



UMA ABORDAGEM SENSÍVEL AO CONTEXTO PARA VISUALIZAÇÃO DE  
INFORMAÇÃO: UM EXEMPLO EM CENÁRIOS DE DESENVOLVIMENTO DE  
SOFTWARE

Renan Ribeiro de Vasconcelos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadora: Cláudia Maria Lima Werner

Rio de Janeiro

Abril de 2015

UMA ABORDAGEM SENSÍVEL AO CONTEXTO PARA VISUALIZAÇÃO DE  
INFORMAÇÃO: UM EXEMPLO EM CENÁRIOS DE DESENVOLVIMENTO DE  
SOFTWARE

Renan Ribeiro de Vasconcelos

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)  
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS  
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM  
CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof<sup>a</sup>. Cláudia Maria Lima Werner, D.Sc.

---

Prof. Claudio Esperança, Ph.D

---

Prof. Renato Lima Novais, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

ABRIL DE 2015

Vasconcelos, Renan Ribeiro de

Uma Abordagem Sensível ao Contexto para Visualização de Informação: Um Exemplo em Cenários de Desenvolvimento de Software/ Renan Ribeiro de Vasconcelos. – Rio de Janeiro: UFRJ/COPPE, 2015.

XIII, 108 p.: il.; 29,7 cm.

Orientadora: Cláudia Maria Lima Werner

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2015.

Referências Bibliográficas: p. 92-97.

1. Visualização de Informação. 2. Sensibilidade ao Contexto. 3. Visualização de Software. I. Werner, Cláudia Maria Lima. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*A meus pais e a meus avós.*

## AGRADECIMENTOS

A Deus, especialmente pela saúde e paz em todos os momentos, e por todas as graças concedidas ao longo de minha vida, que me permitiram caminhar até aqui.

A meus pais, Maria da Graça e Admilton, que me criaram e me deram a educação mais importante. O esforço que sempre fizeram em todos os sentidos para me beneficiar não teria agradecimento suficiente expresso em palavras.

A meus avós, que, embora não mais presentes durante a realização deste trabalho, sempre me deram carinho e senso de responsabilidade desde minhas primeiras lembranças.

Aos meus irmãos e parentes, pelos momentos de apoio e compreensão. Aos meus animais de estimação, pela energia e alegria que me transmitiam.

À minha namorada, Verônica, companheira de todas as horas, pelo estímulo constante e alegria compartilhada a cada conquista. À sua família, pela simpatia e generosidade ao me acolherem sempre.

Aos meus amigos, de todas as fases da vida, que me permitiram vivenciar alegrias e compartilharam aprendizados e emoções.

À minha orientadora Cláudia Werner, pela condução, pelos conselhos e pelas oportunidades oferecidas, não apenas durante este trabalho, mas desde a graduação. Sem sua atenção, os resultados alcançados não seriam possíveis.

Ao Marcelo Schots, pela orientação informal, pela amizade e compreensão em diversos momentos. Seu olhar atento me ensinou a sempre buscar o melhor resultado.

Aos professores Claudio Esperança e Renato Novais, pela participação na banca e pela atenção dedicada ao trabalho.

Aos colegas que se disponibilizaram a participar da avaliação da abordagem, contribuindo para os resultados obtidos.

Ao professor Guilherme Travassos, pela atenção dedicada desde a Iniciação Científica e pelas oportunidades oferecidas, sempre acreditando em mim.

Aos amigos do Lab3D, do grupo de Reutilização da COPPE/UFRJ e da DAbM na Marinha do Brasil, que sempre estiveram disponíveis para colaborar e me incentivar.

À CAPES, pelo apoio financeiro durante o mestrado.

A todos aqueles que passaram em minha vida e contribuíram de alguma forma.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UMA ABORDAGEM SENSÍVEL AO CONTEXTO PARA VISUALIZAÇÃO DE  
INFORMAÇÃO: UM EXEMPLO EM CENÁRIOS DE DESENVOLVIMENTO DE  
SOFTWARE

Renan Ribeiro de Vasconcelos

Abril/2015

Orientadora: Cláudia Maria Lima Werner

Programa: Engenharia de Sistemas e Computação

O desenvolvimento de atividades em cenários dinâmicos e multivariados pode ser de difícil gerenciamento, pois os dados gerados a partir de tais atividades podem ser numerosos, de diferentes tipos e de difícil interpretação. Metodologias de suporte a ambientes dinâmicos envolvem a promoção da percepção (*awareness*), compartilhando informações e viabilizando análises mais eficientes dos dados resultantes das atividades desenvolvidas. A fim de definir quais conceitos devem ser considerados e monitorados, o conceito de gerenciamento de contexto pode ser adotado, destacando as informações mais relevantes, das quais os *stakeholders* envolvidos devem estar conscientes. Mecanismos de representação das situações de interesse devem ser planejados, de forma a tornar a interpretação dos valores uma tarefa inteligível.

