

6 Experimentos

Como estudo de caso da arquitetura desenvolvida, o CAS foi utilizado em dois ambientes para realizar sete capturas distintas em dois grandes eventos: um encontro de Python científico na Petrobras³² e aulas da disciplina de Seminários de Sistemas Distribuídos realizadas na sala Active Classroom, que está sendo montada pelo Departamento de Informática³³ da PUC-Rio³⁴ em parceria com a Microsoft Research³⁵. Neste capítulo são detalhados esses eventos, apresentando para cada um os principais problemas enfrentados bem como alguns exemplos de documentos gerados. Ao final é feita uma avaliação subjetiva do CAS.

6.1. Encontro de Python Científico na Petrobras

No dia 20 de fevereiro foi realizado o primeiro encontro nacional da comunidade de Python científico do Brasil³⁶ dentro das instalações do Centro de Pesquisas (CENPES) da Petrobras. Foram capturadas cinco apresentações distintas em uma sala de visualização chamada de Espaço Galileu.

A próxima subseção analisa detalhes da configuração do ambiente e problemas encontrados na infra-estrutura de captura. As demais subseções apresentam os pormenores de cada evento e detalham aspectos interessantes observados, incluindo alguns exemplos dos documentos gerados ao longo das quatro fases seguidas pelo CAS.

³² <http://www.petrobras.com.br>

³³ <http://www.inf.puc-rio.br>

³⁴ <http://www.puc-rio.br>

³⁵ <http://research.microsoft.com>

³⁶ <http://www.pythonbrasil.com.br>

6.1.1. Configurações e Problemas de Infra-Estrutura

Apesar do Espaço Galileu possuir vários recursos computacionais, como projetores estereoscópicos com projeção traseira e microfones sem fio, em termos de recursos de vídeo ele só possui uma filmadora digital JVC e *webcams*. Dado que a versão atual do CAS suporta apenas câmeras IP, optou-se por pegar emprestadas as duas câmeras IP sobressalentes da Active Classroom, para que a captura pudesse ser realizada sem maiores problemas.

A configuração das câmeras IP dentro desse ambiente foi um problema a parte, pois cada ponto de rede da sala estava configurado em uma sub-rede diferente. Desta forma o aplicativo D-Link Installation Wizard, que localiza as câmeras por *broadcast* e realiza sua configuração inicial definindo os parâmetros de rede como o IP, não conseguia localizar as câmeras. Foi necessário o acionamento da equipe de suporte à rede da Petrobras para a reconfiguração dos três pontos de rede disponíveis para a mesma sub-rede. Esta configuração da infra-estrutura básica está fora do escopo do CAS, mas mostra que o trabalho de instalação do CAS pode ser muito complicado por fatores externos.

Optou-se por utilizar três câmeras neste evento: as duas câmeras IP da PUC-Rio e a câmera JVC da sala. Das câmeras IP emprestadas pela PUC-Rio, uma ficou disposta no fundo da sala capturando uma panorâmica, enquanto a outra ficou no canto da sala junto à tela de projeção, configurada em seu modo automático de patrulha (*auto patrol*), capturando imagens num ângulo de 90° da tela. Foi decidido utilizar também a filmadora digital JVC disponível no fundo da sala para fazer panorâmicas e closes dos apresentadores. Como o desenvolvimento de um SpeedCar para a leitura de sua conexão *firewire* seria algo demorado, optou-se por parar o Post-Production Service, postergando sua execução. Desta forma, após a captura de cada evento o arquivo de vídeo desta câmera poderia ser copiado manualmente para a pasta do evento, de forma que, quando o serviço fosse iniciado, ele percebesse três mídias de vídeo, e não apenas as duas geradas pelos SpeedCars. Esta abordagem acabou não sendo tão rápida como se esperava,

pois a câmera JVC gera na verdade um arquivo do tipo MOD e não um MPEG. Desta forma, apenas no dia seguinte ao evento pôde-se criar uma aplicação que convertia os arquivos MOD em MPEG-2 extraindo as informações de *timestamp* embutidas no MOD para um arquivo XML padrão do CAS.

