

Arquitetura e Requisitos de Rede para Web Labs

Paulo R.S.L. Coelho¹, Rodrigo F. Sassi^{2,1}, Eliane G. Guimarães²
Eleri Cardozo¹, Luis F. Faina³, Alex Z. Lima¹

¹Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas
13083-970 – Campinas – SP

²Centro de Pesquisas Renato Archer
13083-970 – Campinas – SP

³Faculdade de Computação
Universidade Federal de Uberlândia
38400-902 – Uberlândia – MG

pcoelho@dca.fee.unicamp.br

Abstract. *This paper presents an architecture for building remote access laboratories (or Web Labs) following the service-oriented computing approach. In this architecture the application's building blocks are services that can be recursively composed resulting in more comprehensive services. Remote access laboratories can benefit of this approach. Every lab resource (physical or logical) is modeled and implemented as a service (in our case, a Web Service) and lab experiments are assembled by composing these services. A Web Lab built following the proposed architecture is presented with examples of remote experiments in the field of mobile robotics. The performance of such experiments is evaluated in terms of the speed of the accessing network for local area, campus, ADSL, and virtual private networks (VPN).*

Resumo. *Este artigo apresenta uma arquitetura para a construção de laboratórios de acesso remoto (ou Web Labs) segundo o paradigma de computação orientada a serviço. Nesta arquitetura, os blocos elementares são serviços que, recursivamente, podem ser combinados na construção de serviços mais complexos. Cada recurso do laboratório (físico ou lógico) é modelado e implementado como um serviço (no nosso caso, um serviço Web). Experimentos oferecidos pelo Web Lab são construídos por meio da composição destes serviços. Um Web Lab construído segundo a arquitetura proposta é apresentado, juntamente com exemplos de experimentos remotos na área de robótica móvel. O desempenho dos experimentos são avaliados em função da velocidade da rede de acesso para redes locais, de campus, ADSL e redes privadas virtuais (VPN).*

1. Introdução

Laboratórios de acesso remoto ou *Web Labs* têm sido propostos como poderosas ferramentas de suporte ao ensino a distância. O maior desafio na implementação de *Web Labs* é prover uma infra-estrutura onde experimentos possam ser incorporados e alterados com facilidade.

Este artigo explora o paradigma de computação orientada a serviço para a implementação de Web Labs. Considerada como a evolução da computação distribuída, computação orientada a serviço é definida como "o paradigma de computação que utiliza serviços como elementos fundamentais no desenvolvimento de aplicações" [Papazoglou and Georgakopoulos 2003]. Por meio de mecanismos de composição de serviços (ou orquestração) pode-se construir um serviço mais completo que, por sua vez, pode fazer parte de um nível de composição ainda maior [Curbera et al. 2003].

A vantagem mais significativa de computação orientada a serviço para a implementação de Web Labs é o oferecimento de experimentos por meio de composição de serviços. Outro benefício é o uso de recursos laboratoriais presentes em diferentes domínios administrativos por meio de federação de serviços. Este artigo propõe uma arquitetura que tem como base composição e federação de serviços para suporte a Web Labs. Uma implementação completa desta arquitetura é apresentada, bem como um Web Lab para o ensino de robótica móvel construído sobre esta arquitetura. Uma avaliação do Web Lab para diferentes redes de acesso é conduzida, demonstrando que os requisitos de rede impostos por Web Labs de alta interatividade e operação em tempo real apontam para o uso de redes de alto desempenho.

O artigo está organizado da seguinte forma. A Seção 2. apresenta alguns trabalhos correlatos. A Seção 3. apresenta a arquitetura para Web Labs proposta. A Seção 4. descreve um Web Lab no domínio da robótica móvel implementado segundo esta arquitetura. A seção 5. avalia o comportamento do Web Lab em função da velocidade da rede de acesso. Finalmente, a Seção 6. encerra o artigo com conclusões e trabalhos futuros.

2. Trabalhos Correlatos

Web Labs foram desenvolvidos inicialmente com tecnologias Web bem conhecidas, como *applets* Java no lado do usuário e extensões CGI (*Common Gateway Interface*) no lado servidor. Posteriormente, projetos utilizando abordagens baseadas em objetos distribuídos foram propostas. Algumas destas abordagens utilizam no lado do usuário somente navegadores Web com capacidade de executar *applets* Java [Hua and Ganz 2003] ou simplesmente páginas HTML geradas dinamicamente [Khamis et al. 2003]. Outras abordagens requerem software dependente de plataforma no terminal do usuário [Denis et al. 2003]. Estas abordagens possuem em comum o limitado reuso de software e restrita capacidade de customização, devido a baixa granularidade dos objetos distribuídos e a utilização de protocolos de interação comumente bloqueados por *firewalls*.

Recentemente, propostas de arquiteturas de Web Labs baseadas em componentes de software [Guimarães et al. 2003a] [Guimarães et al. 2003b] permitem aumentar o reuso e capacidade de customização, graças à capacidade de interconexão e configuração inerentes aos componentes de software. Apesar de apresentarem claramente um avanço em relação às propostas anteriores, modelos de componentes de software convencionais também utilizam protocolos de interação que não atravessam *firewalls*. Propostas que empregam somente serviços Web [Casini et al. 2004, Coelho et al. 2007] ou serviços Web em conjunto com componentes de software [Sassi et al. 2006] têm surgido com o

intuito de eliminar esta desvantagem.

Hassane e outros [Yan et al. 2006] descrevem uma arquitetura para Web Labs baseada na integração de serviços Web e computação em grade. Esta arquitetura possui similaridades com a descrita neste artigo no sentido que ambas representam recursos como serviços Web e defendem o uso federado de Web Labs por meio da composição de serviços disponíveis em vários domínios. Entretanto, os autores não fornecem qualquer dado experimental ou estudo de caso que demonstre o uso de composição de serviços ou federação de Web Labs.