Este trabalho apresenta a abordagem CAVE, que visa apoiar o uso apropriado de visualizações por meio de técnicas de sensibilidade ao contexto, promovendo a percepção em diferentes situações, com um foco inicial em cenários de desenvolvimento de software. A abordagem provê um mecanismo de suporte através de visualizações sensíveis ao contexto, que possibilitam análises de questões mais relevantes no momento atual de um dado cenário. Esta abordagem foi avaliada a partir da execução de um estudo de observação, cujos resultados fornecem indicadores positivos de sua aplicabilidade em cenários de desenvolvimento de software.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A CONTEXT-AWARE APPROACH FOR INFORMATION VISUALIZATION: AN  
EXAMPLE IN SOFTWARE DEVELOPMENT SCENARIOS

Renan Ribeiro de Vasconcelos

April/2015

Advisor: Cláudia Maria Lima Werner

Department: Computer and Systems Engineering

The development of activities in dynamic and multivariate scenarios may be difficult to manage, because the data generated from such activities can be big, with different types and hard to interpret. Support methodologies to dynamic environments may promote awareness, sharing information and making analysis more efficient. In order to define which concepts must be handled and monitored, the context management concept can be adopted, highlighting the most relevant data, those that the stakeholders should be aware of. Mechanisms of representation of the situations of interest must be planned, in a way to make the interpretation of values an intelligible task.

This work presents the CAVE approach that aims at supporting the appropriate use of visualizations together with context-awareness techniques, promoting awareness in different situations, with an initial focus on software development scenarios. The approach provides a support mechanism through context-aware visualizations, which helps the analysis of the most relevant issues at the current moment in a given scenario. This approach was evaluated through the execution of an observation study, which results show positive indicators for its applicability in software development scenarios.

# ÍNDICE

|  |    |
|--|----|
| CAPÍTULO 1 - INTRODUÇÃO.....                           | 1  |
| 1.1 Motivação .....                                    | 1  |
| 1.2 Hipótese e Objetivo .....                          | 2  |
| 1.3 Organização .....                                  | 4  |
| CAPÍTULO 2 - SENSIBILIDADE AO CONTEXTO.....            | 5  |
| 2.1 Introdução.....                                    | 5  |
| 2.2 Conceitos .....                                    | 5  |
| 2.2.1 Ubiquidade.....                                  | 6  |
| 2.2.2 Contexto.....                                    | 6  |
| 2.2.3 Computação Sensível ao Contexto .....            | 7  |
| 2.3 Modelo de Contexto .....                           | 8  |
| 2.4 Considerações Finais .....                         | 12 |
| CAPÍTULO 3 - VISUALIZAÇÃO DE INFORMAÇÃO .....          | 14 |
| 3.1 Introdução.....                                    | 14 |
| 3.2 Percepção.....                                     | 15 |
| 3.3 Visualização.....                                  | 16 |
| 3.3.1 Técnicas de Visualização.....                    | 18 |
| 3.3.2 Visualização de Software .....                   | 20 |
| 3.4 Elementos de Visualização.....                     | 21 |
| 3.5 Considerações Finais .....                         | 23 |
| CAPÍTULO 4 - VISUALIZAÇÕES SENSÍVEIS AO CONTEXTO ..... | 25 |
| 4.1 Introdução.....                                    | 25 |
| 4.2 Visão Geral da Abordagem .....                     | 25 |
| 4.2.1 Características de Visualização .....            | 27 |
| 4.2.1.1 Análise de Domínio .....                       | 27 |