Apesar do trabalho de configuração para o uso das três câmeras, sendo duas automatizadas com o uso de SpeedCars e uma por um processo manual, tudo funcionou bem e o Post-Production Service, quando ativado alguns dias depois do encontro de Python, recuperou e processou todos os eventos com as três câmeras. Embora o CAS tenha uma ferramenta de conversão de arquivos de filmadoras JVC, ela é manual e exige uma pausa no fluxo normal de trabalho. Esse experimento demonstrou a necessidade de desenvolvimento de mais componentes de captura automatizados. Além de um SpeedCar para captura ao vivo de câmeras *firewire*, o CAS precisa suportar *webcams* e até captura da tela para cobrir um maior escopo de combinações entre aplicações e ambientes utilizados.

Outro aspecto problemático nesse evento foi a restrição do CAS só trabalhar com mídias em formato PowerPoint®. Como a maioria dos apresentadores eram pesquisadores que trabalham com Linux, 80% das apresentações haviam sido geradas em LaTeX e exportadas para PDF. Como as apresentações foram recebidas na véspera, foi iniciado o desenvolvimento de um SpeedCar PDF, mas devido a alguns problemas para a exibição em tela cheia que exigiriam mais tempo, criou-se outra ferramenta auxiliar do CAS que converte PDFs em PPTs. Esta ferramenta exporta todas as páginas do arquivo PDF como imagens e insere cada uma num *slide* de um documento PPT. Esta ferramenta foi desenvolvida usando o Adobe® Acrobat® SDK (*Software Development Kit*), e além das imagens, recupera o primeiro texto de cada página do PDF para ser adicionado como título do slide. Desta forma tanto a apresentação pôde ser capturada com o uso do SpeedCar PPT, quanto os transformadores funcionaram corretamente, já que havia títulos extraídos do PPT. Novamente é exposta a necessidade de desenvolvimento de mais SpeedCars.

6.1.2. Primeira Palestra

A primeira palestra capturada não teve bons resultados na geração de contextos pela falta de padronização dos títulos. Como observado na Tabela 6.1, cada *slide* possui um título diferente, ocasionando a geração de um contexto para cada *slide* pelos transformadores do CAS. Como o autor não colocou um sumário, não é possível precisar a quantidade de contextos ideal ficando a critério do leitor este desafio.

Número do Slide	Título do Slide
1	Considerações sobre a construção e simulação de modelos matemáticos/computacionais de grande porte usando Python
2	Delimitando o Problema
3	Desafios
4	Python está a altura de desafios?
5	Processamento Distribuído
6	Exemplo lpython1
7	Exemplo Parallel-Python
8	Gerenciando grandes volumes de dados
9	Usando o ZODB
10	Instalando o ZODB
11	Usando o ZODB
12	Usando ZODB
13	Usando ZODB – Lidando com grandes coleções
14	ZODB – desvantagens
15	Conclusão

Tabela 6.1: Títulos dos *slides* do primeiro evento do encontro de Python. Observe que não há lógica para a construção de transformadores que contemple casos como este.

Outro problema da captura desse evento foi o acesso durante a apresentação a alguns *web sites*. Os *slides* 6 e 7 de exemplos não possuem nenhum conteúdo e o apresentador acessou diferentes sites com listagens de exemplo. Como o CAS não consegue capturar este tipo de ação, as câmeras foram focalizadas na projeção para garantir o registro, mesmo que não da melhor forma.

6.1.3. Segunda palestra

Esse evento não apresentou nenhuma espécie de problema. O apresentador utilizou um material com títulos muito bem padronizados, permitindo ao CAS detectar 23 contextos em 92 slides. Como não havia um sumário na apresentação, não temos como precisar o número ideal, mas analisando os contextos gerados, certamente este número poderia ser um pouco menor.

6.1.4. Terceira Palestra

Apesar de ser uma palestra grande, com 101 slides, como havia uma estrutura muito bem definida com títulos e subtítulos, o CAS detectou apenas 10 contextos como ilustrado anteriormente pela Figura 5.6. O ideal, segundo o sumário da apresentadora, seria apenas seis. No quesito de recursos do CAS, a ausência de um mecanismo de captura de tela e/ou de arquivos de vídeo não permitiu registrar os 3 vídeos reproduzidos ao longo da apresentação e que estavam com elos nos respectivos *slides*.

6.1.5. Quarta Palestra

Esta palestra foi a única do encontro de Python em que o apresentador utilizou o Microsoft PowerPoint[®], não sendo necessário o uso da ferramenta de conversão de PDF para PPT. Independente do formato, a apresentação não possuía uma estrutura de títulos bem definida e o CAS identificou pelos seus transformadores 16 contextos em um total de 31 *slides*.

