

## 8. Avaliação da Abordagem Integrada

*Neste capítulo apresentamos a avaliação da nossa proposta de abordagem que integra os trabalhos pontuais relacionados aos desafios para a operacionalização da transparência de software. Foram realizadas análises quantitativa e qualitativa da aplicação da nossa abordagem. Avaliamos ainda a abordagem por construção, demonstrando a explicitação dos requisitos de transparência no código. Finalmente, comparamos o código do Lattes-Scholar transparente com versões anteriores do mesmo software.*

Os Capítulos 3 a 6 apresentaram os trabalhos relacionados aos quatro desafios para a transparência de software. O Capítulo 7 apresentou uma abordagem que integra essas contribuições, de forma a se obter o desenvolvimento intencional de software transparente baseado em argumentação. Essa abordagem foi aplicada em um estudo de caso, o Lattes-Scholar, Capítulo 7 – Seção 7.2. O Lattes-Scholar é uma aplicação *Web* de interesse do Grupo de Transparência de Software da PUC-Rio e, como tal, já possuía três versões anteriores implementadas. O Apêndice A apresenta exemplos de uso da nossa versão através de telas do Lattes-Scholar.

A primeira versão, que foi a base para o Lattes-Scholar, é a implementação de um algoritmo para ordenação, pelo número de citações, dos artigos publicados no *Workshop* de Engenharia de Requisitos (WER). Esse algoritmo foi implementado em páginas *Web* na linguagem PHP para o WER Papers (WERPAPERS 2011), e utiliza o serviço do Google Scholar. A segunda versão do Lattes-Scholar é uma aplicação *Web* implementada na linguagem Lua, disponível em (Lattes-Scholar 2011b). Essa versão utiliza o serviço do Lattes como um repositório de currículos de pesquisadores. A terceira versão do Lattes-Scholar é um SMA comportamental na plataforma JADE (Braubach et al. 2004). Essa versão está sendo adaptada para que seja possível sua execução em nuvem.

A terceira versão do Lattes-Scholar, em especial, foi construída já no contexto do Grupo de Transparência de Software, após a maturação do conhecimento sobre transparência na forma de uma das versões mais recentes do

SIG de Transparência e após o uso do método GQO (GQM adaptado). Esse último propiciou obter questões que identificam possíveis operacionalizações para cada uma das metas flexíveis relacionadas à Transparência de Software.

Avaliamos a aplicação da abordagem integrada de forma quantitativa e qualitativa, através da aplicação de um questionário preenchido pelos participantes das reuniões. O objetivo dessas avaliações foi capturar as impressões dos participantes das reuniões sobre o processo de discussão dos requisitos de transparência para o Lattes-Scholar.

Avaliamos por construção a abordagem integrada, com o objetivo de analisar se os requisitos de transparência, relativamente validados pelos interessados, estavam explícitos no código do SMA intencional. Analisamos também o código de versões anteriores para ressaltar as diferenças entre o código produzido pela nossa abordagem e o código produzido por abordagens tradicionais.

É necessário ressaltar que as funcionalidades implementadas na versão transparente do Lattes-Scholar são praticamente as mesmas das versões anteriores, uma vez que: (i) o grupo de interessados era o mesmo que foi consultado durante o desenvolvimento das três versões anteriores; (ii) o grupo de interessados já possuía conhecimento e interesse em transparência de software quando duas das versões anteriores foram desenvolvidas; e (iii) pelo menos dois dos interessados participaram da concepção do SIG de Transparência e da construção do Catálogo de Transparência de Software e atuaram, posteriormente, nos papéis de interessados e desenvolvedores da terceira versão do Lattes-Scholar.

Portanto, as diferenças percebidas entre a versão do Lattes-Scholar produzida pela nossa abordagem e as versões anteriores devem-se ao processo participativo de elicitação de requisitos de transparência através da argumentação e da nossa abordagem dirigida por heurísticas transformacionais para anexar os requisitos ao código.

Este capítulo está organizado em seções. A Seção 8.1 apresenta a avaliação quantitativa da aplicação da abordagem integrada. A avaliação qualitativa, realizada pelos participantes das reuniões, é abordada na Seção 8.2. A Seção 8.3 descreve a avaliação por construção. Comparações com versões anteriores do Lattes-Scholar são discutidas na Seção 8.4. As considerações finais são apresentadas na Seção 8.5.

## 8.1. Avaliação Quantitativa da Aplicação da Abordagem

O objetivo do estudo de caso Lattes-Scholar foi avaliar a aplicação e a abordagem integrada proposta no Capítulo 7. A avaliação foi feita através da aplicação de questionários, cujo preenchimento era facultativo por parte dos interessados.

A Figura 8.1 apresenta o questionário utilizado para avaliar a qualidade das discussões. Esse questionário era entregue aos participantes ao fim de cada reunião. Cada participante preencheu uma cópia do questionário. As discussões foram conduzidas por um mediador e as reuniões foram gravadas usando uma câmera de vídeo. O mediador e o operador da câmera de vídeo não preencheram questionários.

Após o preenchimento dos campos “Nome do participante” e “Data”, o participante preenchia a lacuna correspondente ao requisito não funcional discutido: Disponibilidade, na primeira reunião; Publicidade e Portabilidade, na segunda reunião; Rastreabilidade, na terceira reunião; Amigabilidade, na quarta reunião; Atualização, na quinta reunião; e Concisão, na sexta e última reunião.

<b>Questionário de Avaliação das Discussões</b>	
Nome do Participante: _____	Data: _____
Avaliação da discussão em relação ao requisito não-funcional _____:	
<b>Qualidade da discussão:</b>	
<input type="checkbox"/> Péssima <input type="checkbox"/> Muito Ruim <input type="checkbox"/> Ruim <input type="checkbox"/> Regular <input type="checkbox"/> Boa <input type="checkbox"/> Muito Boa <input type="checkbox"/> Excelente	
<b>Justifique:</b>	
_____	
_____	
_____	

Figura 8.1 – Questionário utilizado para avaliar a qualidade das discussões

A avaliação quantitativa tratada nessa seção refere-se à Qualidade da Discussão, classificada em sete categorias: Péssima, Muito Ruim, Ruim, Regular, Boa, Muito Boa e Excelente. A escolha da classificação da qualidade da discussão em sete categorias foi planejada de modo a oferecer uma categoria neutra, três graus de categorias positivas e três graus de categorias negativas.

O participante assinalava a categoria que melhor representasse sua impressão sobre a discussão. Os participantes também precisavam justificar sua escolha. As respostas dos participantes ao item “Justifique” foi utilizado para a análise qualitativa apresentada na Seção 8.2.

O grupo de participantes era composto por dez pessoas, dentre elas: três mestrandos, três doutorandos e quatro doutores. Todos possuíam interesse e conhecimento prévio em transparência de software, bem como no próprio software em análise, o Lattes-Scholar. Pelo menos quatro pessoas já haviam participado da elicitação de requisitos funcionais para esse software e da implementação de versões anteriores do mesmo sem uma ênfase maior na transparência.

Ao todo foram preenchidos trinta e cinco questionários. Em pelo menos três casos os participantes optaram por não preenchê-los. Na primeira reunião, foram preenchidos seis questionários para avaliar a discussão sobre o RNF Disponibilidade. A discussão sobre Publicidade, na segunda reunião, foi avaliada por quatro questionários. Ainda na segunda reunião, cinco questionários foram preenchidos para qualificar a reunião sobre Portabilidade. Seis questionários foram preenchidos na terceira reunião, que discutiu a Rastreabilidade. Na quarta reunião, sobre Amigabilidade, seis questionários foram preenchidos. Quatro participantes preencheram questionários na quinta reunião, que discutiu a Atualidade. Por último, na sexta reunião, quatro questionários foram preenchidos.