|   |  |    |
|---|--|----|
| 4.2.1.2                                       | Categorias de Características.....       | 28 |
| 4.2.2   | Contexto de Visualização .....           | 31 |
| 4.2.2.1                                       | Modelo de Contexto .....                 | 32 |
| 4.2.2.2                                       | Situações de Contexto .....              | 37 |
| 4.2.2.3                                       | Regras de Contexto.....                  | 40 |
| 4.2.3   | Visualização Orientada ao Contexto ..... | 42 |
| 4.3   | Análise Comparativa .....                | 43 |
| 4.4   | Considerações Finais .....               | 45 |
| CAPÍTULO 5 – IMPLEMENTAÇÃO DA ABORDAGEM ..... |  | 46 |
| 5.1   | Introdução.....                          | 46 |
| 5.2   | Arquitetura.....                         | 46 |
| 5.3   | Obtenção dos Dados .....                 | 47 |
| 5.4   | Monitoramento do Contexto.....           | 48 |
| 5.5   | Composição de Visualizações .....        | 53 |
| 5.6   | Exemplos de Utilização .....             | 55 |
| 5.6.1   | Administrador.....                       | 57 |
| 5.6.2   | Usuário.....                             | 60 |
| 5.7   | Considerações Finais .....               | 63 |
| CAPÍTULO 6 - ESTUDO DE OBSERVAÇÃO.....        |  | 64 |
| 6.1   | Introdução.....                          | 64 |
| 6.2   | Objetivo .....                           | 64 |
| 6.3   | Hipótese .....                           | 64 |
| 6.4   | Definição .....                          | 64 |
| 6.5   | Cenários.....                            | 65 |
| 6.6   | Procedimentos .....                      | 66 |
| 6.7   | Resultados.....                          | 68 |
| 6.7.1   | Caracterização dos Participantes .....   | 68 |

|   |     |
|---|-----|
| 6.7.2 Cenário A.....                                | 71  |
| 6.7.3 Cenário B.....                                | 74  |
| 6.7.4 Cenário C.....                                | 76  |
| 6.7.5 Cenário D.....                                | 80  |
| 6.7.6 Vantagens .....                               | 84  |
| 6.7.7 Dificuldades.....                             | 84  |
| 6.8 Validade.....                                   | 85  |
| 6.9 Considerações Finais .....                      | 86  |
| CAPÍTULO 7 - CONCLUSÃO.....                         | 89  |
| 7.1 Epílogo.....                                    | 89  |
| 7.2 Contribuições.....                              | 89  |
| 7.3 Limitações .....                                | 90  |
| 7.4 Perspectivas Futuras .....                      | 91  |
| REFERÊNCIAS BIBLIOGRÁFICAS .....                    | 92  |
| APÊNDICE A - FORMULÁRIO DE CONSENTIMENTO .....      | 98  |
| APÊNDICE B - QUESTIONÁRIO DE CARACTERIZAÇÃO.....    | 99  |
| APÊNDICE C - TAREFAS.....                           | 100 |
| Introdução.....                                     | 100 |
| Cenário A.....                                      | 101 |
| Cenário B.....                                      | 101 |
| Cenário C.....                                      | 101 |
| Cenário D.....                                      | 101 |
| APÊNDICE D - QUESTIONÁRIO DE OBSERVAÇÃO .....       | 103 |
| APÊNDICE E - QUESTIONÁRIO DE OBSERVAÇÃO GERAL ..... | 104 |
| APÊNDICE F - ROTEIRO DAS TAREFAS.....               | 105 |
| APÊNDICE G - GABARITO DAS TAREFAS.....              | 108 |