6.1.6. Quinta Palestra

A última palestra do encontro de Python também não teve um bom aproveitamento dos transformadores, com 8 contextos detectados em um total de 15 *slides*. Esta apresentação em particular foi gerada a partir do LaTeX com uma estrutura de navegação com índices (*Table of Contents* – TOC). Apesar de não ter um slide com o sumário pelo TOC pode-se observar que o autor definiu apenas 4

tópicos principais. Um SpeedCar que processasse arquivos LaTeX e extraísse essas informações da apresentação seria muito interessante. Além disso, esta apresentação teve uma demonstração em vídeo que também não pode ser capturada automaticamente.

6.2. Aulas de Seminários de Sistemas Distribuídos

O CAS também foi utilizado para capturar duas aulas da disciplina de Seminários de Sistemas Distribuídos da PUC-Rio. As aulas foram realizadas na sala de aula inteligente da PUC-Rio, chamada de Active Classroom, montada com o patrocínio da Microsoft Research. O mais interessante desse experimento foi a diversidade de estruturação de documentos, pois cada aula é apresentada por um aluno diferente.

Assim como no experimento anterior, a próxima subseção analisa detalhes da configuração do ambiente e problemas encontrados na infra-estrutura de captura. As demais subseções apresentam os pormenores de cada evento e detalham aspectos interessantes observados, incluindo alguns exemplos dos documentos gerados ao longo das quatro fases seguidas pelo CAS.

6.2.1. Configurações e Problemas de Infra-Estrutura

A Active Classroom serviu como ambiente de teste durante todo o desenvolvimento do CAS, e também orientou várias decisões de projeto, como o desenvolvimento do SpeedCar D-Link Vídeo, visto que a sala possui 4 câmeras PTZ da D-Link. Desta forma os eventos realizados na Active Classroom, foram capturados pelo CAS sem nenhum problema de infra-estrutura já que todos os dispositivos utilizados eram compatíveis com os SpeedCars disponíveis.

No aspecto de utilização também não foi observado nenhum problema. Todas as apresentações sempre foram realizadas com arquivos no formato PowerPoint®. Nenhum apresentador utilizou durante seu evento outras mídias como vídeos, *sites* na internet ou mesmo várias apresentações PowerPoint®,

permitindo assim a captura integral dos eventos pela infra-estrutura disponível do CAS.

6.2.2. Primeiro Seminário

O primeiro seminário de sistemas distribuídos realizado na Active Classroom ocorreu dia 19 de março de 2008. Devido a um pequeno engano com o endereço IP do serviço SpeedCar Launcher, os componentes de áudio e vídeo não conseguiram se registrar e conseqüentemente não foram iniciados no momento em que o aluno iniciou a apresentação. O problema foi percebido após uma checagem manual no diretório de gravação das mídias, mas a demora na correção introduziu uma defasagem nas mídias como ilustra a Listagem 6.1. O restante da apresentação transcorreu sem problemas, mas este engano inicial gerou um estudo de caso interessante para validar a sincronização das mídias.

```
<Media type="application/vnd.ms-powerpoint" src="R-OSGi.ppt"
seq="1">
  <StartDateTime>19/03/2008 15:14:44</StartDateTime>
  ...
<Media type="audio" src="audio.wav" seq="1">
  <StartDateTime>19/3/2008 15:22:21</StartDateTime>
  ...
<Media type="video" src="cam1.avi" seq="1">
  <StartDateTime>19/3/2008 15:26:37</StartDateTime>
  ...
<Media type="video" src="cam2.avi" seq="1">
  <StartDateTime>19/3/2008 15:26:37</StartDateTime>
  ...
```

Listagem 6.1: Compilação de fragmentos de vários arquivos XML gerados pelos SpeedCars, destacando a diferença no tempo de início das capturas.

O problema sinalizou a necessidade de o CAS ter um mecanismo mais robusto para localização dos componentes, em substituição ao uso de endereços absolutos. Um retorno para a interface com o usuário dos erros internos também é desejável. Atualmente o CAS possui um *framework* de *log* apenas em arquivo e no sistema de eventos do Windows, sem *feedback* para o usuário final.

