].

- **Fraco acoplamento:** Os serviços são fracamente acoplados por natureza. Isto é, são executados em diferentes plataformas, implementados de forma independependente e possuem diferentes proprietários [10]. O acoplamento fraco libera os clientes de qualquer conhecimento da estrutura interna dos serviços, e, os clientes não devem exigir que o serviço conheça o contexto que será utilizado [5]. A ideia de serviços fracamente acoplados é evitar que as modificações, falhas ou até mesmo, exclusões de serviços causem conflitos a outros serviços, devendo estes ser tolerantes a falhas, flexíveis e escaláveis.

- **Localização transparente:** Os serviços devem ter as suas definições e informações de localização armazenadas em um repositório que seja acessível a uma variedade de clientes, possibilitando que o serviço seja localizado e invocado, independentemente de onde será executado [5]. Esta habilidade permite clientes e provedores de serviços agirem independentes de suas localizações, podendo, por exemplo, um cliente descobrir e utilizar um serviço sem saber onde seu provedor está localizado.

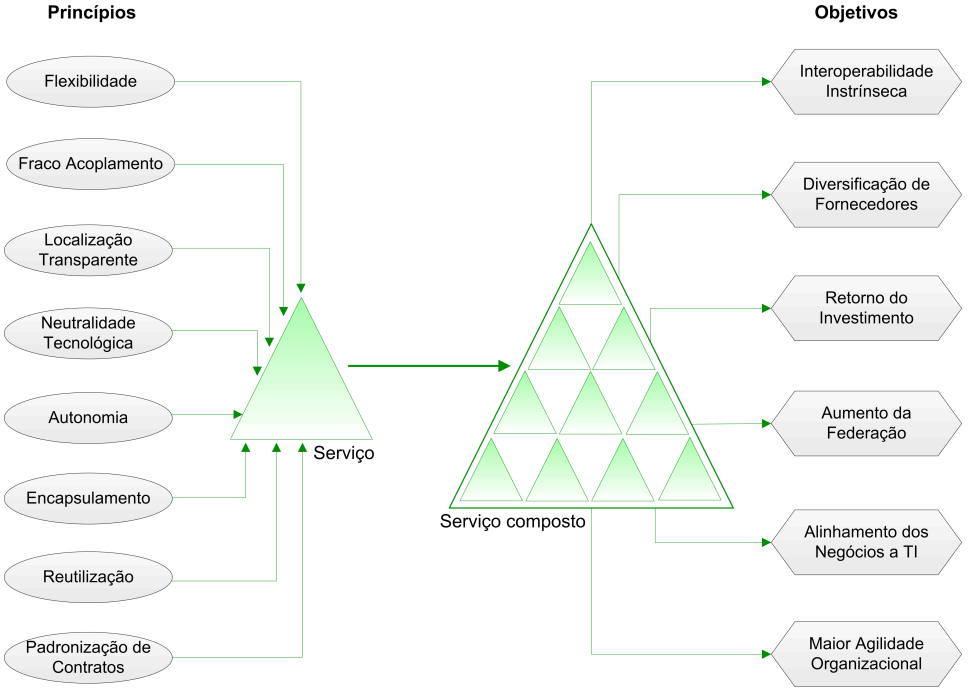

Figura 2. Princípios e objetivos do paradigma de computação orientada a serviços.

- **Neutralidade tecnológica:** Serviços devem ser invocados por meio de tecnologias padronizadas, disponíveis em praticamente todos os ambientes de TI. Isto implica que os mecanismos de invocação dos serviços, como descritores de serviços e mecanismos de descoberta, estejam de acordo com os padrões aceitos [5].

- **Autonomia:** Os serviços devem ser implantados, modificados e mantidos de forma independente um do outro, sendo o ciclo de vida de cada serviço independente do ciclo de vida do outro [11].

- **Encapsulamento:** Os detalhes da implementação interna dos serviços e sua estrutura de dados devem estar "escondidos". Há uma separação rigorosa da interface do serviço (o

que ele faz) da sua implementação (como é feito) [11].

- **Reutilização:** Permite que os serviços existentes sejam compartilhados e reutilizados para a criação de aplicações ou serviços compostos. Desta forma, os provedores e consumidores de serviços desfrutam de vantagens, como a redução no custo de investimento, personalização do *software*, e a redução do tempo que leva desde a concepção de uma aplicação até sua chegada ao mercado.

- **Padronização de contratos:** Os serviços expressam o seu propósito e capacidades através de contratos formais. Isto é, após as etapas de negociação (apresentadas na Subseção 4), faz-se necessário estabelecer um contrato que formalize o entendimento e expectativas das partes envolvidas (consumidor e provedor de serviços). A padronização de contratos é importante para assegurar que os parâmetros estabelecidos pelos serviços são coerentes, confiáveis e gerenciáveis [12]. A SOC propõe uma grande mudança de paradigma na maneira que as aplicações de *software* são concebidas, arquitetadas, disponibilizadas e consumidas.

- **Aumento da interoperabilidade intrínseca:** Interoperabilidade refere-se à partilha de dados. Este objetivo consiste em instituir a interoperabilidade nativa dentro dos serviços, a fim de reduzir a necessidade de integração. Isto possibilita que diferentes serviços possam ser repetidamente agrupados em uma variedade de composições para realizar uma série de processos de negócios [12].

- **Diversificação de fornecedores:** Diversificação refere-se à capacidade que o cliente tem de analisar diversos fornecedores e serviços, invocando o melhor serviço ou aquele que seja mais condizente com sua demanda. O cliente poderá diversificar o fornecedor sempre que julgar necessário.

- **Retorno do investimento (*ROI - Return on Investment*):** Mensurar o retorno do investimento de soluções automatizadas é uma questão crítica na determinação de quão rentável uma determinada aplicação ou sistema realmente é. Para muitas organizações, a sobrecarga financeira requerida pela TI é uma questão preocupante, uma vez que não contribui ativamente para aumentar o valor do negócio. Neste sentido, a computação orientada a serviços defende a criação de serviços que possam ser reaproveitados diversas vezes para automatizar e se adequar a diferentes processos de negócios, como parte de diferentes soluções [12].

- **Escalabilidade da federação:** Um ambiente de TI federado é aquele em que os recursos e aplicações estão unidos mantendo a sua autonomia individual e autogestão. Ambientes orientados a serviços ampliam esta federação através da implantação de serviços padronizados e combináveis, cada um dos quais encapsula um segmento da empresa [12].

- **Alinhamento dos negócios a TI:** O cenário de negócios tradicional exige que as empresas alterem continuamente seus sistemas para atender as mudanças do mercado, como por exemplo, novas exigências e requisitos [8]. SOC permite um melhor alinhamento entre a TI e os negócios, com a criação de novos serviços e a reutilização dos serviços já existentes, a fim de atender de forma eficiente e eficaz as mudanças do mercado [9].

- **Maior agilidade organizacional:** Ser capaz de adaptar-se rapidamente às mudanças da indústria e superar a estratégica da concorrência é um fator atraente para as empresas, especialmente as do setor privado. SOC possibilita esta adaptação por meio da criação de serviços padronizados e reutilizáveis e, portanto, independentes de processos de negócios [12].

2.3 Arquitetura Orientada a Serviços

A Arquitetura Orientada a Serviços (SOA – *Service-Oriented Architecture*) é um estilo de arquitetura de *software* cujo princípio fundamental é que as funcionalidades implementadas pelas aplicações sejam disponibilizadas na forma de serviços independentes, com interfaces bem definidas, que podem ser invocadas em sequência para formar processos de negócio [13], possibilitando fornecer serviços para usuários finais, usuários especialistas e serviços distribuídos em rede.

Para se construir uma SOA é preciso ter uma compreensão clara dos processos e domínios de negócios funcionais, incluindo decisões de como implementar funções de negócios como serviços separados. A modelagem e concepção de serviços dentro desta arquitetura requer abordagens mais sofisticadas que as utilizadas em aplicações tradicionais [14].

A arquitetura básica do SOA envolve o relacionamento entre três participantes: o Provedor do Serviço, o Consumidor do Serviço e o Serviço de Registro, conforme ilustrado na Figura 3 adaptada de [9].