Apresentamos graficamente os dados obtidos através de histogramas e polígonos (Pereira 2004) da frequência das categorias. Analisamos estatisticamente cada histograma através de uma métrica de tendência central, a mediana (Pereira, 2004). Essa métrica-se na área das barras horizontais dos histogramas.

A Figura 8.2 apresenta o histograma da frequência das categorias assinaladas nos questionários sobre a discussão do RNF Disponibilidade. Os resultados foram: 17% Excelente; 50% Muito Boa; 17% Boa e 16% Regular. Considerando que o histograma possui assimetria negativa, a mediana pertence à categoria Muito Boa.

A discussão sobre a Publicidade foi um pouco mais positiva. A Figura 8.3 mostra um histograma simétrico para a frequência das categorias assinaladas nos questionários sobre a Publicidade. Os resultados foram: 25% Excelente, 50%

Muito Boa e 25% Boa. Como o histograma é simétrico, a mediana pertence exatamente ao centro da barra da categoria Muito Boa.

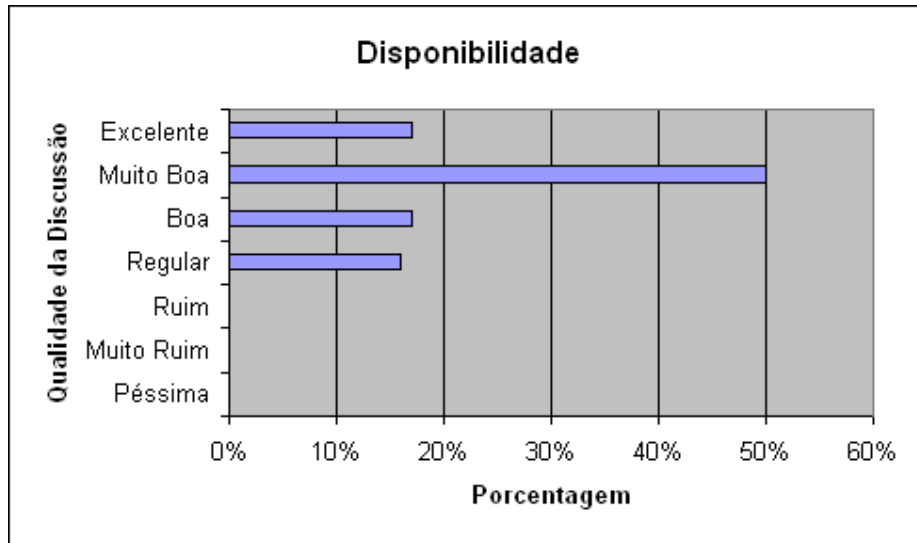


Figura 8.2 - Qualidade da discussão sobre Disponibilidade

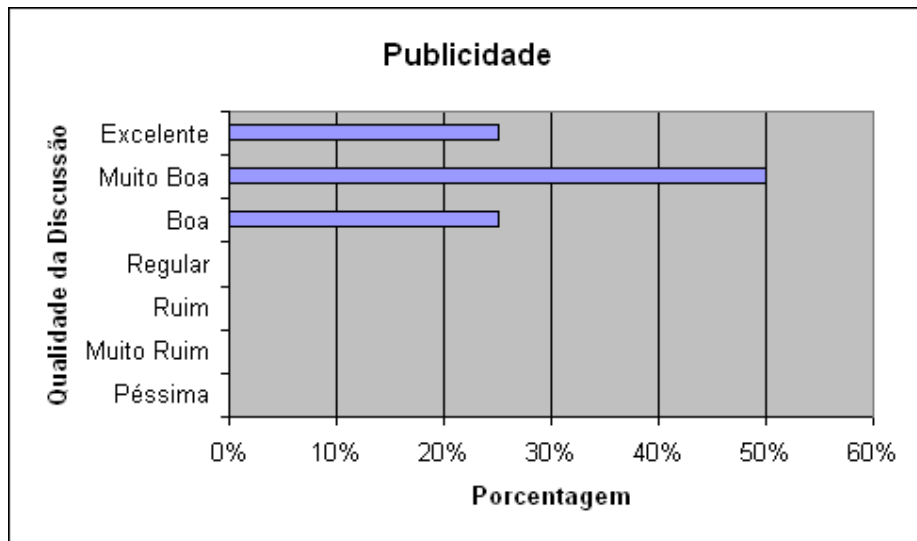


Figura 8.3 - Qualidade da discussão sobre Publicidade

O histograma negativamente assimétrico da Figura 8.4 apresenta graficamente a frequência das categorias assinaladas para a avaliação da Portabilidade. Os resultados foram: 20% Excelente, 40% Muito Boa, 20% Boa e 20% Regular. Embora a mediana seja um pouco inferior à mediana do histograma sobre a Disponibilidade, ela ainda pertence à categoria Muito Boa.

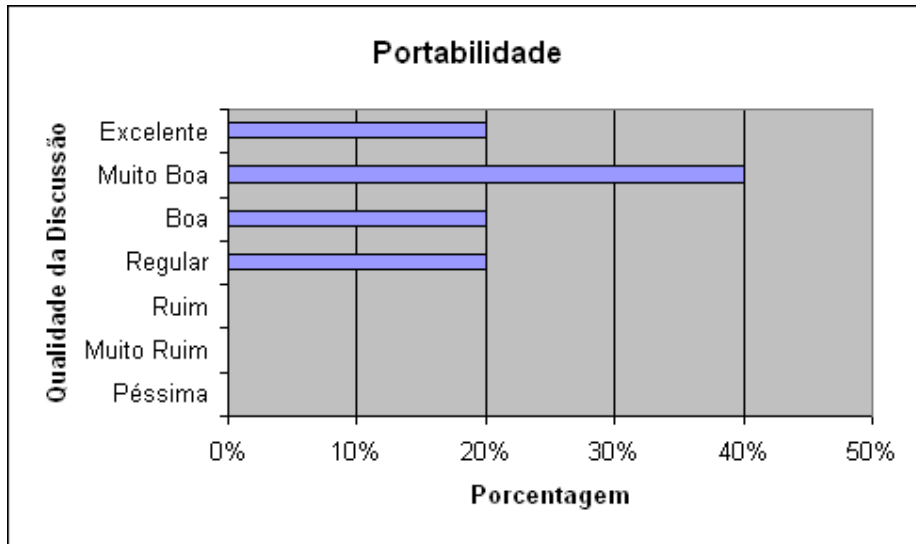


Figura 8.4 - Qualidade da discussão sobre Portabilidade

A Figura 8.5 apresenta o histograma para a frequência das categorias assinaladas para avaliar a qualidade da discussão sobre a Rastreabilidade. Os resultados foram: 34% Excelente, 50% Muito Boa e 16% Boa. O histograma apresenta assimetria positiva e a mediana mais alta de todos os histogramas analisados. A mediana pertence à categoria Muito Boa, mas está acima do centro da barra, próxima à categoria Excelente.