Também é interessante destacar para esse evento, a correta identificação dos contextos pelos algoritmos de filtragem. A validação dos transformadores pode ser feita comparando os contextos identificados automaticamente na Listagem 6.2 com os assuntos enumerados pelo próprio aluno no sumário da apresentação reproduzido pela Figura 6.1. O CAS identificou 10 contextos enquanto o aluno enumerou 8 tópicos abordados. Os contextos adicionais do CAS são o primeiro *slide* com o título e o próprio *slide* de sumário. O primeiro slide é obrigatório, pois ele é a porta para a navegação seqüencial por toda a apresentação. O transformador poderia ser mais sofisticado para remover o “sumário” da lista de contextos, pois o índice de contextos gerado pelo componente de construção da NCL deve ter a mesma capacidade expressiva que uma lista de tópicos gerada pelo autor da apresentação.

```

...
<Slide SlideNumber="1" BeginOfContext="true" >
  <SlideTitle>R-OSGi&#xB;Distributed Applications
through&#xB;Software Modularization&#xB;&#xB;[Jan S. Rellermeier,
Gustavo Alono, and Timothy Roscoe]</SlideTitle>
</Slide>
<Slide SlideNumber="2" BeginOfContext="true" >
  <SlideTitle>Summary</SlideTitle>
</Slide>
<Slide SlideNumber="3" BeginOfContext="true" >
  <SlideTitle>Introduction (i)</SlideTitle>
</Slide>
<Slide SlideNumber="4">
  <SlideTitle>Introduction (ii)</SlideTitle>
</Slide>
<Slide SlideNumber="5">
  <SlideTitle>Introduction (iii)</SlideTitle>
</Slide>
<Slide SlideNumber="6" BeginOfContext="true" >
  <SlideTitle>Background</SlideTitle>
</Slide>
<Slide SlideNumber="7" BeginOfContext="true">
  <SlideTitle>OSGi (i)</SlideTitle>
</Slide>
<Slide SlideNumber="8">
  <SlideTitle>OSGi (ii)</SlideTitle>
</Slide>
<Slide SlideNumber="9" BeginOfContext="true">
  <SlideTitle>R-OSGi (i)</SlideTitle>
</Slide>
<Slide SlideNumber="10">
  <SlideTitle>R-OSGi (ii)</SlideTitle>
</Slide>
<Slide SlideNumber="11" BeginOfContext="true">
  <SlideTitle>Implementation</SlideTitle>
</Slide>
<Slide SlideNumber="12" BeginOfContext="true">

```



```

<SlideTitle>R-OSGi Deployment Tool</SlideTitle>
</Slide>
<Slide SlideNumber="13" BeginOfContext="true">
  <SlideTitle>Conclusions</SlideTitle>
</Slide>
<Slide SlideNumber="14" BeginOfContext="true">
  <SlideTitle>References</SlideTitle>
</Slide>
...

```

Listagem 6.2: Parte do XML do documento do primeiro Seminário de Sistemas Distribuídos capturado com o CAS, que exemplifica a identificação de contextos.

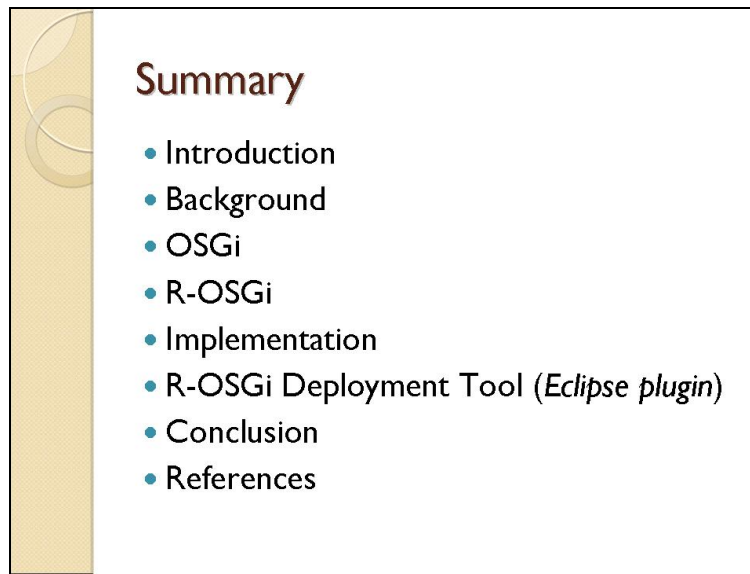


Figura 6.1: Sumário da apresentação com oito tópicos segundo a concepção do autor.