O Consumidor do Serviço ou Cliente pode ser uma aplicação, um módulo de *software* ou outro serviço que esteja requerendo um serviço específico. O Provedor de Serviços é a entidade que cria um serviço, recebe e processa as solicitações de pedidos enviadas pelos Clientes; e por fim, o Serviço de Registro ou *Broker* é a entidade responsável por manter as publicações das descrições de serviços, facilitando sua descoberta por parte dos Consumidores.

Conforme ilustrado na Figura 3, o relacionamento entre os três participantes envolve basicamente as operações de publicação, descoberta e invocação. A publicação ocorre quando a descrição de um serviço é publicada junto ao Serviço de Registro, possibilitando que os Clientes possam descobrir e invocar os serviços de seu interesse. Descoberta acontece quando um solicitante de serviços localiza um serviço por meio do Serviço de Registro. Por fim, a invocação ocorre quando o Consumidor invoca o serviço cuja descrição tenha lhe interessado.

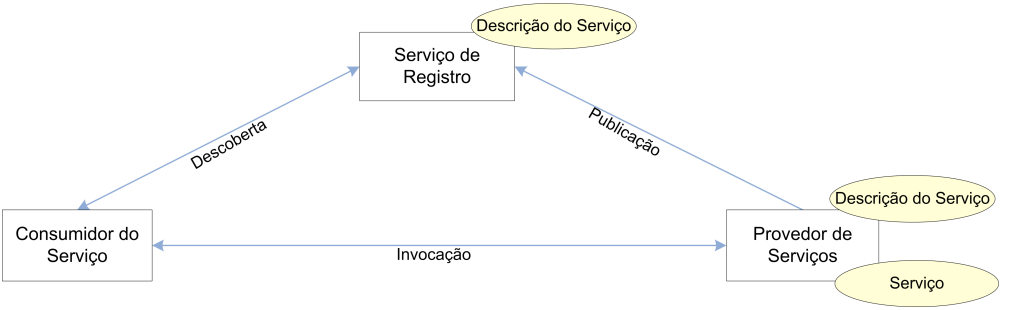

Figura 3. Arquitetura básica do SOA.

2.3.1 Ciclo de Vida SOA

O ciclo de vida representa a forma como os serviços são criados para atender a processos de negócio dinâmicos e específicos. O ciclo de vida SOA contempla quatro fases [15]: Modelagem, Composição, Implantação, e por fim, Execução e Monitoramento.

A modelagem inclui a análise e desenho dos processos de negócios, serviços, eventos e mensagens. Nesta fase, os serviços são modelados para executar uma função específica, podendo vir a se tornar parte de um serviço mais complexo, um serviço composto. Ainda nesta fase, é importante definir uma ontologia que forneça um vocabulário de domínio comum e conhecimento dos modelos de serviços de negócios [15].

A composição de serviços compreende as funcionalidades para criar ou reutilizar serviços já existentes (simples e compostos) e combiná-los. Não apenas os serviços simples, mas também os compostos podem ser utilizados como serviços básicos na composição de outros serviços [16]. Antes do fornecimento de serviços, deve existir uma etapa de negociação e ser estabelecido um SLA[3] entre as partes envolvidas [4].

A implantação envolve a criação de ambientes para a implantação e provisionamento de serviços. Os serviços devem ser implantados de forma integrada e executados em uma sequência que possibilite atender um processo de negócio composto [15].

Por fim, a execução e o monitoramento permitem que o serviço atenda os requisitos acordados, observando seu desempenho, garantindo a qualidade na execução e visando disparar eventos de adaptabilidade caso sejam necessários.

2.3.2 Negociação em Ambientes Orientados a Serviço

O modelo orientado a serviços propõe que o *software* seja desenvolvido e disponibilizado como um serviço, podendo ser entregue e utilizado sob demanda e não comercializado como um produto, como habitualmente é feito. Neste modelo, os elementos dos serviços são identificados, a prestação do serviço é negociada, o serviço é executado e então descartado.

[3] *Service Level Agreement*.Trata-se de um acordo entre partes que visa detalhar níveis de qualidade esperados para um serviço contratado, bem como cláusulas de ressarcimento em caso de quebra do acordo.

O fato de poder ser descartado é o que diferencia o "serviço" do "produto", podendo o serviço ser utilizado sob demanda e por diversos consumidores. A principal vantagem dessa abordagem é o fraco acoplamento entre as demandas de negócios e as soluções de *software*s associadas [17]. Entretanto, esse modelo exige a gerência do ciclo vida do software como serviço [35].

O conceito de disponibilizar *software* como serviço é relativamente simples: "Não compre *software*, apenas use-o como e quando precisar dele" [17]. Em um modelo orientado a serviços, o mesmo serviço pode ser oferecido com diferentes qualidades, preços ou condições de entrega, visando atender as diferentes demandas dos consumidores. Por exemplo, descritores de serviços podem ser disponibilizados contendo metadados que ajudem na negociação dos mesmos. Em *web services*, a descrição de serviços é feita usando o formato WSDL (*Web Services Description Language*), que é processável por máquinas [38].

Em ambientes dinâmicos, onde as ofertas e demandas estão em constante mudança, não é interessante fixar qualquer característica do serviço, como o preço, por exemplo, devendo estas características serem negociadas no momento que o serviço for solicitado.

Em um ambiente orientado a serviços, o mercado é considerado um elemento fundamental. Será no mercado que os provedores, consumidores e possivelmente uma terceira entidade, como, por exemplo, representantes de entidades que não estejam aptas a representar seus interesses, oferecem, solicitam e negociam serviços. Os elementos básicos de um mercado (digital ou físico) são os participantes, as mercadorias e as regras para a interação [17][39].

2.3.3 Processo de Negociação

O objetivo do processo de negociação é estabelecer um SLA entre o cliente e o provedor de serviços, a fim de viabilizar a prestação de serviço requerida pelo cliente ou oferecida pelo provedor [42]. O processo de negociação em um ambiente orientado a serviços envolve três aspectos principais: objetos de negociação, o protocolo de negociação e o modelo de decisão. Os objetos são os elementos sendo negociados, por exemplo, alguns parâmetros de qualidade. O protocolo refere-se à forma como as partes irão proceder na negociação, enquanto o modelo de decisão refere-se a como as partes irão decidir pelo estabelecimento ou não do contrato [42]. Com relação ao tempo, o processo de negociação pode ser dividido em três fases [17][42]: Pré-negociação, Negociação e Entrega do serviço.

Na pré-negociação o objetivo é o estudo e entendimento da negociação, devendo cada negociador planejar e elaborar a melhor alternativa para estabelecer um acordo. Esta fase envolve a seleção do serviço, seu provedor, e a previsão de uso do serviço selecionado. Mais de um protocolo de negociação e provedor podem ser escolhidos [42].

A seleção do serviço em ambientes orientados a serviço depende dos seus atributos funcionais e não funcionais. Funcionais são aqueles ligados diretamente ao objetivo da demanda e as funcionalidades esperadas pelo cliente. Não funcionais, são aqueles que não estão relacionados diretamente com a demanda, mas que são importantes para o desempenho do

serviço, padronização de processos e satisfação do cliente de uma forma geral, como por exemplo, preço e qualidade do serviço prestado [17].

Na seleção do provedor cada invocação de serviços pode envolver provedores distintos. Desta forma, Elfatatry e Layzell [17] propõe que o consumidor de serviços mantenha uma pequena lista de provedores de sua confiança e preferência, sendo esta atualizada frequentemente com base no *feedback* obtido nas fases de negociação e entrega dos serviços. Ismail et al. defende que a negociação pode ser simultânea com vários provedores [42]. Nesse caso, é necessária a coordenação das várias atividades de negociação, bem como mecanismo para decidir qual é a melhor composição de contratos.

Em um sistema baseado em serviços onde grande parte ou todas as funcionalidades são disponibilizadas como serviços, o grande número de negociações gera uma sobrecarga de comunicação, podendo afetar o desempenho do serviço. Desta forma, a opção proposta por [17] é que seja possível estabelecer contratos temporários assim que o serviço é invocado pelo cliente. Considere, por exemplo, um serviço de processador de texto. Neste caso, a aplicação se comunica com provedores de serviço de verificação ortográfica e negocia os termos e condições para que tal serviço seja inicializado assim que o usuário iniciar a edição de textos. Quando o usuário utiliza o editor de textos, o serviço de verificação ortográfica é invocado e o contrato é assinado [17].