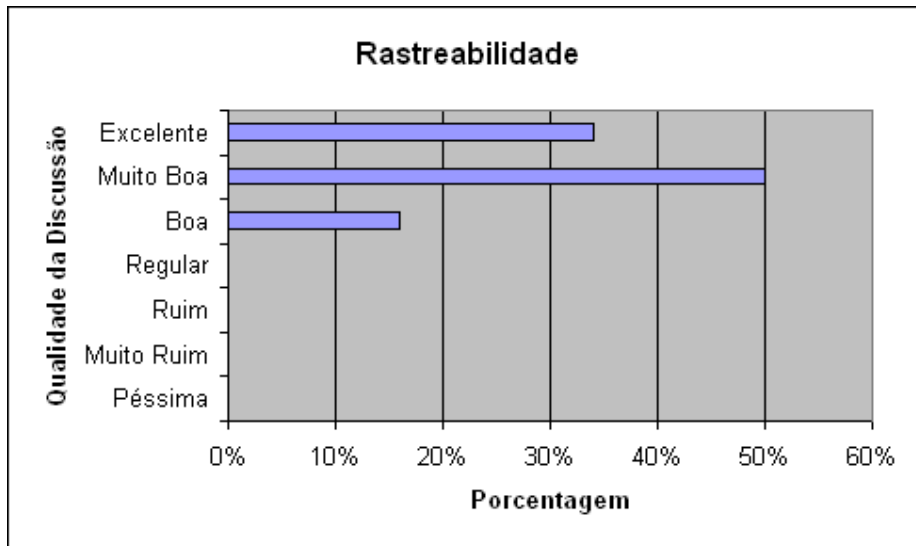


Figura 8.5 - Qualidade da discussão sobre Rastreabilidade

O histograma da Figura 8.6 apresenta assimetria positiva para a frequência das categorias assinaladas para a discussão sobre Amigabilidade. Os resultados foram: 17% Excelente, 83% Muito Boa. O histograma apresenta a segunda

mediana mais alta entre todos os histogramas analisados. A mediana pertence à categoria Muito Boa, um pouco acima do centro da barra.

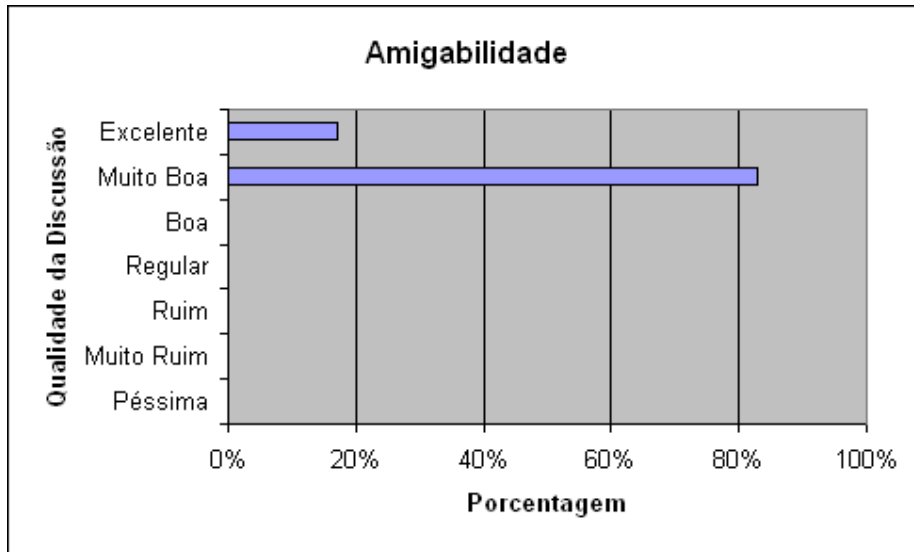


Figura 8.6 - Qualidade da discussão sobre Amigabilidade

A discussão sobre a Atualidade foi menos positiva do que a discussão sobre Amigabilidade, mas mesmo assim mais positiva do que as discussões sobre Disponibilidade e Portabilidade, por exemplo. Os resultados mostrados na Figura 8.7 foram: 75% Muito Boa, 25% Boa. O histograma apresenta assimetria negativa e a mediana também pertence à categoria Muito Boa, embora abaixo do centro da barra.

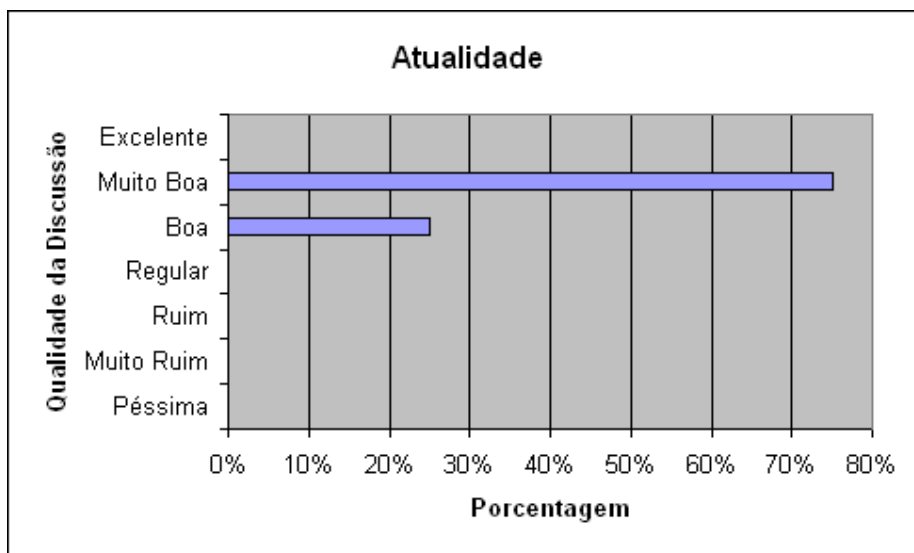


Figura 8.7 - Qualidade da discussão sobre Atualidade

A impressão dos participantes sobre a qualidade da discussão do RNF Concisão foi a mais baixa de todos os histogramas analisados. A Figura 8.8

apresenta o respectivo histograma. Os resultados foram: 25% Muito Boa, 75% Boa. A mediana pertence à categoria Boa, embora o histograma possua assimetria positiva. Assim, a mediana está um pouco acima do centro da barra.

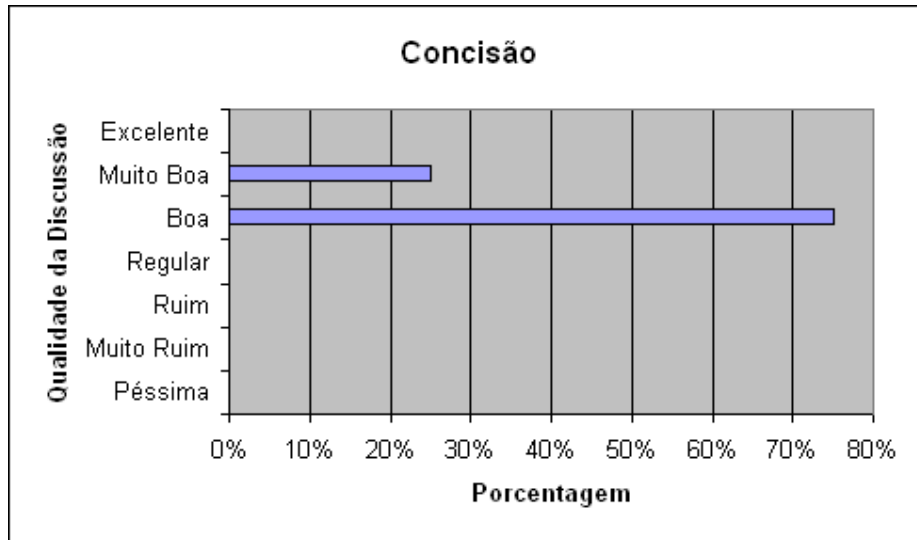


Figura 8.8 - Qualidade da discussão sobre Concisão

Observando-se todos os histogramas, percebe-se que os dados obtidos nesse experimento são estatisticamente normais e controlados (Travassos et al. 2001), ou seja, as barras de cada histograma seguem uma curva normal e o processo está sob controle, o que é positivo. Não foi necessário o descarte de *overlays*, ou pontos fora do gráfico.

Comparando as análises das discussões, percebe-se que a mediana pertence à categoria Muito Boa em praticamente todas as discussões, com exceção da mediana da reunião que discutiu o requisito não-funcional Concisão. A mediana do histograma geral pertence à categoria Muito Boa e o histograma possui assimetria negativa, ou seja, a mediana está um pouco abaixo do centro da barra.