O Projeto iLAB [MIT 2006] conduzido pelo MIT é pioneiro em utilizar serviços Web no domínio de Web Labs. iLAB desacopla as funcionalidades específicas dos Web Labs relativas a execução dos experimentos, das funcionalidades administrativas gerais, tais como autenticação, registro e autorização de usuários, gerência de grupos e armazenamento de resultados dos experimentos. Para as tarefas administrativas, iLAB usa um *Service Broker* que concentra e processa as requisições de gerenciamento e uso de Web Labs.

A proposta aqui descrita apresenta algumas similaridades com a do projeto iLAB, a destacar: desacoplamento entre o uso e a gerência de Web Labs; utilização de serviços Web para gerência e acesso a Web Labs; e uso federado de Web Labs. Entretanto, há uma diferença fundamental entre as propostas: iLAB tem como principal alvo a interligação de Web Labs (via um *Service Broker*) que concentra o acesso aos Web Labs federados. A proposta apresentada neste artigo defende a composição de serviços como forma flexível de disponibilizar experimentos em um Web Lab. No uso federado de recursos, a composição substitui uma invocação de serviço para um elemento centralizador (*Service Broker*) por um fluxo de invocações de serviços mantidos pelos domínios que compõem a federação. Isto evita a centralização de informações sobre usuários e recursos, fato que compromete a privacidade e segurança. A autenticação de usuários é feita em seus respectivos domínios (que mantêm suas próprias bases de usuários) e a verificação de reserva, disponibilidade e autorização de uso de recursos é processada nos respectivos domínios que mantêm os recursos envolvidos na composição. Esta independência de manutenção de usuários e recursos, bem como o estabelecimento de políticas de autenticação e autorização é uma característica fundamental de federações, características estas que jamais podem ser atingidas com o emprego de um elemento centralizador.

3. Uma Arquitetura para Web Labs

Este artigo propõe uma arquitetura orientada a serviço para Web Labs utilizando serviços Web classificados em três categorias: serviços de acesso, de interação e de comunicação. A Figura 1 ilustra a arquitetura em um diagrama de pacotes UML. A arquitetura proposta permite a criação de experimentos a partir da composição dos serviços oferecidos, não interferindo nos experimentos já disponíveis. A composição pode ainda utilizar serviços oferecidos por outros Web Labs formando assim uma federação de Web Labs.

3.1. Serviços de Acesso

Para realizar um experimento em um Web Lab, o usuário utiliza-se dos serviços de acesso para autenticar-se e, só então, seleciona um experimento a partir de uma lista de possíveis experimentos. Para restringir o uso do Web Lab, o serviço de gerência de participantes

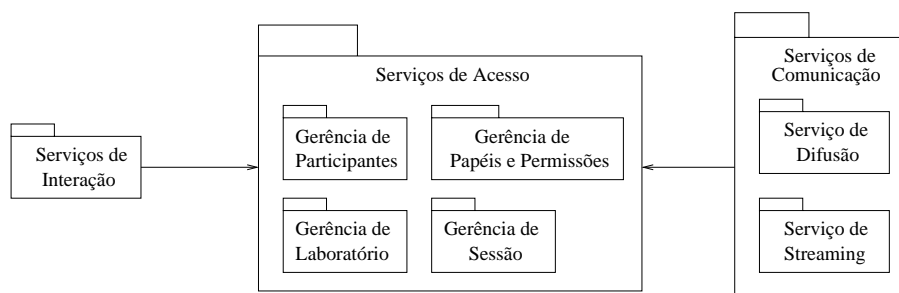


Figura 1. Arquitetura orientada a serviço para Web Labs.

possibilita o cadastramento, atualização e remoção de participantes (usuário individual ou grupo). Cada participante utiliza o Web Lab segundo papéis e permissões a ele atribuídos. Papéis definem o escopo de funções dentro do Web Lab (administrador, instrutor, aluno, etc.). Permissões definem as ações que um determinado papel pode executar no Web Lab. O serviço de gerência de papéis e permissões permite definir, atribuir e verificar papéis e permissões.

O serviço de gerência de laboratórios possibilita o cadastramento, atualização e remoção de Web Labs, experimentos e recursos. Experimentos são unidades de acesso (i.e., usuários acessam experimentos de um dado Web Lab). Recursos são unidades de reserva e de uso necessários para a realização de experimentos.

Serviços interativos como Web Labs necessitam de acesso exclusivo para certos recursos (por exemplo, robôs e câmeras) e, assim, demandam facilidades que possibilitem a reserva dos mesmos por um determinado período de tempo. Ao usuário cabe somente a reserva do experimento, enquanto que aos serviços de acesso cabe garantir que os recursos associados ao experimento estarão disponíveis para a reserva em questão.

O serviço de gerência de sessão é responsável por autenticar os usuários, verificar se o usuário possui reserva válida e iniciar uma sessão de acesso pelo período estabelecido na reserva. A sessão de acesso é finalizada pelo usuário ou pelo sistema, caso o período da reserva tenha se expirado. O serviço de gerência de sessão propaga no serviço de difusão (Seção 3.3.) eventos relativos ao início e término de sessões. Estes eventos são utilizados para fins de *logging*, monitoramento de uso, bem como para configuração dinâmica de experimentos.

3.2. Serviços de Interação

Na arquitetura proposta, os serviços de interação devem ser implementados como serviços Web específicos para o domínio do Web Lab. Por exemplo, em um Web Lab no domínio da robótica móvel, os principais serviços de interação suportam o controle e a operação de robôs móveis. A utilização destes serviços depende do estabelecimento de uma sessão de acesso para autenticar o usuário e verificar a existência de reserva para realização do experimento.