A negociação do serviço é iniciada no momento em que o provedor e o consumidor de serviços trocam mensagens com o objetivo de estabelecer um acordo, que posteriormente será formalizado por meio de um contrato. Cada contrato contém a descrição do serviço negociado, além de uma série de condições do negócio. Ismail et al. apresenta um protocolo de negociação em que o requisitor inicia o processo de negociação, enquanto os provedores aguardam por requisições de contrato [42].

Em um mercado com vários serviços e provedores de serviços, a composição de funcionalidades é uma questão primordial e desafiadora. Diferentes blocos funcionais são fracamente acoplados. Desta forma, é importante levar em consideração os efeitos finais desta composição e qual será a qualidade da aplicação originada pela combinação de diferentes serviços [17]. Ismail et al. apresenta um cenário de negociação para *web services*, incluindo a composição de ofertas individuais [42].

Ao contrário de processos de negócios tradicionais que normalmente são executados de uma forma relativamente previsível e repetitiva, a composição de serviços lida com questões de incerteza e gestão de dependências em um ambiente dinâmico.

Em um ambiente onde o tempo é uma questão crítica, as negociações devem ser concluídas rapidamente. Dois mecanismos podem ajudar a alcançar este objetivo: protocolos de negociação e estratégias de negociação [17][42].

Os protocolos de negociação são usados para alinhar o comportamento das entidades envolvidas, de modo que não haja desperdício de tempo. As estratégias de negociação usam duas abordagens para alcançar uma melhor convergência da negociação: a heurística e a argumentação. A abordagem heurística utiliza diferentes técnicas de aprendizagem que per-

mitem a uma entidade gerar ofertas de serviços, que provavelmente serão aceitas, com base no histórico de negociação da outra entidade. A abordagem de argumentação depende do uso de técnicas para persuadir a outra parte a aceitar a proposta [17].

Como resultado da fase de negociação é estabelecido um contrato entre as partes envolvidas. O contrato descreve a funcionalidade do serviço, seus atributos não funcionais, bem como, as pré- e pós-condições do serviço e do contrato. As condições do serviço são de natureza técnica e descrevem os requisitos e consequências do uso de um serviço específico. As condições do contrato são de natureza empresarial e podem ser aplicadas a mais de um serviço [17], avaliando os resultados obtidos, o compromisso das partes envolvidas na negociação e a satisfação dos negociadores. Dado que o contrato foi estabelecido, a entrega ou prestação de serviços é realizada.

A confiança é um fator de destaque na prestação de serviços. Afinal, em um ambiente orientado a serviços, os consumidores podem decidir mudar de provedor sempre que for necessário [17]. Confiança constitui um elemento básico para se estabelecer parcerias, permitindo aos negociadores explorar novas possibilidades de contrato e correrem menores riscos, aumentando assim, a probabilidade de estabelecerem novos acordos.

3 Internet de Serviços: Algumas Iniciativas

3.1 CASCADAS

O *Component-ware for Autonomic Situation-aware Communications, and Dynamically Adaptable Services* (CASCADAS) [18][21] é um projeto do *Sixth Framework Programme* (FP6) que teve início em Janeiro de 2006 e término em Dezembro de 2008. Foi coordenado por Antonio Manzalini e seus custos elegíveis chegaram a € 6.920.795. O FP6 é um programa de pesquisa e desenvolvimento tecnológico da União Européia que tem como objetivo melhorar a integração e a coordenação da pesquisa na Europa financiando e promovendo a pesquisa e o desenvolvimento tecnológico.

De acordo com os idealizadores deste projeto, o principal objetivo do CASCADAS é desenvolver um ecossistema baseado em elementos autonômicos que permita a execução, composição e implantação de serviços inovadores, flexíveis e capazes de lidar com ambientes imprevisíveis. A proposta consiste em conceber uma "ecologia" de serviços no qual estes poderão prosperar e evoluir para satisfazer as necessidades específicas dos usuários.

CASCADAS prevê aplicações e serviços para interagir com usuários e ao mesmo tempo, avaliar as condições físicas e contextuais da rede. Para este fim, aplicações e serviços são desenvolvidos e implementados como *Autonomic Communication Elements* (ACEs) individuais ou agregados que dinamicamente se auto-organizam e interagem, a ponto de proporcionar a funcionalidade desejada.

Os ACEs são abstrações de *software* para serviços de comunicação autonômica, que se organizam autonomamente com outros ACEs e se adaptam às mudanças do ambiente para

atingir determinados objetivos. ACEs podem ser resumidos como sendo a base para a criação e evolução de serviços autonômicos em ambientes distribuídos [18].

Sua arquitetura é inspirada no modelo biológico e envolve órgãos altamente interconectados, responsáveis por tarefas específicas, que juntos mantêm a entidade "viva". Um ecossistema de ACEs possibilita fornecer serviços ou partes de um serviço a outros ACEs. Portanto, na proposta CASCADAS todos os serviços são implementados como um ou mais ACEs.

Cada ACE consiste de uma parte comum e uma parte específica. A parte comum contém seus órgãos e as funcionalidades comuns. A parte específica varia de ACE para ACE e compreende o Auto Modelo e as funcionalidades específicas de cada ACE. Desta forma, um desenvolvedor ao criar um ACE, só precisará de fato, criar a parte específica. A parte básica é sempre a mesma.

O projeto CASCADAS utiliza ACEs para criar uma rede voltada a síntese de conhecimento a partir de informações trocadas pelos ACEs através dos seus *gateways*. A rede de conhecimento (*KN - Knowledge Network*) é uma estrutura genérica que organiza o conhecimento distribuído independente do seu formato em um sistema que vai permitir que ele seja recuperado de forma eficiente [18][22].

3.2 BIONETS

O *BIOlogically-inspired NETworks and Services* (BIONETS) [24][25][27][29] é um projeto do *Sixth Framework Programme* (FP6) que teve início em Janeiro de 2006 e término em Março de 2010. Foi coordenado por Daniele Miorandi e seus custos elegíveis chegaram a € 6.948.333.

De acordo com os idealizadores deste projeto, a motivação para BIONETS veio das tendências da computação pervasiva e ambientes de comunicação caracterizados por vários dispositivos conectados em rede, como por exemplo, roupas, carros, eletrodomésticos, eletrônicos, entre outros. Essa grande diversidade de dispositivos contribui para o crescimento e complexidade da nova rede, superando a Internet atual e fazendo com que seja necessário se ter novas abordagens que tratem não só a heterogeneidade de recursos e nós, mas também os futuros problemas de escalabilidade e complexidade [24].

O objetivo do projeto é desenvolver uma rede completamente integrada, onde os serviços sejam capazes de se adaptar a um grande número de dispositivos heterogêneos, a mudanças de ambientes e evoluir de forma autônoma, sendo a arquitetura inteiramente distribuída e descentralizada, resistente à falhas de rede, ataques e desconexões, e, que os serviços sejam capazes de evoluir sem qualquer tipo de intervenção humana.

Em BIONETS, os serviços são definidos como entidades que podem fornecer o conhecimento, conteúdo ou funcionalidade para outros serviços e usuários. Dentro deste contexto, as aplicações de usuários, serviços de aplicativos, e serviços de protocolo podem ser definidos como composições de outros serviços ou células de serviços [25].

BIONETS descreve uma arquitetura para sistemas de computação autonômica[4] que pode ser dividida em três partes: Estrutura de Serviços, que inclui os serviços, e as funções de apoio à sua execução, distribuição, pervasividade e gestão. Estrutura de Interação, cuja finalidade é disponibilizar modelos de interação simultânea, apoiando a comunicação entre os diversos serviços distribuídos. Estrutura de Rede, que fornece os recursos básicos de comunicação.

De acordo com o paradigma BIONETS a evolução dos serviços é baseada em organismos digitais vivos, onde o nascimento é a criação do serviço, a reprodução é a evolução do serviço e a morte é a sua desativação. No ramo da biologia, evolução refere-se a mudanças das características hereditárias de uma espécie de uma geração para outra. Este processo faz com que surjam novas espécies mais adaptadas ao longo do tempo. Em termos de serviços, a evolução do serviço implica em uma adaptação de longo prazo para as mudanças no ambiente, podendo os serviços adquirirem novas funcionalidades perante o sistema [28] e [29].