De um modo geral, a impressão dos participantes foi muito positiva, pois se percebe o deslocamento de todas as barras dos histogramas para a parte superior do gráfico. De trinta e cinco questionários preenchidos, em nenhum caso um participante assinalou a categoria Ruim, por exemplo. Em somente dois casos (menos de seis por cento das vezes) os participantes assinalaram a categoria Regular.

A Seção 8.2 faz uma avaliação qualitativa das discussões com base nas justificativas fornecidas pelos participantes.



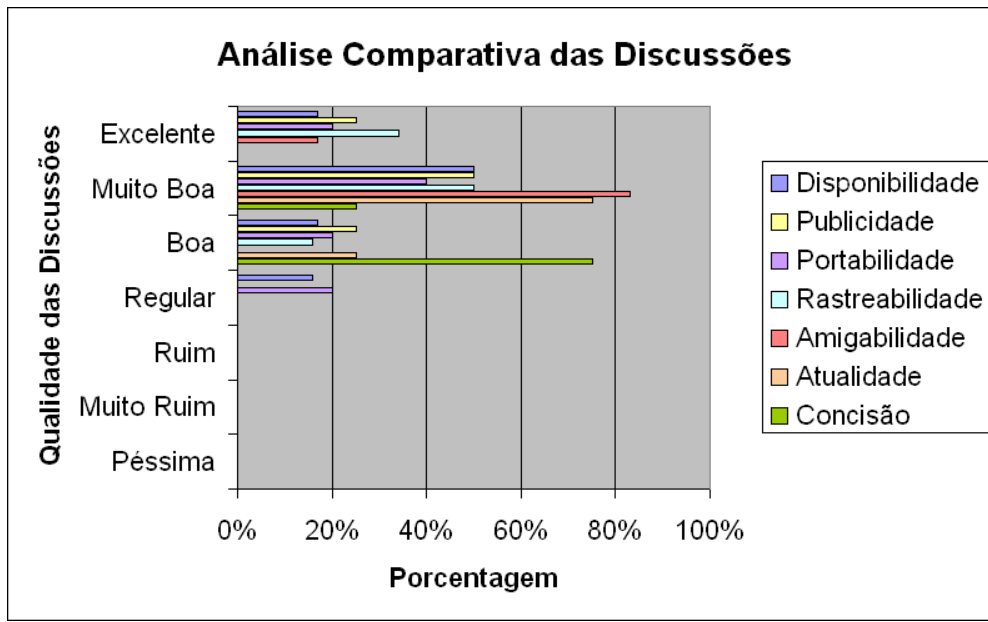


Figura 8.9 - Comparação das qualidades das discussões

## 8.2.

### Avaliação Qualitativa da Aplicação da Abordagem

Esta seção apresenta uma avaliação qualitativa da aplicação da abordagem, realizada, principalmente, assistindo-se às gravações em vídeo das reuniões, e observando-se o *feedback* dos participantes através dos questionários e dos histogramas apresentados na Seção 8.1. Algumas justificativas apresentadas pelos participantes para as escolhas entre as categorias de qualidade da discussão foram utilizadas como evidências.

Todos os comentários dos participantes foram transcritos exatamente com as mesmas palavras com que foram preenchidos. Dispensamos, assim, o uso da expressão *sic* em alguns comentários.

A discussão sobre Disponibilidade foi o primeiro experimento utilizando argumentação em uma discussão aberta. O mediador concentrou-se em não interferir ou pautar a discussão. Percebe-se no vídeo que, a partir de uma dúvida de um dos participantes, relacionada às diferenças entre Disponibilidade e Operabilidade, o foco da discussão passou a ser os padrões de requisitos obtidos do catálogo e não os requisitos de transparência para o Lattes-Scholar. Essa situação perdurou por aproximadamente quarenta minutos. Alguns participantes entendiam que evoluir o catálogo era pertinente, pois fazia parte dos objetivos do

Grupo de Transparência de Software. Segundo um dos participantes, “essa discussão trouxe à tona uma importante discussão sobre a relação Operabilidade/Disponibilidade”.

Entretanto, outros participantes gostariam de ter discutido apenas os requisitos de transparência para o Lattes-Scholar. O participante que classificou a discussão como Regular fazia parte desse grupo. Um dos participantes comentou: “A Disponibilidade levantou uma boa discussão no grupo, criando a necessidade de maior detalhamento em relação à sua definição. No entanto, esta discussão atrasou o trabalho principal da reunião que era discutir os requisitos”.

Devido ao *feedback* dos participantes, o papel do mediador foi alterado. O mediador passou a ter a responsabilidade de permitir breves discussões (em torno de dez minutos) sobre os padrões de requisitos extraídos do CTS, e retomar, posteriormente, o foco da reunião, ou seja, a discussão sobre os requisitos de transparência do Lattes-Scholar.

O histograma da discussão sobre Publicidade (Figura 8.3), na segunda reunião, já apresenta uma melhor avaliação dos participantes. Um dos participantes observou: “A discussão teve um fluxo mais rápido, cada um teve sua oportunidade para falar, [a discussão] foi melhor dirigida”. Durante a discussão, várias possíveis operacionalizações para a Publicidade foram identificadas, uma vez que o foco da discussão permaneceu nos requisitos de transparência. Outro participante conclui: “Vários itens foram identificados e [a versão anterior do] Lattes-Scholar não implementa nenhum.”

O RNF Portabilidade também foi discutido na segunda reunião, porém a discussão recebeu uma avaliação mais baixa. Assistindo às gravações em vídeo, percebeu-se a dificuldade dos participantes em apresentar idéias para a operacionalização da Portabilidade. Um dos participantes justificou sua avaliação: “Ainda temos poucas formas/guias para operacionalização [da Portabilidade]. Precisamos discutir/encontrar mais operacionalizações possíveis.” É precisamente nesse tipo de situação que o apoio do CTS torna-se fundamental. Entretanto, o CTS ainda está em desenvolvimento. Embora padrões dos tipos Objetivo e Questão estejam disponíveis, o CTS ainda não possui padrões do tipo Alternativa. Esse tipo de padrão é utilizado para sugerir possíveis operacionalizações.

Esse tipo de dificuldade não foi observado na terceira reunião, que discutiu a Rastreabilidade dos artefatos do Lattes-Scholar. Nossa abordagem sugere várias

operacionalizações para a Rastreabilidade, como heurísticas transformacionais, rastros do tipo origem-destino, padronização de nomes e modelos de pré-rastreabilidade, entre outros. Os participantes aprovaram as operacionalizações utilizadas e sugeriram algumas novas operacionalizações. Como os participantes puderam ver as operacionalizações sendo aplicadas na prática, a discussão sobre Rastreabilidade recebeu a melhor avaliação de todas as discussões. Um dos participantes argumentou: “O nome do pesquisador [uma das informações utilizadas no processo] veio do Consumidor [um ator] pela ligação de dependência [no modelo i\*].” Outro participante completou: “[a origem do recurso] Está explícito pela ligação de dependência e pela representação [de] Recurso da linguagem [i\*].”

A quarta reunião, que discutiu a Amigabilidade da interface do Lattes-Scholar, recebeu a segunda melhor avaliação de todas as discussões. A boa avaliação dos participantes deveu-se, em parte, ao uso de protótipos de interfaces. Um dos participantes afirmou: “Discutimos todas as perguntas [do padrão Questões de] Amigabilidade de forma detalhada. A crítica em cima do protótipo facilitou a discussão”. Outro participante disse que o uso de protótipos de interfaces “possibilitou confrontar os atributos e as suas operacionalizações com as interfaces de usuário de um sistema real. Isto permitiu corrigir alguns problemas e entender melhor o NFR [RNF Amigabilidade].” A discussão sobre Amigabilidade permitiu a evolução da interface do Lattes-Scholar (Apêndice A).