3.3. Serviços de Comunicação

Para suporte à comunicação foram especificados dois serviços: serviço de difusão e serviço de *streaming*. O serviço de difusão é responsável por prover comunicação ponto-multiponto entre usuários e Web Lab. O serviço é centrado na difusão de

informação com capacidade de persistência e filtragem. Esta informação é expressa por meio de documentos XML (*Extensible Markup Language*) de conteúdo arbitrário. O serviço é baseado no modelo produtor-consumidor, sendo responsável pela entrega aos consumidores dos documentos submetidos pelos produtores de informação.

O serviço de difusão (Figura 2) expõe uma interface WSDL que permite o registro de produtores e consumidores, a submissão de documentos para difusão e a consulta de documentos armazenados. Aos documentos submetidos, o Gerente Produtor acrescenta uma marca de tempo e os repassa ao Canal de Difusão. No canal, é averiguado se o documento é persistente ou transiente, de acordo com o tempo de vida estipulado pelo produtor. Em caso de persistência, o documento é armazenado na base de dados. Por último, o Gerente Consumidor procede a entrega do documento aos consumidores registrados.

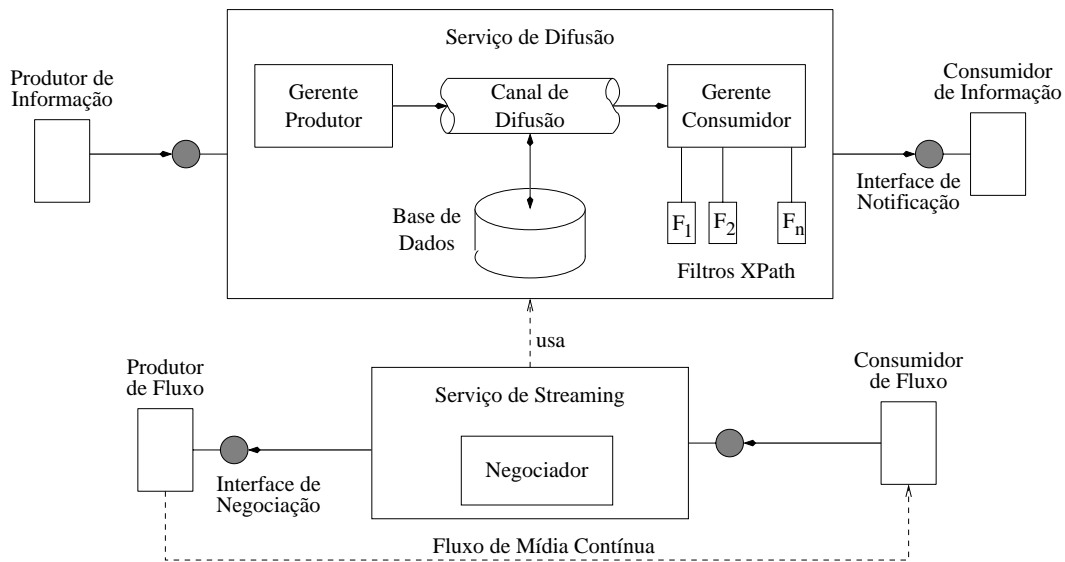


Figura 2. Componentes do serviço de difusão e streaming.

Consumidores registram-se no serviço fornecendo um filtro e o endereço da interface de submissão de documentos. Filtros são expressões XPath que, aplicadas ao documento submetido, extraem apenas a parte da informação que interessa ao consumidor.

O serviço de difusão pode ser empregado para notificação de eventos (por exemplo, alarmes), difusão de mensagens textuais (por exemplo, em aplicativos tipo bate-papo), difusão de informação gráfica (por exemplo, em aplicativos de quadro branco) e difusão periódica de informação (por exemplo, dados de monitoramento).

O serviço de *streaming* permite a gerência de fluxo de mídia contínua entre produtores e consumidores de fluxo, sendo implementado sobre o serviço de difusão. O serviço de *streaming* suporta gerência de conexões multimídia ponto-multiponto com negociação de parâmetros de mídia tais como codificação e tamanho do quadro de vídeo, codificação e tamanho da amostra de áudio, taxa de amostragem, etc. Ao se registrarem no serviço, produtores de mídia suprem uma interface de negociação e uma faixa para seus parâmetros de mídia contínua. Durante o registro de consumidores, estes fornecem ao serviço os valores de sua preferência para os parâmetros de mídia e o produtor do

qual desejam receber fluxo. O serviço, via interface de negociação, supre o produtor com os parâmetros do consumidor que retorna ao serviço um conjunto de valores para os parâmetros (eventualmente ajustando-se às preferências do consumidor) e o endereço de rede para onde o fluxo está sendo transmitido. O serviço repassa esta informação para o consumidor que procede os ajustes necessários para a recepção do fluxo. Caso os parâmetros do produtor e consumidor sejam incompatíveis, o serviço retorna ao consumidor um indicativo de erro e a faixa aceitável pelo produtor.

4. GigaBOT Web Lab

GigaBOT é um Web Lab para o ensino de robótica móvel implementado segundo a arquitetura proposta. Por serem independentes do domínio de aplicação, os serviços de acesso e comunicação não necessitam de qualquer adaptação para o domínio de robótica móvel. Os serviços de interação, ao contrário, foram desenvolvidos para este domínio específico e são descritos na seqüência.

O GigaBOT Web Lab dispõe de dois robôs Pioneer P3-DX da ActivMedia com 16 sonares e *bumpers* (parachoques de proteção). Um dos robôs possui processador de bordo, interface de rede 802.11g sem fio e câmera de bordo Canon VC-C4 com PTZ (*pan/tilt/zoom*) conectada à placa de captura de vídeo. O segundo robô não possui processador de bordo, sendo controlado por um computador de mão IPaq H5555 usando o sistema operacional Familiar Linux v0.8.1 com suporte a rede sem fio 802.11b. Este robô possui uma câmera fixa Axis 206W equipada com interface de rede sem fio 802.11b.