3.3 XIA

A *eXpressive Internet Architecture* (XIA) [30] é um projeto que aborda a crescente diversidade de modelos de uso da rede. Foi iniciado pela *Carnegie Mellon University* e recentemente selecionado pela junta *de Computer and Information Science and Engineering* (CISE) da *National Science Foundation* (NSF) como parte do programa Future Internet Architecture (FIA).

O XIA é uma proposta de rede futura centrada no essencial (*principal-centric*). O centro da arquitetura é assim definido devido agrupar as abordagens essenciais pesquisadas para redes futuras, podendo uma entidade ser nomeada como um host, um domínio, um serviço, ou parte de um conteúdo específico.

De acordo com os idealizadores do projeto, os objetivos do XIA são preservar os pontos fortes da arquitetura centrada em *hosts*, melhorar a segurança e construir uma rede que seja capaz de suportar nativamente os modelos de redes centradas em conteúdo, serviço e *host*, e, que possa evoluir para suportar outros modelos de comunicação, como por exemplo, usuários e grupos.

Uma premissa desta arquitetura é que todos os objetos nela contidos sejam intrinsecamente seguros. A identificação destes objetos deverá ser realizada por meio de identificadores expressivos (XIDs) de 160 *bits* que serão utilizados como origem e destino para o encaminhamento de pacotes na rede. Para diferenciar significativamente as identificações dos objetos, são utilizados os termos HID, CID, e SID para se referir aos identificadores expressivos de *hosts*, conteúdos e serviços, respectivamente. A nomeação do conteúdo é baseada no *hash* do conteúdo; a nomeação do serviço e *host* é baseada no *hash* de suas res-

[4] **Computação Autonômica** se refere às características de autogestão dos recursos de computação distribuída, adaptação às mudanças imprevisíveis, enquanto esconde a complexidade dos usuários [26].

pectivas chaves públicas, utilizando assim, identificadores seguros que evitam, por exemplo, a falsificação de endereços na rede [30].

3.4 SOA4All

O *Service Oriented Archictetures For All* (SOA4All) ou Arquitetura Orientada a Serviço para Todos [31], é um projeto do *European Seventh Framework Programme* (FP7). Segundo seus idealizadores, apesar do sucesso inicial das pesquisas e desenvolvimento de arquiteturas orientadas a serviço (SOA) para apoio aos grandes prestadores de serviços, estas soluções permanecem, em sua maioria, especificas para cada prestador, distanciando-se cada vez mais do objetivo principal do paradigma de computação orientada a serviço, que é a criação de sistemas flexíveis e distribuídos globalmente, tornando-se assim, necessário repensar o SOA.

O SOA4All é o resultado deste esforço, e tem como o objetivo a materialização de um mundo onde bilhões de entidades expõem ou consomem serviços ou parte deles, através de tecnologias avançadas para a *web* [31][36]. O projeto ambiciona a criação de um *framework* e infraestrutura de *software* visando à integração do SOA com quatro avanços técnicos complementares e revolucionários que emergiram nas últimas décadas (*web*, *web* 2.0, *web* semântica e tecnologias cientes do contexto).

O SOA4All defende a ideia de que operações como, descoberta, seleção, composição, negociação e invocação de serviços, em ambiente com uma grande quantidade de serviços precisam de uma infraestrutura robusta e autogerenciável. Desta forma, é necessário fornecer descrições semânticas para os serviços disponíveis, criar localizadores de serviços com base nestas descrições e criar, ainda, mecanismos de mapeamento para permitir a invocação destes serviços. É importante que as descrições e serviços estejam mapeados corretamente para permitir a localização e invocação dinâmica dos serviços [31].

No SOA4ALL, existe o *Framework* de Adaptação Contextual do Serviço que fornece uma plataforma genérica de apoio à adaptação dos serviços nos diversos contextos, de acordo com suas informações relevantes. É importante observar que a informação chave para o contexto depende de cada serviço. Por exemplo, a localização do consumidor pode ser a informação contextual chave para alguns serviços ou considerada uma informação totalmente irrelevante para outros.

O SOA4ALL incorpora contextos no SOA como uma forma de personalizar o consumo e prestação de serviços em escala global, personalizando o consumo de um serviço a partir da perspectiva do usuário e das expectativas do provedor. A mistura de tecnologias de semântica e de contexto na infraestrutura SOA4ALL é um fator essencial de adaptação dinâmica dos serviços ao seu contexto de uso. Para facilitar a implantação de serviços semânticos em uma escala global, o conceito de contexto SOA4ALL abrange uma série de aspectos que vão desde informações do sistema e sua localização, até contextos sociais e normas legais [31].

SOA4ALL usa a tecnologia *Web* 2.0 [32] em sua interface de usuário, denominada SOA4ALL *Studio*, através do uso da tecnologia AJAX (*Asynchronous Javascript and XML*), que é elemento central desta interface. Esta tecnologia permite ao projeto fornecer aplicações com interfaces que reagem instantaneamente às solicitações dos usuários, em vez de recarregar as paginas em cada clique [31].

O componente central da arquitetura SOA4All que atua como uma infraestrutura central de comunicação e integração é chamado DSB (*Distributed Service Bus*). O DSB amplia as tradicionais tecnologias ESB (*Enterprise Service Bus*), tornando os ambientes computacionais abertos, altamente distribuídos e de grande escalabilidade, permitindo a implementação de registros de serviços distribuídos, a realização de integração de barramentos e a melhoria dos protocolos de coordenação e comunicação por meio da introdução dos espaços semânticos [33].

3.5 SLA@SOI

O SLA@SOI (*Service Level Agreement at Service Oriented Infrastructures*) [35] é um dos projetos afiliados ao FISO (*Future Internet Service Offer*) que é um dos *cluster* da iniciativa da União Européia para a construção da Internet do Futuro, o FIA (*Future Internet Assembly*). É um consórcio formado pela indústria, acadêmicos e institutos de pesquisa de toda Europa com o objetivo de pesquisar e criar tecnologias que possam incorporar infraestruturas cientes do SLA[5] para a economia de mercado.

O projeto de pesquisa SLA@SOI visa proporcionar um marco importante para a evolução na direção de uma economia orientada a serviços, onde serviços baseados em TI possam ser facilmente negociados como bens econômicos, ou seja, em condições bem definidas e confiáveis, e com custos claramente associados [34].

Este projeto tem como visão a "criação de uma infraestrutura de negócio orientada a serviços, fortalecendo de forma flexivel e confiável a economia de serviços" [34]. O foco do projeto é o gerenciamento de SLAs. Basicamente, a ideia do projeto é permitir que um prestador de serviço ofereça serviços com SLAs diferenciados, confiáveis e ajustáveis, e possa negociar estes SLAs de forma automatizada com os clientes. A arquitetura permite que serviços possam ser compostos de outros serviços, fornecidos por outras entidades ou até mesmo por outros serviços. Assim, é necessário um mapeamento gradual dos requisitos de SLA de níveis superiores para os níveis inferiores, bem como a agregação das capacidades de nível inferior aos níveis mais elevados. O processo de gerenciamento global de um SLA inclui as diferentes partes interessadas, ou seja, clientes, provedores de serviços e provedores de infraestrutura.

[5] O projeto SLA@SOI providencia uma infraestrutura para a prestação de serviços de acordo com o SLA negociado, isto é, a infraestrutura deve ser capaz de dividir seus recursos entre os diversos serviços com base no SLA, permitindo que alguns serviços recebam mais ou menos recursos.

4 Internet de Serviços: Requisitos e Desafios

4.1 Descrição de Serviços

No projeto CASCADAS, um protocolo chamado *Goal Needed* (GN) cobre a divulgação de descritores de serviços com riqueza semântica [19], ou seja, contextualizada, de serviços necessários em um dado momento à composição dinâmica. Um ACE originador envia uma mensagem *Goal Needed* (GN) a todos os ACEs habitantes do domínio semântico (ambiente composto por ACEs com funcionalidades específicas similares) que lhe interessa. Esta mensagem é uma espécie de pedido, com uma descrição semântica em anexo, que especifica que tipo de funcionalidades o ACE solicitante necessita de outros ACEs para atingir seus objetivos [23]. Os demais ACEs ao receberem o pedido, poderão responder utilizando uma mensagem *Goal Achievable* (GA), utilizada para informar semanticamente a outros ACEs seu endereço e qual tipo de tarefa um determinado ACE é capaz de fornecer. Após a identificação dos ACEs aptos a atender a solicitação, o ACE originador seleciona aquele que julga mais adequado.