A avaliação das terceira e quarta reuniões sugerem que os participantes possuem uma melhor impressão das discussões que se baseiam em artefatos reais. Dessa forma, eles podem observar na prática os impactos das operacionalizações que estão discutindo.

Assistindo as gravações em vídeo da quinta reunião, nota-se que a discussão sobre Atualidade transcorreu normalmente. Todas as questões do Padrão Questões de Atualidade foram respondidas sem maiores dificuldades, ou seja, atingiram-se todos os objetivos dessa reunião. Por isso, estranha-se a avaliação Boa ou Muito Boa dos interessados. Um dos participantes afirmou: “Sob a ótica do software há que implantar uma política de versões e de configuração. Sob a ótica do serviço, podemos qualificar de Muito Bom, tendo em vista a arquitetura que depende de serviços externos, sem ter o problema de atualização, já que não dispõe de repositório”. Outro participante respondeu: “Conseguimos identificar uma

diferença entre softwares desktop e Web. Os mecanismos de atualização são diferentes, mas as informações sobre atualização são as mesmas.”.

Atribui-se a avaliação modesta da quinta reunião a não-utilização de protótipos ou artefatos reais do Lattes-Scholar, ou seja, nessa reunião retornou-se ao modelo de discussão das primeiras reuniões. A pequena queda na avaliação da qualidade da reunião sugere que os participantes preferem o modelo utilizado na quarta e na quinta reuniões.

Na sexta reunião, que discutiu o RNF Concisão, voltamos a observar a dificuldade dos participantes em sugerir idéias para a operacionalização do RNF em discussão. Nas gravações em vídeo, percebem-se situações em que os participantes ficam em dúvida, o que diminui o ritmo da reunião. Essas dificuldades podem justificar a avaliação dessa discussão, sendo a mais baixa de todas as reuniões. Segundo um dos participantes, discutir “foi bom, porque se identificou a necessidade de criar padrões para alcançar Concisão.” Outro participante afirmou: “Boa, mas precisamos discutir mais sobre operacionalização de Concisão.” Novamente, a inclusão de padrões do tipo Alternativa no CTS poderia ajudar nesse tipo de situação.

Em três casos os participantes optaram por não preencher os questionários. Pessoalmente, dois dos participantes afirmaram que optaram por não preencher, pois haviam perdido parte da discussão. Os vídeos das reuniões confirmaram as afirmações. O terceiro caso foi relacionado a um professor visitante que optou por não preencher, pois não havia participado das reuniões anteriores.

### **8.3. Avaliação da Abordagem Integrada por Construção**

Esta seção apresenta uma avaliação da abordagem integrada por construção. Em linhas gerais, expõe-se a transparência do software produzido pela abordagem integrada através da identificação explícita dos requisitos de transparência no código.

A Figura 8.10 apresenta um grafo de argumentos na linguagem ACE que representa alguns argumentos dos participantes durante a discussão do RNF Rastreabilidade, na terceira reunião. A discussão durou aproximadamente cinco minutos e resultou em um consenso: o Lattes-Scholar deveria manter um *log* do

processo realizado durante a sessão. Esse *log* deve ser apagado ao final da sessão, pois o Lattes-Scholar não deve replicar informações do Google Scholar. Todo esse processo de manter/apagar *logs* é uma operacionalização para o RNF Transparência e, portanto, é considerado um novo requisito de transparência para o Lattes-Scholar.

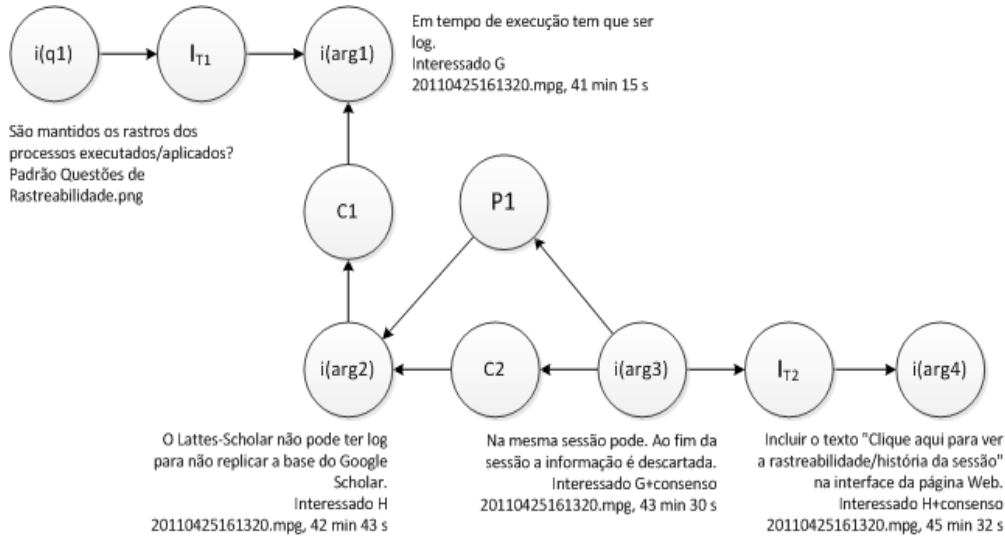


Figura 8.10 – Gráfico de argumentos de uma discussão sobre Rastreabilidade

O novo requisito de transparência é representado em um modelo SD para a situação “Manter Rastro/História do Processo”, apresentado na Figura 8.11. O *log* rastreia as informações fornecidas, como o nome do pesquisador, e em qual passo do processo a solicitação do Consumidor encontra-se. O Consumidor pode, a qualquer momento, selecionar na interface do Lattes-Scholar que deseja ver o rastro/história do processo. O Consumidor espera uma boa apresentação desse rastro. O SMA do Lattes-Scholar é o responsável por manter o rastro/história do processo. Para isso, o SMA rastreia as informações já fornecidas pelo “Consumidor”, como o nome do pesquisador, a URL do currículo escolhido e as sessões do currículo selecionadas. O rastro/história do processo é fornecido pelo SMA quando solicitado. A página *Web* do Lattes-Scholar depende do SMA para manter o *log*, através da meta “Rastro/História do processo seja mantido”. O SMA, por sua vez, depende da página *Web* para informar o rastro/história do processo ao Consumidor, através da meta “Rastro/História do processo seja informado”.

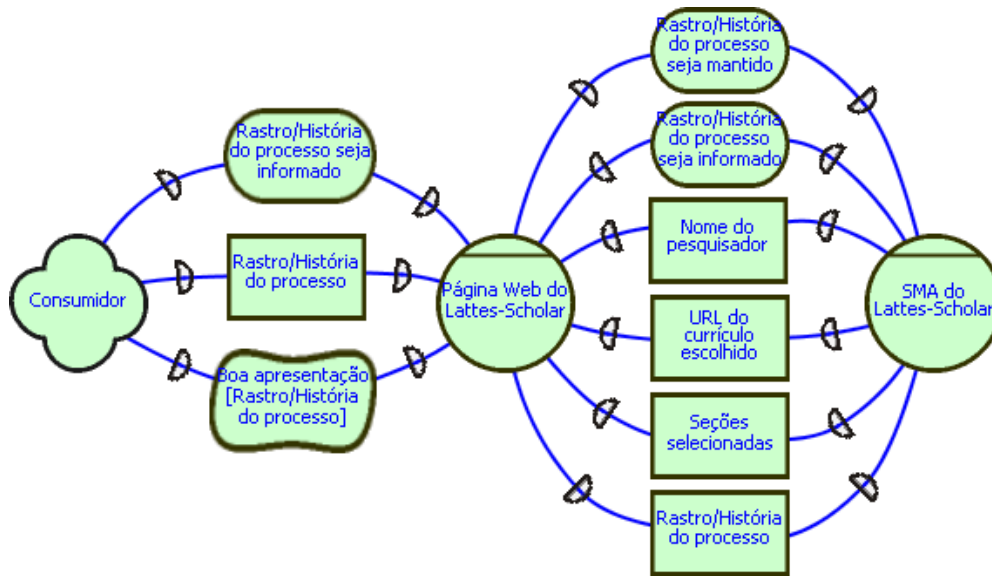


Figura 8.11 - Modelo SD para a situação “Manter Rastro/História do Processo”

O modelo SD de requisitos da Figura 8.11 serviu de base para o modelo SR arquitetural da Figura 8.12. No modelo SR arquitetural, o agente “Rastreador Historiador” assume a responsabilidade de atingir a meta “Rastro/História do processo seja mantido”. O agente possui duas formas de atingir a meta: a tarefa “Monitorar o processo” ou a tarefa “Registrar apenas o passo do processo”.