# ÍNDICE DE FIGURAS

|   |    |
|---|----|
| Figura 2.1 - Exemplo de modelo de características com a notação Odyssey-FEX.....                      | 10 |
| Figura 2.2 – Exemplo de modelo de características de contexto.....                                    | 10 |
| Figura 2.3 - Formação de uma expressão de definição de contexto (FERNANDES <i>et al.</i> , 2011)..... | 11 |
| Figura 2.4 - Exemplo de definição de contexto.....  | 11 |
| Figura 2.5 – Exemplo de regra de contexto.....  | 12 |
| Figura 3.1 - Estágios de visualização (WARE, 2004).....   | 16 |
| Figura 4.1 - Visão geral da abordagem CAVE.....   | 26 |
| Figura 4.2 - Modelo de características para (a) interação e (b) apresentação.....                     | 29 |
| Figura 4.3 - Modelo de características para visualização.....   | 30 |
| Figura 4.4 - Atores da abordagem.....   | 31 |
| Figura 4.5 - Trecho de modelo de contexto para entidade Dado.....                                     | 33 |
| Figura 4.6 - Trecho de modelo de contexto para entidade Interação.....                                | 34 |
| Figura 4.7 - Trecho de modelo de contexto para entidade Tarefa.....                                   | 35 |
| Figura 4.8 - Trecho de modelo de contexto para entidade Foco de Representação.....                    | 37 |
| Figura 4.9 - Termos buscados em um repositório de reuso.....  | 38 |
| Figura 4.10 - Informações referentes a dados com valores.....   | 38 |
| Figura 4.11 - Informações referentes a tarefas com valores.....                                       | 39 |
| Figura 4.12 - Informação referente a foco de representação com valor.....                             | 39 |
| Figura 4.13 - Situação de contexto para necessidade de ativos reutilizáveis.....                      | 40 |
| Figura 4.14 - Padrão de regra de contexto para visualizações.....                                     | 40 |
| Figura 4.15 - Regra de contexto para identificar necessidade de novo ativos.....                      | 41 |
| Figura 4.16 - Visualização para identificar necessidade de novo ativos.....                           | 42 |
| Figura 4.17 - Visão conceitual da abordagem CAVE.....   | 43 |
| Figura 5.1 - Papéis previstos na plataforma CAVE.....   | 47 |
| Figura 5.2 - Diagrama de componentes para a plataforma CAVE.....                                      | 47 |
| Figura 5.3 - Exportação do modelo de contexto em formato XMI.....                                     | 49 |
| Figura 5.4 - Trecho de arquivo XMI gerado na exportação do modelo de contexto.....                    | 50 |
| Figura 5.5 - Representação visual do modelo de contexto importado, gerada pela CAVE.....              | 50 |

|   |    |
|---|----|
| Figura 5.6 - Tela para realizar o mapeamento entre informações de contexto importadas e informações suportadas..... | 51 |
| Figura 5.7 - Tela com as opções de foco de representação para uma visualização.....                                 | 52 |
| Figura 5.8 - Tela com a visualização orientada ao contexto .....  | 53 |
| Figura 5.9 - Recorte do diagrama de características de visualização .....   | 54 |
| Figura 5.10 - Função para adicionar leiaute de grafo em uma visualização .....                                      | 54 |
| Figura 5.11 - Função para adicionar texto indicativo em uma visualização .....                                      | 55 |
| Figura 5.12 - Função para adicionar mecanismo de <i>zoom</i> geométrico em uma visualização .....                   | 55 |
| Figura 5.13 - Visualização composta por diferentes elementos .....  | 55 |
| Figura 5.14 - Barra superior para autenticação ou cadastro .....  | 56 |
| Figura 5.15 - Página para autenticação do usuário .....   | 56 |
| Figura 5.16 - Página para cadastro de novo usuário.....   | 56 |
| Figura 5.17 - Página de gerenciamento da plataforma .....   | 57 |
| Figura 5.18 - Página para inserir novo modelo de contexto na plataforma.....  | 58 |
| Figura 5.19 - Página para mapear informações modeladas e informações suportadas ..                                  | 59 |
| Figura 5.20 - Página para mapear definições de contexto modeladas e suportadas.....                                 | 59 |
| Figura 5.21 - Menu de usuário.....  | 60 |
| Figura 5.22 - Notificação de definição de contexto ativa.....   | 60 |
| Figura 5.23 - Página para escolher o foco de representação desejado .....   | 61 |
| Figura 5.24 - Página com a visualização .....   | 62 |
| Figura 5.25 - Página para seleção de definição de contexto.....   | 63 |
| Figura 6.1 - Número de participantes para cada nível de experiência em gerenciamento de projetos de software .....  | 69 |
| Figura 6.2 - Número de participantes para cada nível de experiência em visualização de informação/dashboards.....   | 70 |
| Figura 6.3 - Porcentagem de participantes em cada categoria de experiência em desenvolvimento de software .....     | 70 |
| Figura 6.4 - Visualização para analisar dependências entre classes.....   | 71 |
| Figura 6.5 - Classe Midia destacada e as dependências de outras classes.....  | 72 |
| Figura 6.6 - Visualização com a estrutura de um projeto .....   | 74 |
| Figura 6.7 - Visualização com as colaborações entre os desenvolvedores .....  | 77 |
| Figura 6.8 - Visualização com as tarefas abertas.....   | 81 |
| Figura 6.9 - Visualização com dados sobre as tarefas alocadas a desenvolvedores.....                                | 81 |