O projeto BIONETS considera ontologias semanticamente relacionadas. Descritores de serviços baseados em XML são utilizados para representar a semântica dos serviços ofertados. Esses descritores podem ser mapeados para outras representações utilizando serviços de transformação [40]. Um descritor pode conter um conjunto de serviços básicos a serem invocados, chamados de células de serviço. Assim, o modelo de descrição de serviços do BIONETS é hierárquico e recursivo, permitindo a invocação de conjuntos de serviços básicos ou individuais. A proposta captura portanto as dependências entre os serviços descritos.

Já no projeto XIA, a divulgação se dá por meio da escolha de nomes legíveis representativos da funcionalidade de cada serviço. A documentação do projeto carece de detalhes sobre a descrição de serviços.

O SOA4All oferece uma API que é integrada ao DSB. Através desta API (*Application Programming Interface*), espaços semânticos distribuídos (ontologias) podem ser criados, além de permitir a publicação e assinatura de metadados relativos aos serviços. A descrição de serviços se dá através das linguagens WSDL (*Web Services Description Language*) e MicroWSMO (*Micro Web Service Modelling Ontology*). Existe ainda a linguagem WSMO-Lite que foi desenvolvida pelo projeto para descrever serviços usando anotações semânticas. Além disso, a proposta também suporta a descrição em linguagem natural (por exemplo, em formato .pdf). A descrição dos serviços existentes pode ser encontrada através de uma URL (*Uniform Resource Locator*).

O SLA@SOI utiliza um modulo de negociação para fazer o *upload* e *download* de *templates* de SLA. Portanto, não possui descritores de serviços.

Percebe-se que estas abordagens são bastante diversas. Enquanto o CASCADAS, o SOA4All e o SLA@SOI utilizam *software* especifico para descrever serviços, o XIA não contempla explicitamente a divulgação de serviços e SLAs. Entretanto, isto poderia ser feito através da publicação/assinatura de descritores de serviços e SLA na forma de objetos de

conteúdo. Ou seja, uma estrutura geral feita para distribuição de conteúdos poderia ser usada para este fim. A vantagem do uso de *software* geral ao invés de específico está na simplificação e redução de funcionalidades redundantes na arquitetura. Assim, um tópico de pesquisa em aberto é como descrever serviços em consonância com a identificação dos mesmos – seja do seu executável ou código fonte.

A Tabela 1 sumariza como cada uma das propostas estudadas acolhe o aspecto de descrição de serviços, bem como os demais aspectos que serão discutidos a seguir.

4.2 Identificação

Com relação a identificação dos serviços, que é o processo responsável pela atribuição de identificadores perenes e únicos aos serviços em um dado escopo, a maior parte das abordagens é incompleta. Somente a iniciativa XIA contempla este aspecto através dos SIDs (*Service Identifiers*). A identificação única é importante por diversos motivos, dentre eles rastreabilidade e melhoria da segurança. A ausência destes identificadores únicos publicados dificulta a rastreabilidade de serviços mal comportados. É um problema semelhante ao da rastreabilidade de endereços IP na Internet atual. A identificação única permite ainda atribuir sem repúdio a autoria de comportamentos ilícitos. Dentre os desafios de pesquisa existentes, estão a geração de identificadores autocertificáveis – que possam ser verificáveis a qualquer momento por entidades autorizadas, a geração de identificadores globais e a colisão de identificadores de serviços.

4.3 Descoberta

A descoberta de parceiros para a composição dinâmica de serviços é outra funcionalidade que apresenta diversidade de implementação. O CASCADAS utiliza novamente o protocolo GN/GA. Neste caso, é possível encontrar o serviço que melhor atende a um determinado objetivo, levando em conta o contexto exato desejado. A primeira etapa da descoberta é realizada quando um ACE originador envia uma mensagem *Goal Needed* (GN) a todos os ACEs habitantes do domínio semântico (ambiente composto por ACEs com funcionalidades específicas similares) que lhe interessa. Esta mensagem é uma espécie de pedido, com uma descrição semântica em anexo, que especifica que tipo de funcionalidades o ACE solicitante necessita de outros ACEs para atingir seus objetivos [23]. Os demais ACEs ao receberem o pedido, poderão responder utilizando uma mensagem *Goal Achievable* (GA), utilizada para informar semanticamente a outros ACEs seu endereço e qual tipo de tarefa um determinado ACE é capaz de fornecer. Após a identificação dos ACEs aptos a atender a solicitação, o ACE originador seleciona aquele que julga mais adequado.

No XIA, a descoberta se dá somente através dos nomes. Entretanto, como recém comentado, é possível ampliar esta arquitetura através da criação de novos suportes essenciais (*principals*) para esse fim ou simplesmente publicar descritores e oportunidades de composição via o suporte para conteúdos. No SOA4All, dois tipos de mecanismos de descoberta são oferecidos: descoberta baseada em texto completo e descoberta semântica. A descoberta baseada em texto completo busca *keywords* em arquivos de descrição WSDL e MicroWS-

MO. A descoberta semântica realiza uma busca estruturada e a verificação de adequação dos candidatos usando a linguagem WSMO-Lite, que considera objetivos, pré-condições, efeitos, e classificação de serviços. A descoberta pode ser disparada por classe de serviço ou funcionalidade. No SLA@SOI, a descoberta se dá através da consulta a um catálogo de registros acessível via o modulo de negociação.

Um aspecto comum a todas as abordagens é que elas utilizam *software* distribuído para esse fim. O ideal é que a descoberta de serviços possa usar os mesmos suportes arquiteturais que a descoberta de conteúdos. Isto elimina a sobreposição desnecessária de funcionalidades. A final, provavelmente após o estabelecimento dos SLAs, os serviços farão exatamente isto: trocar conteúdos e informações. Neste sentido, a proposta XIA está a frente das demais, através do suporte a distribuição de conteúdos. Embora, careça de suporte explícito (já implementado) para tanto. Assim, a descoberta conjunta de serviços e conteúdos ainda é um aspecto a ser pesquisado.

4.4 Composição Dinâmica

Quanto a composição dos serviços, o CASCADAS é muito engenhoso. Permite a agregação e diferenciação semântica dos serviços. Ou seja, uma determinada funcionalidade que esteja faltando a uma aplicação em um dado contexto (por exemplo, transcodificação de formato de mídia) pode ser agregada dinamicamente aos demais serviços da aplicação. Isto é feito de forma autonômica, sem interferência humana. Ainda, o CASCADAS permite que os serviços se especializem sob demanda em determinadas funcionalidades. Além disso, os agrupamentos podem ser fragmentados, e.g. quando uma aplicação precisa ser reutilizada para outro fim ligeiramente diferente ou quando funcionalidades não são mais necessárias. No BIONETS, a composição dinâmica autonômica também é suportada. Além disso, o BIONETS permite a seleção genética dos melhores candidatos a uma dada composição. No XIA, a composição dinâmica é possível através do relacionamento entre SIDs. Mas, não existe suporte autonômico para isto. No SOA4All, a composição de serviços é semi-automática e implementada na ferramenta SCS (*Service Construction Suíte*). No SLA@SOI, a agregação é dinâmica e automática, de forma semelhante ao SOA4All. Assim, as abordagens CASCADAS, BIONETS e SOA4All oferecem suporte para a composição dinâmica de serviços.

4.5 Localização Transparente

A transparência com relação a localização é importante para suportar a continuidade dos serviços no caso de eventos de mobilidade de substrato (terminal real ou virtual) e/ou dos próprios serviços (de uma máquina para outra). No CASCADAS este desacoplamento é feito utilizando um mecanismo de publicação e assinatura de eventos distribuídos. Assim, no caso de um evento de mobilidade, basta o serviço cadastrar-se novamente para receber eventos e a operação continuará normalmente. No XIA o desacoplamento é natural, ou seja, a arquitetura naturalmente desacopla identificadores de localizadores. Neste caso, existe um servidor de *lookup* que mapeia uma coisa na outra, dinamicamente. No SOA4All este problema é tratado como qualquer outro problema de contexto, i.e. o contexto é a localização do serviço. Ou

seja, o SOA4All reaproveita o suporte a contextos para tratar da localização. No SLA@SOI a localização é acoplada ao endereçamento na Internet atual. Esta é a abordagem mais limitada. O desejável é que cada serviço fosse unicamente identificado no escopo mais amplo possível e sua localização fosse armazenada separadamente em alguma estrutura de dados visível na rede, como por exemplo, tabelas *hash* distribuídas (DHT – *Distributed Hash Tables*). Neste quesito, a abordagem XIA se destaca novamente. Assim, o suporte ao desacoplamento identificador/localizador nos ambientes orientados a serviços ainda é um problema em aberto. Dentre as vantagens das arquiteturas que possuem esse desacoplamento, podemos citar a rastreabilidade a um serviço independentemente de sua localização e o não-repúdio em caso de comprovação de autoria. Outro desafio é o *design* e implementação de mecanismos de resolução de identificadores para localizadores que sejam genéricos o suficiente para atacarem esse problema tanto no contexto de serviços, quanto de conteúdo.