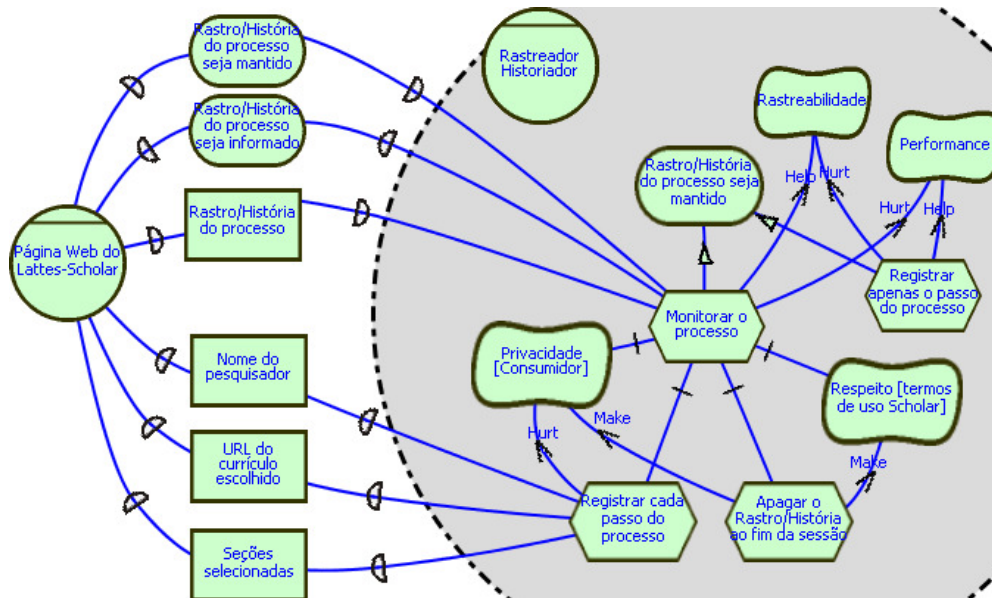


Figura 8.12 - Modelo SR para a situação “Manter Rastro/História do Processo”

Entretanto, a escolha da tarefa para atingir a meta deve levar em conta os critérios de qualidade, ou metas flexíveis, impactadas por essa escolha. Se, por um lado, o agente escolher a tarefa “Monitorar o processo”, estará impactando positivamente a meta flexível “Rastreabilidade” e, negativamente, a meta flexível

“Performance”. Por outro lado, se escolher a tarefa “Registrar apenas o passo do processo”, estará impactando negativamente a meta flexível “Rastreabilidade” e, positivamente, a meta flexível “Performance”, uma vez que registrar apenas o passo do processo é mais simples.

A tarefa “Monitorar o processo” é decomposta em duas tarefas e duas metas flexíveis. A execução das sub-tarefas impacta as metas flexíveis “Privacidade [Consumidor]” e “Respeito [termos de uso do Scholar]”. A tarefa “Monitorar o processo” também depende do agente “Página Web do Lattes-Scholar” para atingir a meta “Rastro/História do processo seja informado”. O elo de dependência possui a mesma semântica de uma decomposição da tarefa em uma sub-meta.

Para a demonstração de que o requisito de transparência está presente no código de forma explícita, o agente, as metas, as metas flexíveis, as tarefas, as contribuições e os recursos dos modelos SD e SR serão mostrados diretamente no código do agente “RastreadorHistoriador”. A aplicação das heurísticas transformacionais e a explicação dos rastros como pares origem-destino não serão explanadas para evitar a redundância, pois ambos os Capítulos 3 e 7 já o fizeram.

O cabeçalho do ADF do agente “RastreadorHistoriador” é apresentado na Figura 8.13. O agente é implementado como um agente executável JADEX.

```
<agent xmlns="http://jadex.sourceforge.net/jadex"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://jadex.sourceforge.net/jadex
    http://jadex.sourceforge.net/jadex-0.96.xsd"
  name="RastreadorHistoriador"
  package="lattescholar.sma.rastreadorhistoriador">

  <imports>
    <import>jadex.util.*</import>
    <import>jadex.adapter.fipa.*</import>
    <import>istar.metamodel.MetaPlan</import>
  </imports>

  <capabilities>
    <capability name="ReasoningEngine"
      file="istar.metamodel.PropagationSimulator"/>
  </capabilities>
```

Figura 8.13 - Cabeçalho do ADF do agente RastreadorHistoriador

As metas “rastros\_historia\_do\_processo\_seja\_mantido” e “rastros\_historia\_do\_processo\_seja\_informado” estão presentes explicitamente no ADF do agente.

A Figura 8.14 apresenta o trecho em XML do ADF que contém a declaração das metas.

O trecho em XML do ADF que contém a declaração explícita das tarefas “monitorar\_o\_processo” e “registrar\_apenas\_o\_passo\_do\_processo” é apresentado na Figura 8.15.

```
<goals>
  <maintaingoal name="rastros_historia_do_processo_seja_mantido"/>
  <achievegoal name="rastros_historia_do_processo_seja_informado"/>
  <metagoal name="metagoal1" recalculate="false">
    <parameterset name="applicables" class="ICandidateInfo"/>
    <parameterset name="result" class="ICandidateInfo" direction="out"/>
    <trigger>
      <goal ref="rastros_historia_do_processo_seja_mantido"/>
    </trigger>
  </metagoal>
</goals>
```

Figura 8.14 - Declarações das metas no ADF

```
<plans>
  <plan name="monitorar_o_processo">
    <body>new MonitorarOProcesso()</body>
    <trigger>
      <goal ref="rastros_historia_do_processo_seja_informado"/>
    </trigger>
  </plan>

  <plan name="registrar_apenas_o_passo_do_processo">
    <body>new RegistrarApenasOPassoDoProcesso()</body>
    <trigger>
      <goal ref="rastros_historia_do_processo_seja_informado"/>
    </trigger>
  </plan>
  ...
</plans>
```

Figura 8.15 - Declarações das tarefas/planos no ADF

As tarefas “Registrar cada passo do processo” e “Apagar o Rastro/História ao fim da sessão” são sub-tarefas da tarefa “Monitorar o processo” e, portanto, são implementadas como métodos a serem utilizados pelo plano “MonitorarOProcesso”. A Figura 8.16 apresenta um trecho do arquivo Java da classe “MonitorarOProcesso” que estende a classe “Plan” do JADEX.



```

public class MonitorarOProcesso extends Plan {
    + public void registrarCadaPassoDoProcesso () {...}
    + public void apagarORastroHistoriaAoFimDaSessão () {...}
    @Override
    + public void body () {...}
}