## ÍNDICE DE TABELAS

|  |    |
|--|----|
| Tabela 3.1 - Popularidade de termos referentes a visualização (CHEN, 2006) ..... | 19 |
| Tabela 3.2 - Elementos de visualização.....                                      | 23 |
| Tabela 6.1 - Caracterização dos participantes .....                              | 69 |
| Tabela 6.2 - Resultado da execução da tarefa do Cenário A.....                   | 72 |
| Tabela 6.3 - Resultado da execução da tarefa do Cenário B .....                  | 75 |
| Tabela 6.4 - Resultado da execução da tarefa do Cenário C .....                  | 77 |
| Tabela 6.5 - Resultado da execução da tarefa do Cenário D.....                   | 82 |

# CAPÍTULO 1 - INTRODUÇÃO

## 1.1 Motivação

O desenvolvimento de qualquer produto ou serviço novo pode ser caracterizado como uma tarefa de conhecimento intensivo (RAMESH & TIWANA, 1999). O desenvolvimento de software, comumente centrado em projetos, ainda costuma envolver muitos desafios e dificuldades específicas. Ambientes de engenharia de software, visando prover suporte ao desenvolvimento, acabam reunindo uma variedade sempre dinâmica de tecnologias, processos e conhecimentos (GRAMBOW *et al.*, 2012).

Em ambientes multivariados, onde a quantidade de informações disponíveis é grande e a necessidade pela tomada de decisão rápida é alta, metodologias de suporte são bem-vindas, podendo ser essenciais, dependendo da complexidade dos dados disponíveis. Indicadores apropriados que sintetizem as informações e viabilizem análises rápidas e precisas são demandas comuns em organizações de diferentes domínios.

Conhecer todos os aspectos importantes que devem ser considerados para uma análise apurada das necessidades atuais em um projeto, como dados relativos à equipe, ao prazo disponível e aos recursos utilizados, pode ser uma tarefa complexa. Algum aspecto pode ser involuntariamente ignorado, passando despercebido por diferentes membros de uma organização. Mesmo com tais aspectos mapeados, acompanhar seus estados em tempo real e verificar quando uma determinada ação é necessária amplia ainda mais a complexidade deste cenário.

Destacar um contexto de um cenário com muitas situações relevantes pode ser uma abordagem necessária e um desafio a ser superado. A consciência (*awareness*) de uma pessoa sobre os elementos existentes em seu ambiente em certo local e tempo específico, denominada como “percepção de situação” (LI *et al.*, 2013), pode ser considerada como requisito para tomada de decisão apropriada, ou pelo menos, suficientemente baseada em indicadores.

No desenvolvimento de software, tais características de ambientes multivariados acabam sendo verificadas em diferentes escalas, de forma que desafios referentes à percepção também estão presentes. Projetos de software acabam envolvendo questões como planejamento atual, prazos, prioridade de componentes e dependências entre estes e desenvolvedores (TREUDE & STOREY, 2010). Diferentes aspectos podem ser

necessários para cada envolvido em um processo de desenvolvimento de software, a fim de lidar com problemas e decisões críticas.

Mecanismos e estudos para entender e apoiar metodologias de promoção de percepção podem ser considerados como de grande interesse para ambientes multivariados. Como os dados provenientes de diferentes cenários necessitam de diferentes alternativas para análise, abstrações gráficas disponibilizam técnicas para destacar registros importantes e realizar interações para ampliar as formas de efetuar tal análise (SCHOTS *et al.*, 2012).

Como processos de desenvolvimento de software podem ser bem dinâmicos, visualizações podem facilitar o acompanhamento dos dados gerados. No entanto, com um número grande de tarefas sendo realizadas e diferentes papéis envolvidos, reconhecer rapidamente situações críticas pode não ser tão simples ou mesmo imediato. Uma perspectiva de sensibilidade ao contexto pode colaborar nesse sentido, atuando como uma alternativa a fim de ampliar a percepção dos diferentes aspectos que devem ser analisados através do destaque de visualizações apropriadas em determinado momento.

## **1.2 Hipótese e Objetivo**

Reconhecendo a utilidade de técnicas de visualização de informação, que envolvem a adoção de estruturas para representação de informações (CHEN, 2006), este trabalho parte da hipótese de que *a incorporação de conceitos de sensibilidade ao contexto pode indicar a adoção de determinadas representações gráficas mais adequadas e orientadas a problemas atuais*. A partir das dificuldades mencionadas para analisar uma grande quantidade de dados, visando uma tomada de decisão mais eficiente, o objetivo dessa dissertação é *promover a percepção em diferentes situações, com um foco inicial em cenários de desenvolvimento de software, apoiando o uso apropriado de visualizações por meio de técnicas de sensibilidade ao contexto*. Neste sentido, o diferencial deste trabalho está na integração de sensibilidade ao contexto aos conhecimentos obtidos na área de visualização de informação.