A infra-estrutura de suporte aos serviços consiste de um contêiner de serviços Web Axis Java instalado no servidor de aplicação Apache Tomcat. Estes servidores, disponibilizados pelo Projeto Apache [Apache Software Foundation 2006], executam em uma máquina com dois processadores Xeon e sistema operacional Linux. Uma câmera panorâmica AXIS 213 PTZ é utilizada para captura de imagens quadro a quadro, sendo tipicamente utilizada para experimentos de navegação baseada em visão e registro (gravação) do experimento. Uma segunda câmera AXIS 213 PTZ com módulo de áudio é utilizada para acompanhamento do experimento. Esta câmera gera um fluxo de vídeo e áudio no formato MPEG/RTP, e pode ser compartilhada (via serviço de *streaming*) por um grupo de usuários que realizam o experimento em equipe.

A Figura 3 ilustra os componentes da infra-estrutura e os protocolos de interação entre componentes nos lados do usuário e do Web Lab. Os componentes no lado do usuário consistem de navegadores Web que suportam o protocolo HTTP (*Hypertext Transfer Protocol*), aplicativos Java (processos Java que executam no *desktop* do usuário) que suportam o protocolo SOAP (*Simple Object Access Protocol*) sobre HTTP e apresentadores de mídia que suportam os protocolos MPEG-4 sobre RTP (*Real Time Protocol*).

4.1. Serviços de Interação Implementados

O GigaBOT Web Lab disponibiliza cinco serviços de interação: serviço de controle de locomoção, serviço de telemetria, serviço de ações, serviço de visão e serviço de submissão de código.

Para controle de locomoção do robô define-se um serviço básico de locomoção que possibilita a operação do robô através da invocação remota de métodos, sem

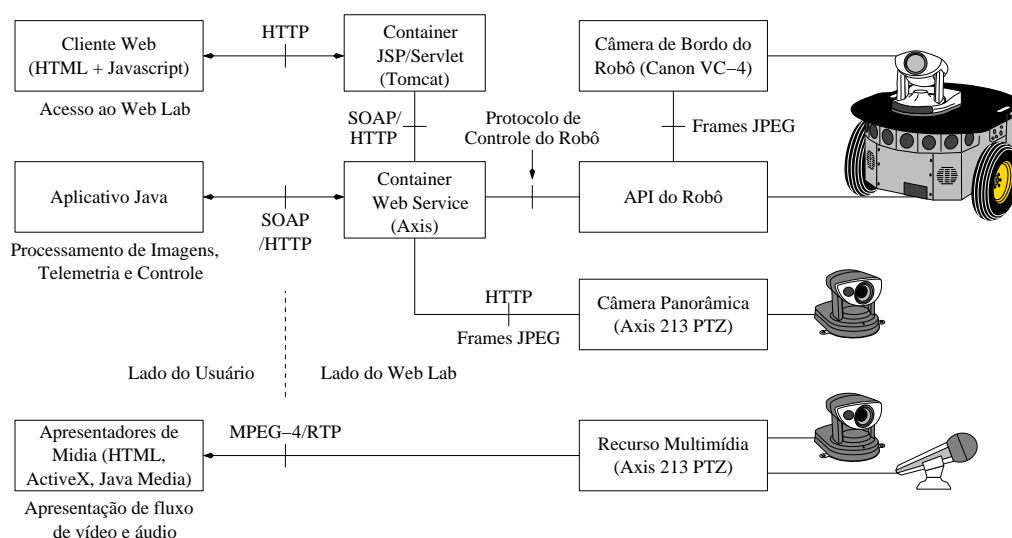


Figura 3. Infra-estrutura do GigaBOT Web Lab.

necessidade de nenhum conhecimento prévio da extensa interface de programação (API) nativa do robô. Este controle permite mover o robô para uma posição definida (relativa ou absoluta) de forma síncrona (bloqueante durante a movimentação) ou assíncrona.

O serviço de telemetria permite ao usuário a obtenção de informações tais como: posição atual do robô, leitura dos sonares, dimensões do robô, número de sonares, raio mínimo de curvatura e tensão mantida pela bateria.

O serviço de visão permite controlar a câmera embarcada no robô. Entre as grandezas controladas estão os valores, tanto absolutos quanto relativos, de *pan* (movimentação horizontal), de *tilt* (movimentação vertical), de *zoom* e de foco. O serviço permite ainda a captura de uma imagem estática no formato JPEG.

O serviço de ações mapeia algumas ações da API do robô para serviços Web. Uma ação estabelece um conjunto de operações básicas que uma vez realizadas pelo robô realizam uma tarefa específica, por exemplo, locomoção com desvio de obstáculos.

Finalmente, o serviço de submissão de código permite ao usuário submeter e executar código C++, Java ou Python no servidor diretamente conectado ao robô. As operações contempladas por este serviço possibilitam também o monitoramento do programa em execução por meio da obtenção de suas saídas padrão e de erro, bem como da possibilidade de terminar a execução a qualquer instante. Para utilizar este serviço o usuário deve utilizar a API do robô.

4.2. Experimentos Implementados

Os experimentos disponibilizados pelo GigaBOT Web Lab são programados por meio da composição dos cinco serviços de interação mais os serviços de acesso e de comunicação. Dependendo do nível e objetivo do curso, o usuário pode implementar os algoritmos necessários à realização dos experimentos ou apenas ajustar parâmetros para algoritmos fornecidos. Conforme o experimento, os algoritmos implementados podem executar remotamente no terminal do usuário ou nos processadores de controle dos robôs disponibilizados no Web Lab.