```

Figura 8.16 - As sub-tarefas implementadas como métodos

Os recursos “nome do pesquisador”, “URL do currículo escolhido” e “seções selecionadas” são representados como crenças do agente “RastreadorHistoriador”. O recurso “Rastro/História do processo” é composto pelos três recursos citados mais o recurso “Passo do processo”. A Figura 8.17 apresenta o trecho em XML do ADF que contém a declaração das crenças. Além disso, mostra os três conjuntos de crenças “softgoals”, “contributions” e “tasks”.

```

<beliefs>
  <belief name="passo_do_processo" class="Integer"/>
  <belief name="nome_do_pesquisador" class="String"/>
  <belief name="url_do_curriculo_escolhido" class="String"/>
  <beliefset name="secoes_selecionadas" class="String"/>

  <beliefset name="tasks" class="Task"/>
  <beliefset name="contributions" class="Contribution"/>
  <beliefset name="softgoals" class="Softgoal"/>
</beliefs>

```

Figura 8.17 - O recursos declarados como crenças do agente

As quatro metas flexíveis do modelo arquitetural SR são representadas como variáveis nebulosas e inseridas no conjunto de crenças “softgoals”. As contribuições das tarefas para as metas flexíveis são representadas como crenças inseridas no conjunto de crenças “contributions”. As tarefas também são representadas como crenças no conjunto de crenças “tasks”. As Figuras 8.18, 8.19 e 8.20 apresentam trechos em Java do método “body()” do plano “Setup”, que faz a inicialização do agente. Nesse plano, objetos das classes “Softgoal”, “Contribution” e “Task” para as metas flexíveis, contribuições e tarefas do modelo arquitetural SR são instanciados e inseridos nos conjuntos de crenças.

```

LeafSoftgoal rastreabilidade = new LeafSoftgoal("Rastreabilidade", 1.0);
this.getBeliefbase().getBeliefSet("softgoals").addFact(rastreabilidade);

LeafSoftgoal performance = new LeafSoftgoal("Performance", 1.0);
this.getBeliefbase().getBeliefSet("softgoals").addFact(performance);

LeafSoftgoal privacidadeConsumidor = new LeafSoftgoal("Privacidade Consumidor", 1.0);
this.getBeliefbase().getBeliefSet("softgoals").addFact(privacidadeConsumidor);

LeafSoftgoal respeitoTermosDeUsoScholar =
    new LeafSoftgoal("Respeito Termos De Uso Scholar", 1.0);
this.getBeliefbase().getBeliefSet("softgoals").addFact(respeitoTermosDeUsoScholar);

```

Figura 8.18 – As metas flexíveis inseridas no conjunto de crenças “softgoals”

```

Task monitorarOProcesso = new Task("MonitorarOProcesso");
Contribution[] contr1 = {
    new Contribution("MonitorarOProcesso",
        ContributionType.HELP, "Rastreabilidade"),
    new Contribution("MonitorarOProcesso",
        ContributionType.HURT, "Performance")};
monitorarOProcesso.setSoftgoalContributions(contr1);
this.getBeliefbase().getBeliefSet("tasks").addFact(monitorarOProcesso);

Task registrarApenasOPassoDoProcesso =
    new Task("RegistrarApenasOPassoDoProcesso");
Contribution[] contr2 = {
    new Contribution("RegistrarApenasOPassoDoProcesso",
        ContributionType.HURT, "Rastreabilidade"),
    new Contribution("RegistrarApenasOPassoDoProcesso",
        ContributionType.HELP, "Performance")};
registrarApenasOPassoDoProcesso.setSoftgoalContributions(contr2);
this.getBeliefbase().getBeliefSet("tasks").
    addFact(registrarApenasOPassoDoProcesso);

```

Figura 8.19 – As tarefas e as suas contribuições para as metas flexíveis

```

Task registrarCadaPassoDoProcesso =
    new Task("RegistrarCadaPassoDoProcesso");
Contribution[] contr3 = {
    new Contribution("RegistrarCadaPassoDoProcesso",
        ContributionType.HURT, "Privacidade Consumidor")};
registrarCadaPassoDoProcesso.setSoftgoalContributions(contr3);
this.getBeliefbase().getBeliefSet("tasks").
    addFact(registrarCadaPassoDoProcesso);

Task apagarORastroHistoriaAoFimDaSessão =
    new Task("ApagarORastroHistoriaAoFimDaSessão");
Contribution[] contr4 = {
    new Contribution("ApagarORastroHistoriaAoFimDaSessão",
        ContributionType.MAKE, "Privacidade Consumidor"),
    new Contribution("ApagarORastroHistoriaAoFimDaSessão",
        ContributionType.MAKE, "Respeito Termos De Uso Scholar")};
apagarORastroHistoriaAoFimDaSessão.setSoftgoalContributions(contr4);
this.getBeliefbase().getBeliefSet("tasks").
    addFact(apagarORastroHistoriaAoFimDaSessão);

```

Figura 8.20 – As sub-tarefas e suas contribuições para as metas flexíveis

É importante ressaltar que todos os trechos do ADF e do plano do agente apresentados são fundamentais para o funcionamento do sistema. As metas flexíveis inseridas no conjunto de crenças “softgoals” (Figura 8.18) são utilizadas em tempo de execução pela máquina de raciocínio qualitativa.

#### 8.4. Análise das Versões Anteriores do Lattes-Scholar

Esta seção analisa a transparência de duas versões anteriores do Lattes-Scholar: a segunda versão, implementada na linguagem Lua e a terceira versão, implementada como um SMA comportamental na plataforma JADE.

A segunda versão do Lattes-Scholar foi implementada na linguagem Lua, uma linguagem *script*. O sistema é composto por páginas *Web* modularizadas através da arquitetura *Model-View-Control*. Os requisitos foram modelados no framework *i\**. Portanto, estão descritos por atores, metas, metas flexíveis, tarefas, recursos e dependências estratégicas entre os atores. Como a implementação não segue o paradigma intencional, o código produzido não utiliza as metas e as metas flexíveis, por exemplo.

Foi realizada, nessa segunda versão do Lattes-Scholar, uma tentativa de se anexar os requisitos ao código, através da inserção de comentários. As tarefas do modelo *i\** foram especificadas em cenários e inseridas manualmente no código, como pode ser observado na Figura 8.21. A manutenção desses cenários sofre do POTLO BOAD TIP – *Problem of the Lack of Benefit of a Document to its Produces*. Notam-se alguns problemas, como, por exemplo, no campo “Recursos”. O recurso “status” é utilizado pelo código, mas não consta no cenário.

```
--[[
@Titulo: Obter página
@Objetivo: Obter o código fonte da página web correspondente ao currículo do pesquisador.
@Contexto: lattes.lua
@Recursos: url da página.
@Atores: função requisição.
--]]

function obterPagina (urlCurrículo)
  --@Episódio 1: Requisição obtém código fonte do urlCurrículo e o status da requisição .
  local pagina, status = requisicao(urlCurrículo)
  --@Excessão 1:Caso a urlCurrículo seja invalida, não será retornado o código fonte da página.
  if not pagina or status ~= 200 then
    return nil, status
  end
  return pagina, status
end
```

Figura 8.21 – O cenário “Obter página”

Percebe-se também que a forma de se especificar o cenário varia entre os desenvolvedores. As diferenças vão desde o uso de capitalização até a semântica dos campos. O cenário mostrado na Figura 8.21, por exemplo, considera como ator a função que chama o cenário. Já o cenário exposto na Figura 8.22 considera o sistema e a camada de controle como os atores.

```
--[[
@Titulo: OBTENOME DO PESQUISADOR
@Objetivo: Obter nome do pesquisador autor dos tópicos relacionados.
@Contexto: CONTROLE EXIBIR TÓPICOS INTERESSANTES
@Recursos: lattes.lua, auxiliar.lua
@Atores: Sistema, Camada de Controle.
]]--
function obterNomePesquisador(curriculo)
--@Episódio 1: chamar módulo lattes para obter nome do pesquisador.
    local nome = lattes.obterNomeCur(curriculo)

--[[
@Episódio 2: Retornar o nome.
@Exceção: Nome vazio retorna string vazia "" .
]]--
    if not nome or nome == "" then
        return "", auxiliar.formataStringParaHTML("Nome não encontrado!")
    else
        return nome
    end
end
end
```