4.6 Negociação, Contratação e Gerenciamento

A negociação e a contratação no CASCADAS também usam o protocolo GN/GA. Um detalhe interessante é que ela é feita diretamente pelos serviços de forma distribuída, sem qualquer entidade centralizadora. Isto é fundamental para o suporte à auto-organização autonômica. A contratação usa um ambiente de comunicação orientada a conexão entre agentes chamado DIET (*Decentralised Information Ecosystem Technologies*) [20]. No XIA, não existem mecanismos implementados para a negociação e contratação de serviços. No SOA4All, a negociação e contratação utilizam o DSB, portanto de forma distribuída e semi-automática (dependente das necessidades dos usuários e dos negócios). Mas, a decisão final é de módulos de controle implementados no barramento. A negociação no SLA@SOI faz parte da gerência de SLAs, assim como a contratação. Quesitos de QoS são negociados. O SLA@SOI oferece também diversidade de metamodelos de SLAs para melhor atender as necessidades de contratação.

Aqui tem-se um duelo interessante. De um lado o CASCADAS com um modelo totalmente distribuído e do outro o SOA4All com a decisão centralizada. O tempo irá dizer qual das abordagens produz melhor resultados ou se ambas são adequadas. O XIA poderia evoluir para acomodar estas funcionalidades extremamente importantes à Internet de Serviços. Esta observação também vale para o gerenciamento de serviços. O XIA não oferece nenhum recurso de gerenciamento. Já o CASCADAS implementa uma ideia original chamada de supervisão pervasiva. Os próprios serviços (ACEs) contêm órgãos internos de supervisão que são integrados hierarquicamente com serviços especializados na supervisão de redes inteiras. Outro detalhe, os serviços especializados na supervisão são estruturalmente idênticos aos demais serviços, reaproveitando o mesmo código. Por isso, pode-se dizer que os serviços no CASCADAS são autossimilares. Isto é ótimo do ponto de vista de redução da sobreposição de funcionalidades, bem como do reuso de código e eficiência. A solução vai na direção da criação de *software* essencial. O BIONETS utiliza a autogestão de serviços. Os serviços são autônomos e deverão manter e ajustar seu funcionamento de acordo com as condições externas, exigências e carga de trabalho. A função de autogestão deve suportar todos os aspectos tradicionais de gerenciamento de sistema (tolerância a falhas, configura-

ção, contabilização, desempenho e segurança), bem como refletir a natureza altamente auto-organizada dos sistemas BIONETS. Tanto o SOA4All, quanto o SLA@SOI oferecem mecanismos tradicionais de monitoramento de serviços e SLAs. O SLA@SOI monitora a qualidade dos serviços prestados, enquanto o SOA4All inclui o monitoramento da infraestrutura de rede abaixo do estrato de serviços. Aqui, novamente o CASCADAS apresenta uma solução bastante engenhosa, com a criação de um sistema distribuído, auto-organizável, autoconfigurável e contextualizado de monitoramento. Tal abordagem permite uma redução da interferência humana, criando sistemas autogerenciáveis, adaptáveis às mudanças de contexto, situação e de suas próprias capacidades/estados. Porém, as pesquisas em soluções autonômicas ainda possuem alguns problemas em aberto, tais como: hierarquia de ciclos de decisão, estabilidade e técnicas para implementação.

4.7 Confiança

Quanto ao suporte da confiança na prestação de serviços, as abordagens CASCADAS, BIONETS, SOA4All e XIA oferecem este suporte. Já a abordagem SLA@SOI mantém somente o arquivamento de SLAs já realizados, a fim de verificar o grau de sucesso em contratos anteriores. A abordagem CASCADAS inclui também um sistema de gestão de reputação de serviços. O ideal é que as informações de contratos anteriores e monitoramento de qualidade sejam usadas para determinar a reputação dos serviços prestados, permitindo que os serviços estabeleçam redes de confiança no momento da composição dinâmica. A abordagem BIONETS implementa um sistema de gerenciamento de confiança e reputação. Para avaliar o desempenho de um serviço específico se faz necessário algum mecanismo de identificação que permita identificá-lo. É preciso identificar inequivocamente os serviços e monitorar a satisfação dos seus clientes, incluindo a justiça no uso de recursos, o consumo de energia, a confiabilidade, e, então avaliar se vale a pena manter, remover ou evoluir os serviços identificados. O estabelecimento de redes de confiança entre serviços é uma abordagem promissora. Entretanto, para que ela possa ser implementada com segurança, existe a necessidade da geração e armazenamento perene dos identificadores. Além disso, os identificadores devem ser desacoplados dos localizadores utilizados para encaminhamento de informações. A criação de redes de confiança nesses moldes ainda é um problema em aberto.

4.8 Mobilidade

No CASCADAS a mobilidade é nativa. Ou seja, o suporte a mobilidade de serviços é naturalmente suportada. Mas, pouco é comentado sobre o suporte a mobilidade de terminais. No BIONETS a mobilidade também é nativa. Entre os tipos de serviços, existem aqueles denominados serviços de migração, que como o próprio nome já diz, não são limitados a um nó específico, podendo sua funcionalidade migrar para outros nós da rede. No XIA o suporte a mobilidade de serviços também é nativa. No SOA4All, aparentemente, não existem mecanismos explícitos para este fim. O SLA@SOI suporta a migração de serviços de uma máquina virtual para outra. O suporte a mobilidade é fundamental em redes futuras, pois cada vez mais os usuários estão utilizado terminais móveis. Espera-se que uma arquitetura de serviços neste contexto suporte a mobilidade de serviços, terminais reais ou virtuais, pessoas e até

mesmo redes, tudo isso sem reduzir a disponibilidade dos serviços prestados. Trata-se de um desafio considerável. Soluções como a proposta pelo XIA, que fazem o desacoplamento de identificadores e localizadores, tem se destacado no suporte à mobilidade. A mobilidade sem perda de identificação ainda é um desafio a ser explorado.

4.9 Adaptabilidade e Autonomicidade

Nos quesitos adaptabilidade e autonomicidade, o CASCADAS oferece suporte as chamadas propriedades **-awareness*, que significam suporte a ciência de diversos aspectos (o asterisco identifica cada aspecto). O CASCADAS oferece *context-*, *situation-*, e *self-awareness*. A ciência da situação caracteriza o ambiente externo a um serviço, enquanto a ciência do *self* (eu) caracteriza o autoconhecimento do serviço sobre suas próprias competências, estado, etc. No CASCADAS os serviços podem alterar seus planos de ações em função de mudanças no estado interno, do ambiente ou de contexto. Ou seja, o serviço evolui para melhor se adaptar ao ambiente, as necessidades do cliente e as suas capacidades internas. Além disso, o CASCADAS oferece amplo conjunto de propriedades auto-*, tais como auto-organização, autoconfiguração, autoproteção, auto-otimização, autocontextualização, autogerenciamento, etc. As propriedades auto-* foram propostas pela IBM para reduzir a interferência humana em sistemas computacionais [26].

No BIONETS existem nós que fazem o sensoriamento da infraestrutura de suporte aos serviços, permitindo que os serviços evoluam em função do estado desses (*infrastructure-awareness*). Também, é feita uma seleção dos serviços que mais se adaptam ao estado do ambiente e as necessidades dos clientes (*user-awareness*). O BIONETS possui ainda uma funcionalidade bastante inovadora: a chamada autocatálise ou geração espontânea de serviços.