Inicialmente foram desenvolvidos cinco classes de experimentos para o Web Lab. A primeira classe de experimentos, Movimentação e Telemetria, permite ao usuário familiarizar-se com os robôs. Estes experimentos são programados por meio da composição dos serviços de locomoção, telemetria e comunicação. A Figura 4 ilustra esta composição e apresenta uma interface com duas imagens de câmeras (panorâmica e de bordo). Um mapa de navegação no lado superior direito apresenta um ícone do robô, a leitura dos sonares (linhas emanando do ícone) e obstáculos detectados pelos sonares até o momento (retângulos em torno do ícone). O mapa é atualizado por eventos de telemetria que o robô submete periodicamente ao serviço de difusão. Abaixo do mapa, um painel de navegação permite ajustar velocidade e ângulo de giro do robô, iniciar/parar a movimentação, zerar a posição do robô e inspecionar a tensão suprida pela bateria. Ainda neste experimento, o robô pode ser movimentado segundo uma trajetória traçada com o mouse no painel de navegação. Com a realização do experimento o usuário obtém noções da dinâmica e das capacidades de navegação, bem como da telemetria do robô.

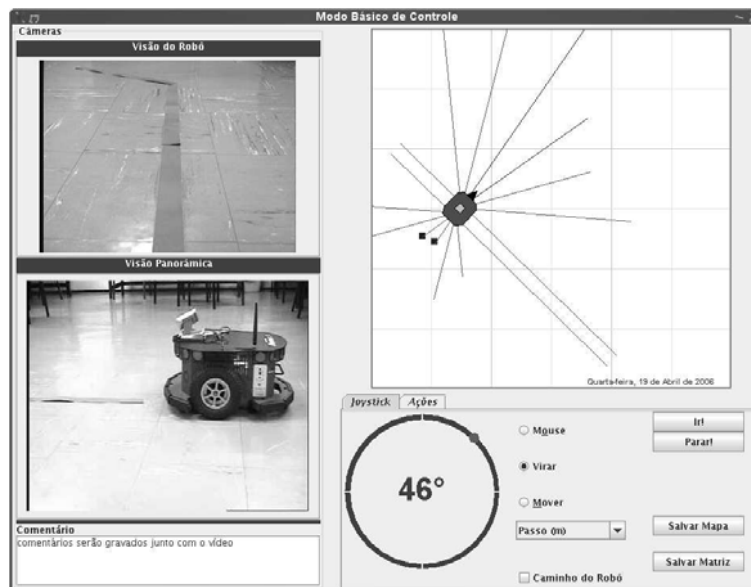


Figura 4. Interface de telemetria básica.

A segunda classe de experimentos, Navegação em Ambientes Estruturados, tem por objetivo proceder o mapeamento de ambientes e a determinação de trajetórias ótimas. Estes experimentos são programados compondo-se os serviços de ações, telemetria e comunicação. O mapeamento do ambiente requer duas ações simultâneas: navegação exploratória e registro das distâncias dos obstáculos fornecidas pelos sonares.

A terceira classe de experimentos, Navegação em Ambientes Não Estruturados, tem por objetivo navegar o robô até um ponto estabelecido no mapa de navegação sem que o ambiente tenha sido mapeado. Estes experimentos são implementados via composição dos serviços de telemetria, locomoção e comunicação.

A quarta classe de experimentos, Navegação por Visão, explora as potencialidades do sistema de visão do robô. Estes experimentos são implementados por meio da composição dos serviços de visão, locomoção e comunicação. Um experimento clássico é fazer com que o robô siga uma fita colorida disposta no piso. Um ciclo de controle

tem por objetivo manter a fita sempre no centro da imagem. O ciclo de controle inicia com a invocação do serviço de visão para capturar de uma imagem da câmera de bordo do robô. Um processamento desta imagem extrai o padrão da fita e determina a posição relativa do robô em relação à fita. Com base na posição relativa, uma correção na rota do robô é computada. O serviço de locomoção é então invocado para direcionar o robô para a nova rota, quando um novo ciclo de controle se inicia. Todo o controle executa no computador do usuário, o que torna o experimento altamente sensível ao atraso imposto pela rede (quanto maior o atraso menor deve ser a velocidade de locomoção do robô e a frequência do ciclo de controle). A Figura 5 ilustra a interface do experimento. A imagem a esquerda mostra a visão da câmera com a fita destacada em preto. A imagem ao centro mostra as bordas da fita detectadas. A imagem a direita mostra a trajetória computada a partir das duas imagens anteriores. O processamento da imagem é suportado pelo sistema *Java Advanced Imaging (JAI)*.

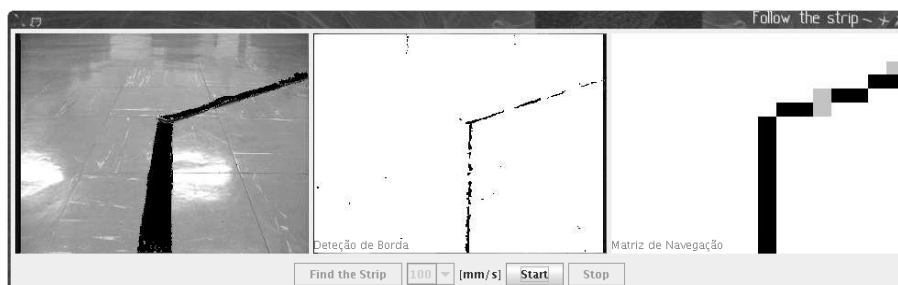


Figura 5. Experimento de navegação por visão.

A quinta classe de experimentos contempla a Cooperação de Robôs, onde, por exemplo, efetua-se o mapeamento do ambiente utilizando simultaneamente dois robôs.

Os experimentos descritos acima utilizam serviços Web para acesso às funcionalidades do robô. Uma alternativa, propiciada pelo serviço de submissão de código, é programar experimentos que utilizam diretamente a API do robô. Como este código não possui qualquer limitação, uma infinidade de temas podem ser explorados, por exemplo, controle e automação de sistemas, sistemas de tempo real, visão robótica e sistemas inteligentes.