Como esse foi o menor de todos os eventos capturados e possui dados interessantes para serem analisados, ilustramos no Apêndice B todos os documentos gerados pelo CAS para esse evento.

6.2.3. Segundo Seminário

O segundo seminário de sistemas distribuídos capturado com o CAS na Active Classroom ocorreu dia 26 de março de 2008. A captura também transcorreu sem problemas, mas o fato interessante desse evento foi que o apresentador iniciou o sistema assim que copiou a apresentação para o computador, antes mesmo de toda a turma chegar. Desta forma a apresentação ficou sendo apresentada no primeiro *slide* por um longo período como ilustrado

pela Listagem 6.3, sem que de fato nada estivesse sendo apresentado. Analisando a apresentação identificamos que dos quase 18 minutos de captura apenas do primeiro slide, apenas os últimos 45 segundos apresentam informações relevantes já que aluno começou de fato a apresentar aos 1009 segundos.

```
<SlideChange seq="1">  
  <Slide>1</Slide>  
  <Start>0</Start>  
  <End>1054,9364</End>  
</SlideChange>
```

Listagem 6.3: Fragmento de código XML do SpeedCar PPT ilustrando o alto tempo (quase 18 minutos) gasto no primeiro slide durante a apresentação.

Esta particularidade mostra a necessidade do CAS oferecer a possibilidade de realizar cortes manuais na apresentação capturada. Um algoritmo para a detecção automática deste tipo de situação é difícil, pois não há muitas informações para a análise. Nesse caso específico, qualquer análise do áudio e vídeo não seria conclusiva, pois o apresentador permaneceu durante este tempo inicial se movimentando e conversando com a platéia que já estava presente.

A correspondência dos tópicos propostos pelo autor e os contextos identificados automaticamente não foi muito boa nesse evento. O autor indicou 7 tópicos no sumário da apresentação e o CAS identificou 17 contextos nos 47 *slides* que compõe o evento. Isto se deve principalmente ao fato do autor não ter sido rígido na nomenclatura dos slides sobre o mesmo assunto. Para ilustrar este comportamento, para o segundo assunto identificado como “JavaGAT” no sumário, haviam três títulos distintos: “Java Grid Application Toolkit (Java GAT)”, “Java GAT” e “JavaGAT”.

6.3. Avaliação Geral dos Experimentos Realizados

Os cinco eventos do encontro de Python, por serem os primeiros experimentos com a versão atual da infra-estrutura, apresentaram muitos problemas e exigiram um grande esforço de desenvolvimento, configuração e ajustes na véspera do evento. Um complicador foi o fato do uso de mídias não suportadas pelo CAS, no caso apresentações em PDF e arquivos da filmadora digital JVC. Apesar dos problemas terem recebido apenas soluções de contorno que não são completamente automatizadas, o resultado final ficou muito bom, viabilizando a captura do evento de uma forma superior à imaginada pelos organizadores.

O segundo experimento também foi muito rico, estressando diversas situações do sistema. O erro de configuração do primeiro evento foi ocasionado por falta de atenção aos detalhes de reconfiguração do sistema que havia sido executado anteriormente em outro espaço ativo. Com a documentação final do sistema e a geração de *wizards* de instalação, esta classe de problemas não deve se repetir.

Os vários eventos com apresentações criadas por pessoas diferentes, seguindo estilos e padrões diferentes, proporcionaram um bom material de estudo e testes para os transformadores do CAS. Apesar de não conseguir um bom desempenho global, os transformadores provaram ser eficientes quando as apresentações seguem estilos e padrões bem definidos de nomenclatura dos títulos dos *slides*. Não deve ser desconsiderado um padrão de nomenclatura como recomendação para eventos que se deseje capturar com o CAS, mas para atender ao preceito de mínima interferência de um sistema de computação ubíqua, os transformadores podem também ser refinados e enriquecidos com informações de outras mídias e dispositivos.

O objetivo de prover captura e acesso dos eventos foi atingido com sucesso nos experimentos, mas o sistema deve ser posto em um ambiente de produção

para uma avaliação mais profunda. Antes o sistema precisa tratar os problemas já identificados e implementar algumas das melhorias citadas no próximo capítulo.