Na proposta XIA não existem mecanismos implementados para a adaptabilidade ou autonomicidade dos serviços. Ou seja, não existe mecanismo explícito que permita que os serviços se adaptem à mudanças no ambiente ou a outros eventos. Existe, entretanto, a possibilidade de ciência dos nós de rede (*host-awareness*), de ciência aos conteúdos trocados (*content-awareness*) e de ciência aos domínios pertencentes (*domain-awareness*). Mas, para tirar proveito dessas ciências, novas funcionalidades terão que ser incorporadas ao XIA.

No SOA4All, a adaptabilidade as necessidades dos usuários e negócios é automática, mas dependente de interferência dos usuários, negócios e contexto. O mesmo vale para o quesito autonomicidade. Na verdade, o SOA4All pode ser classificado como imperativo. Já o SLA@SOI fornece ciência da infraestrutura, negócio e SLA, mas em termos de suporte a autonomicidade se parece com o SOA4All.

Comparativamente, as soluções autonômicas têm se destacado quando o assunto é redução de gastos com operação, adaptabilidade a novos contextos e aos recursos da infraestrutura, e crescimento exponencial no número de serviços. Imagine o desafio que seria gerenciar uma rede com milhões de serviços diferentes e que requer frequente intervenção humana. Observa-se, portanto, que uma solução coesa para prover **-awareness* na Internet de Serviços ainda precisa ser desenvolvida.

4.10 Evolução

Por fim, tem-se o requisito da evolução. O CASCADAS e o BIONETS oferecem evolução autonômica. Ou seja, a arquitetura é capaz de evoluir com o mínimo de interferência humana. Observe que isto não significa que os humanos não façam parte desta evolução. Eles fazem, definindo os objetivos, regras, condições de contorno, negócios, etc. Quer dizer que os humanos não precisam gastar tempo excessivo de operação para fazer com que os serviços evoluam. Na atual versão do XIA, a evolução dos serviços ainda é totalmente dependente das ações humana na arquitetura. Já o SOA4All e o SLA@SOI fornecem uma solução intermediária.

Tabela 1. Comparação de iniciativas em Internet de Serviços.

Princípios	**CASCADAS**	**BIONETS**	**XIA**	**SOA4All**	**SLA@SOI**
Descrição	GN abrange descrição semântica	Semântica	Nomes legíveis	Semântica	Via *templates* registrados
Identificação	Carece de identificação de serviços	Carece de identificação de serviços	Prove a identificação única de serviços	Carece de identificação de serviços	Carece de identificação de serviços
Descoberta	Seleção de serviços usando GN/GA		Baseada na resolução de nomes para identificadores	Baseada na *web* semântica	Feita através de consultas a um catálogo de serviços
Composição	Agregação/diferenciação dinâmica, contextualizada	Dinâmica, genótipo de serviços	Dinâmica baseada em identificadores	Provisionamento *business-aware* e amigável para os usuários	Dinâmica com agregação e invocação semântica
Localização transparente	Desacoplamento via publicação/assinatura de eventos		Desacoplamento via servidor de *lookup*	Tratada como um problema de contexto	Acoplada, via gerenciador de serviços
Negociação	Direta usando GN/GA		Não possui[6]	Baseada na *web* semântica	Faz parte da gerência de SLA
Contratação	Direta usando agentes DIET		Não possui[5]	Automática de acordo com a necessidade dos usuários e negócios	Via gerente de SLA, com QoS e metamodelos
Gerenciamento	Supervisão pervasiva dos serviços	Autogestão	Não possui[5]	Monitoramento de serviços, SLAs e infraestrutura	De SLAs e serviços, qualidade
Confiança	Gestão de confiança e reputação	Gestão de confiança e Reputação.	Gestão de confiança	Gestão de confiança	Arquivamento de SLAs
Mobilidade	Nativa	Nativa	Nativa	Não possui[5]	Migração de serviços de uma máquina para outra.
Adaptabilidade	*-*awareness*, rede de conhecimento, mudança de planos de ação nos serviços.	Sensoriamento do ambiente, *fitness* aos desejos dos usuários	Manual. *Context-awareness*.	Automática, mas dependente de interferência dos usuários, negócios, contexto	*Infrastructure-, business-,* SLA-*awareness*
Autonomicidade	Amplo suporte de	Autogeração de	Manual	Imperativa[7],	Imperativa, depen-

[6] Não possuí mecanismos explícitos.

	funcionalidades auto-*, tais como auto-organização, auto-otimização, autoproteção	serviços (autocatálise)		dependente de interferência dos gerentes, usuários, etc. Automação do ciclo de vida	dente de interferência dos gerentes. Estado dos serviços, recursos.
Evolução	Autonômica[8], via agregação/diferenciação, adequação de planos de ações a mudanças de contexto e objetivos	Autonômica, inspirada em algoritmos genéticos. Guiada pelos usuários e recursos	Manual	Imperativa, guiada pelos usuários e recursos	Imperativa, através da substituição dos serviços ofertados

5 Considerações Finais

Embora já tenha ocorrido um grande avanço na direção de trazer a economia de serviços para a Internet, muitos princípios da abordagem SOA ainda continuam apenas parcialmente atendidos pelas propostas existentes. Nenhuma das iniciativas discutidas na seção anterior atende sozinha a todos os atributos do SOA, tampouco aos desafios ainda presentes na Internet de Serviços e Internet do Futuro.

As iniciativas oriundas da *web*, do SOA e dos *web services*, tais como SOA4ALL e SLA@SOI aproximam a plataforma de serviços dos requisitos de negócios e dos usuários. Mas, carecem de suporte ao crescimento exponencial do número de serviços esperado na Internet do Futuro (autonomicidade, adaptabilidade, evolução). Também, carecem de suporte a identificação única de serviços e ao desacoplamento identificadores/localizadores. As iniciativas bioinspiradas, a.k.a. CASCADAS e BIONETS surpreendem na questão da auto-organização semântica, gerenciamento de serviços, autonomicidade, adaptabilidade e evolução. Mas, carecem de maior proximidade as necessidades dos negócios e dos usuários, além de não suportarem o desacoplamento de identificadores/localizadores. Por fim, a abordagem XIA possui suporte a este desacoplamento e inova na questão das entidades expressarem as suas necessidades a rede. Mas, carece dos demais recursos existentes nas soluções bioinspiradas e evolucionárias advindas do SOA.

Assim, a principal conclusão deste artigo é que somente abordagens integradoras, mais amplas, mas restritas a essência do necessário (simplicidade), poderão avançar no sentido a satisfazer o real potencial da Internet de Serviços. Novas pesquisas devem integrar melhor seus princípios e decisões de projeto aos demais esforços de pesquisa em Internet do Futuro. Pouco adianta resolver um problema específico da Internet de Serviços, se a solução é insatisfatória com relação ao estado da arte de outros aspectos da Internet do Futuro. Considere por exemplo o caso da ausência de identificadores perenes e únicos para os serviços nas abordagens CASCADAS, SOA4All, SLA@SOI e BIONETS. Embora esses projetos ataquem vários problemas relevantes da Internet de Serviços, sua aplicabilidade em uma Internet do Futuro que exige identificadores perenes se torna questionável. O pior é que a falta de integração cria a sobreposição de funcionalidades, reduzindo a eficiência e evitando soluções mais simples e gerais.

[7] Segue o padrão tradicional de desenvolvimento de software de prateleira.
[8] Segue a ideia da computação autonômica.

A Internet de Serviços é ingrediente fundamental da Internet do Futuro. A visão de que todo *software* pode ser orquestrado como um serviço é desafiadora e ao mesmo tempo revolucionária, pois torna o uso do *software* mais flexível, dinâmico, independente da interferência humana, reduzindo custos e aumentando a eficiência. É, portanto, preciso explorar essa ideia de forma mais profunda, melhor integrando-a aos demais aspectos da Internet do Futuro. Existe então uma ótima oportunidade de pesquisa e desenvolvimento no atual *status-quo*, principalmente quando se considera os ganhos de sinergia que poderiam alimentar os demais ingredientes da Internet do Futuro, bem como a importância do terceiro setor na economia dos países desenvolvidos e em desenvolvimento (mais de 50% do produto interno bruto).

6 Referências

[1] INTERNET WORLD STATS. **World Internet Usage and Population Statistics**. Disponível em http://www.internetworldstats.com/stats.htm. Último acesso em: 14 de Março de 2013.

[2] REDING, Viviane. **The Future of the Internet: A Compendium of European Projects on ICT Research Supported by the 7th Framework Programme for RTD.** European Commission White Paper, 2008.