5. Avaliação

Web Labs são aplicações interativas onde altas taxas de informação são trocadas entre o Web Lab e o usuário remoto. O usuário recebe do Web Lab fluxos de vídeo, áudio, imagens e telemetria. O Web Lab recebe do usuário fluxos de controle e, em certos casos, fluxos de vídeo e áudio. O atraso imposto pela rede tem influência marcante no experimento, notadamente quando o controle do experimento executa no computador do usuário. A fim de avaliar a interatividade do Web Lab apresentado acima, um experimento foi instrumentado e avaliado em quatro cenários de acesso: via rede local, rede de campus, rede privada virtual (VPN) e ADSL (*Asymmetric Digital Subscriber Line*).

O experimento de navegação por visão foi instrumentado para monitorar a banda de rede utilizada pelo experimento e o limite de desempenho do mesmo. Este limite é dado pela velocidade máxima com que o robô consegue seguir uma fita com dois segmentos retos dispostos em ângulo de 45°. Acima desta velocidade, o robô não

consegue processar imagens com frequência suficiente para corrigir a rota na junção dos dois segmentos da fita (situação em que o robô “perde” a fita). Esta velocidade máxima depende da frequência do ciclo de controle, que por sua vez depende da vazão e do atraso da rede, bem como da capacidade de processamento no terminal do usuário. A vazão determina o fluxo máximo de telemetria que se consegue obter do robô. O atraso da rede combinado com a capacidade de processamento do terminal do usuário determinam o atraso na atuação do controle. Como terminal de acesso foi utilizado em todos os cenários um computador portátil Dell D520 com processador Intel Centrino Duo de 1.8 Mhz e 2GB de memória RAM.

Os resultados para os cenários avaliados estão apresentados nas tabelas a seguir. Deve-se salientar que os testes conduzidos nos vários cenários oferecem uma visão comparativa e ilustrativa, sendo insuficientes para uma análise metodológica e elaborada do desempenho destas redes no suporte a Web Labs. As redes não foram instrumentadas, para, por exemplo, proceder uma medição do atraso e largura máxima de banda.

Na tabela 1, acesso via conexão cabeada à rede local do Web Lab, a velocidade máxima do robô foi 180 mm/s, com frequência de controle de 3 ciclos/s. Na tabela 2, acesso via conexão sem fio à rede local do Web Lab, a velocidade máxima foi de 170 mm/s, com frequência de controle também de de 3 ciclos/s. Como esperado, o desempenho do experimento para redes locais Ethernet com e sem fio foram semelhantes, dado que ambas possuem largura de banda e atraso adequados para aplicações de Web Labs. Graças a boa largura de banda das redes locais, a apresentação do fluxo multimídia não impacta no desempenho do experimento nestas redes. O consumo de banda do recurso multimídia foi estimado com base no gerenciador de tarefas do Windows. Esta ferramenta fornece a banda total consumida em termos de percentuais da capacidade da interface de rede e sua precisão permite apenas estimar ordens de grandeza (razão pela qual o valor é dado em múltiplos de 1000 Kbps).

Vel. (mm/s)	Telemetria (Kbps)	Câm. Robô (Kbps)	Câmera Pan. (Kbps)	Rec. Mutimídia (Kbps)	Total (Kbps)
150	748,90	716,86	125,64	22.000	23.591,40
160	717,19	680,24	125,54	22.000	23.522,97
170	741,06	561,95	125,21	22.000	23.428,22
180 (max)	731,80	553,86	126,04	22.000	23.411,70
Média	734,74	628,23	125,61	22.000	23.488,57

Tabela 1. Desempenho para Rede Ethernet de 100 Mbps.

Vel. (mm/s)	Telemetria (Kbps)	Câm. Robô (Kbps)	Câmera Pan. (Kbps)	Rec. Mutimídia (Kbps)	Total (Kbps)
100	495,01	507,28	122,16	15.200	16.324,46
150	494,33	484,53	122,51	15.200	16.301,37
170 (max)	502,74	416,32	122,72	15.200	16.241,78
180	475,93	436,66	123,23	15.200	16.235,81
Média	492,00	461,20	122,66	15.200	16.275,86

Tabela 2. Desempenho para Rede 802.11g de 54 Mbps.

Na tabela 3, acesso via rede de campus, obteve-se velocidade máxima de 90 mm/s com frequência do ciclo de controle de 2 ciclos/s. A apresentação do fluxo multimídia

teve pouco impacto no desempenho do experimento. A rede de campus utilizada foi a Uninet que interliga as unidades de ensino, pesquisa e administração no campus da Universidade Estadual de Campinas. Esta rede possui núcleo de roteadores conectados via enlaces Ethernet de 1 Gbit/s. As conexões com as unidades se dá por meio de enlaces Ethernet de 100 Mbit/s. Esta rede não permite ao usuário reservar recursos.

Vel. (mm/s)	Telemetria (Kbps)	Câm. Robô (Kbps)	Câmera Pan. (Kbps)	Rec. Mutimídia (Kbps)	Total (Kbps)
50	60,76	154,95	46,90	1.000	1.262,61
80	57,46	186,85	69,17	1.000	1.313,48
90 (max)	67,18	124,88	61,86	1.000	1.253,92
100	60,33	221,29	77,22	1000	1.358,84
120	71,32	140,78	50,01	1.000	1.262,11
Média	63,41	165,75	61,03	1.000	1.290,19

Tabela 3. Desempenho para rede de campus.

Na tabela 4, acesso via VPN, observa-se um desempenho ligeiramente superior àquele obtido para rede local. Aumentando-se a frequência do ciclo de controle para 4 ciclos/s, obteve-se uma velocidade máxima de 200 mm/s. O aumento da frequência do ciclo de controle foi possível graças ao baixíssimo atraso propiciado pela VPN. Graças à grande largura de banda e enlaces dedicados, a apresentação do fluxo multimídia não impacta no desempenho do experimento para esta rede.