Figura 8.22 – O cenário “Obter nome do pesquisador”

Outro problema é que as metas já descrevem **o que** fazer, e não **o porquê**, tornando o nome da meta mais longo e mais específico do que o próprio título do cenário que descreve a tarefa. O cenário “Obter endereço”, mostrado na Figura 8.23, trata do retorno da função no campo relacionado à meta. O campo “Recurso” desse cenário não faz referência ao recurso “endereço”, o principal recurso utilizado.

```
--[[
@Titulo: OBTENOME DO ENDEREÇO
@Objetivo: Obter endereço e validá-lo. Retorna também possíveis mensagens de erro durante a validação.
@Contexto: Exibir lista de obras e número de citações.
@Atores: contagem.lp
@Recursos: auxiliar.lua , string .
]]--
function obterEndereco(endereco)
    if not endereco or not string.find(endereco, "^http://[%w%.]+") then
        return endereco, auxiliar.formataStringParaHTML("Endereço Inválido!")
    else
        return endereco, nil
    end
end
end
```

Figura 8.23 – O cenário “Obter endereço”

Em resumo, a segunda versão do Lattes-Scholar faz uma tentativa de anexar os requisitos ao código na forma de cenários estruturados, inseridos como comentários. Alguns problemas observados dificultam a localização dos requisitos

no código, como o próprio nome dos arquivos, que seguem o padrão *Model-View-Control*, e não as divisões por tarefas do modelo *i\**. Outros problemas mais difíceis de superar são as metas que descrevem **o que** fazer, e não **o porquê**, e a inexistência de metas flexíveis no código.

A terceira versão do Lattes-Scholar também foi modelada através de modelos intencionais do *framework i\**. Portanto, os requisitos da terceira versão também estão descritos por atores, metas, metas flexíveis, tarefas, recursos e dependências estratégicas entre os atores. O sistema foi implementado através de um SMA comportamental na plataforma JADE, o que trouxe alguns benefícios, tais como: os atores do modelo *i\** estão explícitos no código como agentes, e as tarefas estão explícitas no código como comportamentos.

Porém, o paradigma comportamental assume que as metas e as metas flexíveis estão implícitas, ou seja, a inteligência do agente restringe-se à existência, ao número e à interdependência entre os planos. Este paradigma é muito útil para, por exemplo, modelar sistemas biológicos. Assim, é possível replicar razoavelmente a inteligência de insetos sociais, por exemplo, apenas replicando seus comportamentos.

De modo a inserir explicitamente as metas no código dos agentes, a terceira versão do Lattes-Scholar utiliza uma infra-estrutura para a execução de comportamentos. Essa infra-estrutura exige que uma meta seja instanciada para que um comportamento seja executado. Assim, torna-se explícito no código que um comportamento visa atingir uma determinada meta.

Entretanto, a infra-estrutura utilizada não pode ser considerada uma máquina de raciocínio orientada à meta, como a fornecida pelo JADEX. A infra-estrutura não oferece, por exemplo, um sistema de deliberação de metas, o que impede o agente de raciocinar sobre várias metas em tempo de execução. Também não existe uma máquina de raciocínio meios-fim. Assim, o sistema implementado ainda configura um SMA comportamental.

A Figura 8.24 mostra um agente implementado na terceira versão do Lattes-Scholar. O agente é especificado em um documento XML que representa o ator do modelo *i\**. Como pode ser observado, o agente possui uma única meta. Além disso, só existe uma forma possível de se atingir a meta, ou seja, executar a tarefa/comportamento 3.5. Todos os agentes implementados possuem essas limitações.

```

<actor id="3" type="role" name="manager">
  <boundary>
    <ielement id="3.1" type="task" name="ReceberUrlPesquisador" basic="true" />
    <ielement id="3.2" type="task" name="SolicitarObrasCurriculo">
      <ielementLink type="decomposition" run-mode="sequential">
        <ielement iref="4" />
      </ielementLink>
    </ielement>
    <ielement id="3.3" type="task" name="SolicitarCitacoesObras">
      <ielementLink type="decomposition" run-mode="sequential">
        <ielement iref="5" />
      </ielementLink>
    </ielement>
    <ielement id="3.4" type="task" name="CalcularHIndex" basic="true" />
    <ielement id="3.5" type="task" name="ObterCitacoesPesquisador">
      <ielementLink type="decomposition" run-mode="sequential">
        <ielement iref="3.1" />
        <ielement iref="3.2" />
        <ielement iref="3.3" />
        <ielement iref="3.4" />
      </ielementLink>
    </ielement>
    <ielement id="3.6" type="goal" name="CitacoesObrasObtidas" main="true">
      <ielementLink type="means-end" run-mode="unique">
        <ielement iref="3.5" />
      </ielementLink>
    </ielement>
  </boundary>
</actor>

```

Figura 8.24 – O agente “manager”

As limitações observadas devem-se à ausência das máquinas de deliberação de metas e de raciocínio meios-fim. Dessa forma, o agente possui uma única meta, utilizada pela infra-estrutura para executar a única tarefa/comportamento. A maior desvantagem dessa abordagem é a perda de variabilidade, ou seja, o agente não possui diversas metas nem diversos meios de se atingir cada uma dessas metas.

As metas flexíveis não estão presentes no código, uma vez que o agente não faz seleções de tarefas em tempo de execução. Dessa forma, as metas flexíveis, que representam critérios de qualidade, são desnecessárias.

Finalmente, uma importante contribuição para a transparência da nossa versão, em comparação com as duas versões anteriores, é a existência de modelos de pré-rastreabilidade ITrace para os artefatos. Esses modelos permitem rastrear um dado artefato de volta para os processos da Engenharia de Software que o produziram. Esse recurso permite, dentre outras contribuições: (i) tratar as relações entre indivíduos ou artefatos e as fontes de informações e entre indivíduos ou artefatos e os recursos como "cidadãos de primeira classe"; (ii)

identificar quais atores foram responsáveis por manipular um artefato, e durante quais interações sociais essas manipulações sucederam-se; e (iii) dedicar especial atenção para com as razões (i.e. a dimensão “porquê” da informação) por detrás das contribuições de cada interação envolvendo, por exemplo, a rede social, os eventos e as fontes de informação.

## 8.5. Considerações Finais

Este capítulo apresentou diferentes avaliações da aplicação da abordagem e da própria abordagem integrada, proposta no Capítulo 7.

A aplicação da abordagem foi avaliada de forma quantitativa e qualitativa, através de questionários preenchidos pelos participantes das reuniões. Os participantes eram doutores, doutorandos e mestres que possuíam interesse no software e no conceito de transparência de software.

A avaliação quantitativa da aplicação da abordagem foi realizada através da construção e análise estatística de histogramas das frequências das categorias, utilizadas no questionário para classificar a qualidade da discussão.

A avaliação qualitativa baseou-se no *feedback* dos participantes ao justificarem as suas classificações quanto a qualidade da discussão, tendo em vista a análise dos histogramas das discussões e as gravações em vídeo das discussões.

A abordagem integrada foi avaliada por construção, demonstrando que os requisitos de transparência estão presentes explicitamente no código dos agentes. Mostramos no código do SMA intencional cada abstração que compõe os requisitos, tais como: os atores, as metas, as metas flexíveis, as tarefas e os recursos.

Também foi realizada uma breve análise da transparência de duas versões anteriores do Lattes-Scholar. Comparando essas versões com a nossa proposta, notamos que algumas limitações das mesmas – i.e. dificuldade dos agentes raciocinarem considerando duas ou mais alternativas para atingir uma determinada meta em tempo de execução bem como necessidade de lidar com a dimensão “porquê” da informação – foram superadas usando, por exemplo e respectivamente, SMA intencionais centrados no modelo BDI e modelos de pré-rastreabilidade ITrace.