[3] STANFORD UNIVERSITY. **Clean Slate: An Interdisciplinary Research Program.** Disponível em: http://cleanslate.stanford.edu/about_cleanslate.php. Último Acesso: 23/01/2012.

[4] CARDOSO Jorge, WINKLER Matthias, VOIGT Konrad, BERTHOLD, Henrike. **IoS-based services, Platform Services, SLA and Models for the Internet of Services.** University of Coimbra, Portugal. SAP Research CEC, Chemnitzer Strasse 48, 01187 Dresden, Germany. 2011.

[5] PAPAZOGLOU Michael P., TRAVERSO Paolo, DUSTDAR Schahram, LEYMANN Frank. **Service-Oriented Computing Research Roadmap.** Dagstuhl Seminar Proceedings 05462. 2006.

[6] DUBRAY, Jean-Jacques. **Fundamentals of Service Orientation.** White Paper. 2005.

[7] KOTLER, Philip. **Administração de Marketing: Análise, Planejamento, Implementação e Controle.** 5ª ed. São Paulo, Brasil. Editora Atlas. 1998.

[8] CARTER, Sandy. **The New Language of Business: SOA & Web2.0.** IBM Press. 2007.

[9] ENDREI, M., et al. **Patterns: Service-Oriented Architecture and Web Services.** IBM Press. First Edition. April 2004.

[10] AIELLO Marco, DUSTDAR Schahram. **Service-Oriented Computing: Service Foundations.** Dagstuhl Seminar Proceedings 05462. 2006.

[11] ROSEN Michael, LUBLINSKY Boris, SMITH T. Kevin, BALCER J. Marc. **Applied SOA: Service-Oriented Architecture and Design Strategies.** Wiley Publishing, Inc. 2008.

[12] SOA PRINCIPLES. **An Introduction to the Service-Orientation Paradigm.** Disponível em http://en.wikipedia.org/wiki/Autonomic_computing. Último acesso em: 16 de Janeiro de 2012.

[13] CHANNABASAVAIAH Kishore, HOLLEY Kerrie, TUGGLE Edward M. **Migrating to a Service-Oriented Architecture.** Disponível em http://www.ibm.com/developerworks/library/ws-migratesoa2/. Ultimo acesso em: 14/03/2013.

[14] GÜNER Shelda. **Architectural Approaches, Concepts and Methodologies of Service Oriented Architecture.** Software System Institute Technical University Hamburg, Germany, August 2005.

[15] LALIWALA, Zakir. **Event-driven Service-oriented Architecture for Dynamic Composition of Web Services.** Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar. 2007.

[16] PAPAZOGLOU Mike P., HEUVEL Willen-Jan van den. **Service Oriented Architecture: approaches, technologies and research issues.** The VLDB Journal — The International Journal on Very Large Data Bases. Vol. 16. 2007.

[17] ELFATATRY Ahmed, LAYZELL Paul. **Negotiating in Service-Oriented Environments.** Communications of the ACM, v. 47, n. 8. August 2004.

[18] MANZALINI Antonio, MANNELLA Antonietta, MOISO Corrado, HASELOFF Sandra, KUSBER Rico, BRGULJA Nermin, ZAMBONELLI Franco, DEUSSEN Peter H., SAFFRE Fabrice, BARESI Luciano, LENT Ricardo, GELENBE Erol, FERDINANDO Antonio D., CASCELLA Roberto. **CASCADAS - Bringing Autonomic Services to Life**. Deliverable 8.4. January 2009.

[19] HOFIG H., et al., **On Concepts for Autonomics Communication Elements.** In Proc. Of the 1st IEEE International Workshop on Modeling Autonomic Communication Environments (MACE 2006), Dublin, October 2006.

[20] MARROW P., et al. **Agents in Decentralized Information Ecosystems: the DIET Approach.** In Proc. of the Artificial Intelligence and Simulation Behaviour Convention (AISB 2001). York, March 2001.

[21] HASELOFF S., et al. **CASCADAS - First Prototype Integration (release 1).** Deliverable 1.3. January 2008.

[22] BAUMGARTEN, Matthias, ZAMBONELLI Franco, CASTELLI Gabriella, BICOCCHI Nicola, KUSBER Rico, BRGULJA Nermin. **Final Report on Open Toolkit for Knowledge Networks and on Advanced Knowledge Networks Concepts and Models.** Deliverable 5.4. December 2008.

[23] MANZALINI Antonio, MANNELLA Antonietta, ALFANO Rosario, HÖFIG Edzard, MAMEI Marco, BENKŐ Borbála K., KATONA Tamas, KUSBER Rico, BRGULJA Nemin. **Bringing Autonomic Services to Life: Report on state-of-art, requirements and ACE model.** Deliverable 1.1. January 2007.

[24] BIONETS. **Overview: BIOlogically-Inspired Networks and Service**. Disponível em http://www.bionets.eu/index.php?area=11. Último acesso em: 21 de Julho de 2011.

[25] LINNER David, PELLEGRINI Francesco D., MIORANDI, Daniele, MOISO Corrado, BACSARDI Laszlo. **BIONETS Architecture: from Networks to SerWorks.** IEEE. 2007.

[26] KEPHART, J., CHESS, D., **The Vision of Autonomic Computing**, IEEE Computer 36 (1) (2003).

[27] PELLEGRINI F., CARRERAS I., ALFANO G., TAHKOKORPI M., SZABO S., BORGIA E., LATVAKOSKI J., SCHRECKLING D., PANAGAKIS A., VAIOS A. **BIONETS Requirements and Architectural Principles: Architecture, Scenarios and Requirements Refinements.** Deliverable 1.1.1. August 2006.

[28] TSCHUDIN C., YAMAMOTO Lidia. **Self-evolving Network Software.** Praxis der Informationsverarbeitung und Kommunikation (PIK Magazine), pp. 206-210. K.G. Saur Verlag, Munich, Germany. December 2005.

[29] MIORANDI, Daniele, YAMAMOTO, Lidia, DINI, Paolo. **Service Evolution in Bio-inspired Communication Systems**. European Commission. Project: BIONETS. 2006.

[30] ANAND A., et al. **XIA: An Architecture for an Evolvable and Trustworthy Internet**. Technical Report. January 2011.

[31] SOA4ALL. **Welcome to SOA4ALL.** Disponível em: http://www.SOA4All.eu. Ultimo acesso em: 03 de Novembro de 2011.

[32] O'REILLY, Tim. **Web 2.0: Principles and Best Practices.** O'REILLY radar. 2004.

[33] KRUMMENACHER, Reto, NORTON, Barry, SIMPERL, Elena, PEDRINACI, Carlos. **SOA4ALL: Enabling Web-scale Service Economies**. IEEE International Conference on Semantic Computing. 2009.

[34] WOLFGANG Theilmann, GUINEA Sam, BROSCH Franz, YAHYPOUR Ramin. **Deliverable D.A1a Framework Architecture-Evaluated Architecture**. 2011.

[35] SLA@SOI. **Project Web Site.** Disponível em: http://sla-at-soi.eu. Último acesso em:20 de Setembro de 2011.

[36] SOA4All. **KPI-Based Process Modeling. A Public Sector Case Study.** White Paper. 2010.

[37] CARDOSO, Jorge. WINKLER, Matthias. VOIGT, Konrad. BERTHOLD, Henrike. **IoS-based services, Platform Services, SLA and Models for the Internet of Services**. ICSOFT09. 2009.

[38] ANDRIKOPOULOS, V., **State of the art report on software engineering design knowledge and survey of HCI and contextual knowledge**. Deliverable PO-JRA-1.1.1. (2008).

[39] HOURCADE, J., SARACCO, R., WAHLSTER, I., POSCH, R. **Future Internet 2020: Visions of an Industry Expert Group. DG Information Society and Media – Directorate for Converged Networks and Service Manifesto**. White Paper. (2009).

[40] BIONETS. **Autonomic Service Life-Cycle and Service Ecosystems**. Deliverable 3.2.1. (2007).

[41] PAN, J., PAUL, S., JAIN, R. **A Survey of the Research on Future Internet Architectures**. Communications Magazine, IEEE 49 (7) (2011).

[42] ISMAIL, A., YAN, J., SHEN, J. **An Offer Generation Approach to SLA Negotiation Support in Service Oriented Computing.** Journal Service Oriented Computing and Applications Archive. Volume 4 Issue 4. (2010)