A rede privada utilizada para teste foi a rede Giga operada pela Rede Nacional de Ensino e Pesquisa (RNP) em parceria com a Fundação CPqD. A rede Giga conecta instituições no eixo São Paulo-Rio de Janeiro com enlaces em fibra ótica de 1 Mbps (Gigabit Ethernet). O acesso foi realizado a partir do Instituto de Computação da Unicamp, estando o Web Lab conectado ao nó da rede Giga na Faculdade de Engenharia Elétrica e de Computação da Unicamp. No roteador de acesso à rede Giga foram utilizadas 4 portas de 100 Mbps. Estas portas conectam o servidor do Web Lab, o ponto de acesso IEEE 802.11g para conexão ao robô, a câmera panorâmica e o recurso multimídia.

Vel. (mm/s)	Telemetria (Kbps)	Câm. Robô (Kbps)	Câmera Pan. (Kbps)	Rec. Mutimídia (Kbps)	Total (Kbps)
150	680,19	604,76	285,58	22.000	23.570,53
180	671,66	551,21	290,11	22.000	23.512,98
190	706,42	561,58	291,90	22.000	23.559,90
200 (max)	735,29	692,21	285,18	22.000	23.712,68
210	681,87	543,89	297,89	22.000	23.523,65
Média	695,08	590,73	290,13	22.000	23.575,95

Tabela 4. Desempenho para VPN (rede Giga).

O fato do desempenho da rede Giga ser ligeiramente superior ao da rede local pode ser explicado por dois fatores. Primeiro, diferentemente da rede local, a rede Giga estava integralmente dedicada ao experimento. Segundo, o terminal do cliente estava conectado à rede Giga por cabo, o que propicia uma comunicação muito mais eficaz que a rede sem fio empregada no acesso via rede local.

Na tabela 5, acesso residencial via ADSL de 2 Mbps, a velocidade máxima foi de apenas 30 mm/s, com frequência de controle de 0,5 ciclo/s. Nota-se aqui o efeito da baixa

largura de banda e o elevado atraso do acesso residencial. Apesar da velocidade máxima de acesso ser de 2 Mbps, a banda máxima utilizada foi de 1 Mbps, devido, provavelmente, a gargalos nas diversas redes que separam o provedor de acesso e o Web Lab.

Na tabela 6, com o mesmo acesso residencial, a velocidade máxima foi de 70 mm/s, com frequência de controle de 1 ciclo/s. Este desempenho superior ao caso anterior se deve à não apresentação do fluxo multimídia no terminal do usuário. Diferentemente da rede de campus, a apresentação de fluxo multimídia teve impacto significativo no desempenho do experimento.

Vel. (mm/s)	Telemetria (Kbps)	Câm. Robô (Kbps)	Câmera Pan. (Kbps)	Rec. Mutimídia (Kbps)	Total (Kbps)
30 (max)	35,90	92,42	51,28	800	979,61
40	62,03	128,85	63,21	800	1.054,09
60	47,37	106,13	68,80	800	1.022,30
80	31,59	54,30	23,60	800	909,48
100	35,31	120,09	46,51	800	1001,90
Média	42,44	100,36	50,38	800	993,48

Tabela 5. Desempenho para rede ADSL de 2 Mbps com recurso multimídia.

Vel. (mm/s)	Telemetria (Kbps)	Câm. Robô (Kbps)	Câmera Pan. (Kbps)	Rec. Mutimídia (Kbps)	Total (Kbps)
70 (max)	49,09	96,92	54,34	0	200,34
80	54,28	94,71	49,63	0	198,62
100	50,25	85,96	35,68	0	171,90
Média	51,20	92,53	46,55	0	190,29

Tabela 6. Desempenho para rede ADSL de 2 Mbps sem recurso multimídia.

Em redes com restrições de largura de banda e atraso, como as redes roteadas de campus sem capacidade de reserva de recursos e redes de acesso residenciais, pode-se empregar o esquema da Figura 6 (parte superior). Para estas redes, o controle do experimento deverá ser executado no Web Lab, que por sua vez disponibiliza uma interface Web para o usuário. Por meio desta interface, o usuário pode controlar o experimento e acompanhar sua execução. Apesar desta solução ser comum nas implementações de Web Labs, o controle descentralizado no domínio do usuário (parte inferior da Figura 6) se constitui numa solução muito mais rica, onde o usuário pode programar seu próprio experimento via composição de serviços Web. Os dois modos de acesso podem ser oferecidos pelo Web Lab, em função do tipo de rede de acesso empregada pelo usuário.

6. Conclusões

Computação orientada a serviço é um paradigma de computação distribuída que certamente irá impactar positivamente em muitas aplicações, notadamente aplicações que se distribuem por múltiplos domínios administrativos (federações). No caso de Web Labs, este paradigma facilita a interação inter-domínios por utilizar protocolos universalmente aceitos como HTTP e HTTP Seguro.

Serviços Web eliminam a necessidade de utilização de sistemas de software proprietários no terminal do usuário. Graças a um mecanismo padrão de composição

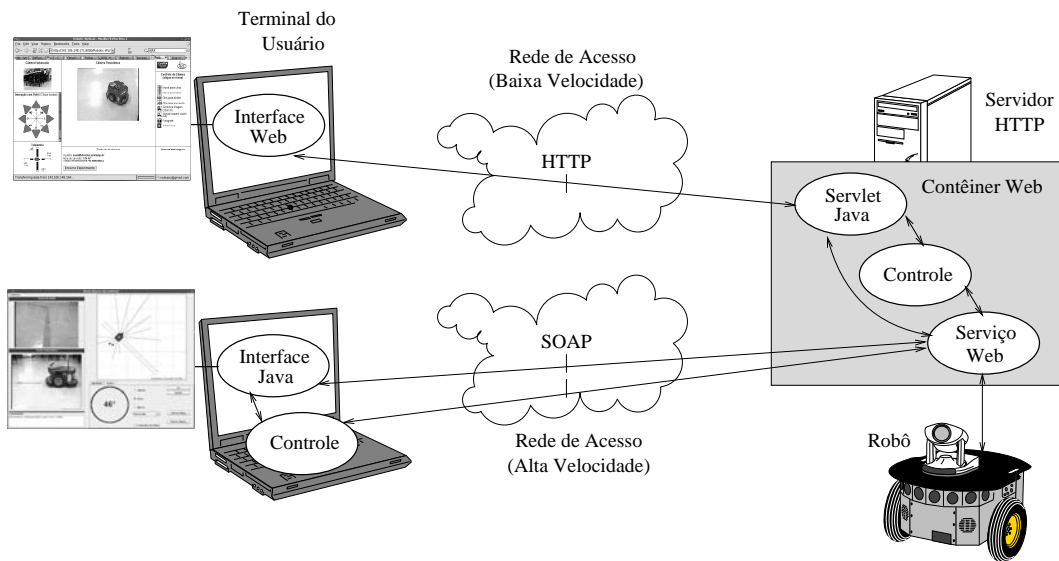


Figura 6. Formas de acesso ao Web Lab em baixa velocidade (acima) e em alta velocidade (abaixo).

(ausente nos demais paradigmas de computação distribuída), computação orientada a serviço permite a criação de novos serviços por meio da combinação de serviços existentes. No caso de Web Labs, novos experimentos podem ser criados via composição de serviços existentes. O maior benefício de computação orientada a serviço está na possibilidade de utilização de aplicações de forma federada. Um modelo de federação baseado em composição de serviços inter-domínios aliado a mecanismos de segurança para serviços Web está em fase de desenvolvimento.

A infra-estrutura baseada em SOA descrita neste artigo tem facilitado o desenvolvimento de Web Labs na FEEC, CenPRA e UFU, devido a disponibilização dos serviços de acesso e de comunicação independentes de domínio de aplicação. O uso de ferramentas como ActiveBPEL facilitam a criação de experimentos por meio da composição de novos serviços de interação, bem como dos serviços existentes (reuso de serviços). O Web Lab de robótica descrito neste artigo atenderá a um curso de pós-graduação em robótica móvel a ser ministrado, em parceria com o CenPRA, na FEEC/Unicamp.

Finalmente, observa-se que Web Labs que operam equipamentos com restrições de tempo real demandam redes de alto desempenho. Consta-se que redes de acesso com limitações de banda e alto atraso se mostram inadequadas para Web Labs. É o caso de redes roteadas de campus sem capacidade de reserva de recursos e redes de acesso residenciais ADSL e cabo. Para estas redes, resta o modo de operação onde o controle do experimento executa no próprio Web Lab (e não no terminal do usuário). Estamos desenvolvendo estratégias que permitem adaptar o experimento em função das capacidades da rede de acesso dos usuários. Outra limitação que está sendo investigada é a ausência do uso federado de recursos. Uma arquitetura do serviço de acesso federado encontra-se na fase de projeto.

Agradecimentos

Este trabalho é suportado pela Rede Nacional de Ensino e Pesquisa (RNP) através do Projeto Giga (subprojetos 2463 e 2460). Os autores agradecem os alunos de iniciação científica Victor Peticarrari, Danielle R. Santos e Ewerton Barbosa pelo auxílio no desenvolvimento de software.

Referências

- Apache Software Foundation (2006). *Apache Projects*. <http://www.apache.org>.
- Casini, M., Prattichizzo, D., and Vicino, A. (2004). The Automatic Control Lab. *IEEE Control Systems Magazine*, 25(1).
- Coelho, P., Sassi, R., Cardozo, E., Guimarães, E., Faina, L., Lima, A., and Pinto, R. (2007). A Web Lab for Mobile Robotics Education. In *IEEE International Conference on Robotics and Automation*, Roma, Italia.
- Curbera, F., Khalaf, R., Mukhi, N., Tai, S., and Weerawarana, S. (2003). The Next Step in Web Services. *Comm.of the ACM*, 10(46).
- Denis, D., Bulancak, A., and Ozcan, G. (2003). A Novel Approach to Remote Laboratories. In *ASE/IEEE Frontiers in Education Conference*, Boulder, USA.
- Guimarães, E., Maffei, A., Pereira, J., Russo, B., Cardozo, E., Bergerman, M., and Magalhães, M. (2003a). REAL: A Virtual Laboratory for Mobile Robots Experiments. *IEEE Transactions on Education*, 46(1).
- Guimarães, E., Maffei, A., Pinto, R., Miglinski, C., Cardozo, E., Bergerman, M., and Magalhães, M. (2003b). REAL: A Virtual Laboratory Built from Software Components. *Proceedings of the IEEE*, 91(3).
- Hua, J. and Ganz, A. (2003). Web Enabled Laboratory (R-LAB) Framework. In *ASEE/IEEE Frontiers in Education Conference*, Boulder, USA.
- Khamis, A., Rivero, D. M., Rodríguez, F., and Salichs, M. (2003). Pattern-based Architecture for Building Mobile Robotics Remote Laboratories. In *IEEE International Conference on Robotics and Automation*, Taiwan.
- MIT (2006). Projeto iCampus/iLAB. <http://icampus.mit.edu/ilabs>.
- Papazoglou, M. P. and Georgakopoulos, D. (2003). Service-oriented computing: Introductions. *Comm.of the ACM*, 10(46):24–28.
- Sassi, R., Coelho, P., Cardozo, E., Guimarães, E., Faina, L., and Pinto, R. (2006). Uma Plataforma de Middleware para Geração de Software Orientado a Componentes. In *VI Workshop de Desenvolvimento Orientado a Componentes (WDBC'2006)*, Olinda, PE.
- Yan, Y., Liang, Y., Du, X., Saliyah-Hassane, H., and Ghorbani, A. (2006). Putting Labs Online With Web Services. *IEEE IT Professional*.