Para alcançar o objetivo deste estudo, é importante destacar formas de organizar elementos de contexto, que sejam relevantes para diferentes análises, em um modelo estruturado. A função de cada elemento é importante para relacionar o reconhecimento de eventos em ocorrência com ações a serem apresentadas como resultado. Categorias comuns em modelos de contexto, destacando dimensões e elementos frequentes em

qualquer domínio, também podem ser consideradas como forma de guiar no processo de compilação das informações relevantes para um cenário, a fim de compor o contexto.

Visando oferecer formas de customizar a visualização resultante, fator importante para flexibilizar e ampliar as possibilidades de análise, um levantamento dos elementos de visualização de informação existentes é necessário. Uma estruturação das metáforas e técnicas de visualização pode colaborar para a identificação de qual elemento seria útil para atender a uma necessidade de interação ou apresentação dos dados. Ou seja, espera-se que visualizações possam ser claras, cumprindo seu objetivo geral de permitir análises, com usuários explorando informações, respondendo questões e descobrindo tendências e relações inesperadas (BERTINI, 2004).

A fim de concretizar os conceitos explorados, este trabalho inclui o desenvolvimento de uma plataforma, denominada CAVE (Context-Aware Visualization Engine). Por meio desta plataforma, pretende-se permitir a visualização de dados e a verificação de alertas para situações que mereçam atenção no instante em que ocorrem. A plataforma utiliza como entrada um modelo de contexto, verifica valores de dados provenientes de sistemas de interesse e apresenta visualizações para permitir a análise de situações relevantes em um cenário.

Como forma de aplicação da abordagem, em uma organização que lide com diferentes situações de interesse, o monitoramento de contexto identifica a ocorrência de eventos mapeados. A fim de apoiar os *stakeholders* relacionados a tais eventos, visualizações apropriadas são apresentadas, facilitando a análise de questões e problemas destacados no momento atual. Com tal mecanismo de visualização sensível ao contexto, as representações gráficas que devem ser verificadas em determinado instante são indicadas aos *stakeholders* automaticamente, possivelmente agilizando a identificação de problemas e a tomada de decisão.

Um exemplo que demonstra um caminho de adoção para a presente abordagem pode ser notado em um cenário de desenvolvimento de software, onde diversas tarefas são realizadas e cabe ao gerente de projeto observar o cumprimento de prazos e estabelecer planejamentos. Se muitas tarefas se encontram atrasadas ou pelo menos uma tarefa com alta prioridade foge ao prazo estabelecido, ações devem ser realizadas a fim de mitigar os impactos que venham a acontecer. Com visualizações sensíveis ao contexto, o gerente de projeto poderia perceber, através de representações gráficas automaticamente apresentadas, que o andamento das atividades não se encontra em um ritmo adequado. Regras estabelecidas no contexto do cenário reconheceriam prazos não

respeitados de fases do projeto e indicariam visualizações para tratar o problema de tarefas atrasadas.

### **1.3 Organização**

Esta dissertação está organizada em seis capítulos. O capítulo atual apresentou a motivação para o desenvolvimento da pesquisa, além de destacar os objetivos do trabalho realizado.

O Capítulo 2 apresenta os principais conceitos sobre sensibilidade ao contexto. Uma fundamentação teórica aborda desde os pilares da área de ubiquidade até as características comuns em sistemas sensíveis ao contexto. Uma apresentação dos diferentes elementos existentes em um modelo de contexto é realizada.

A área de visualização de informação é explorada no Capítulo 3. Sua importância é reconhecida em conjunto com os conceitos de percepção, onde visualizações podem ser adotadas para ampliá-la. O levantamento de diferentes elementos que podem compor visualizações também é detalhado.

Após a fundamentação teórica, o Capítulo 4 descreve a abordagem de visualização sensível ao contexto, destacando os conceitos utilizados. A fim de verificar o diferencial com relação a outras abordagens, uma análise comparativa é realizada.

O Capítulo 5 detalha a implementação da plataforma CAVE com suas possibilidades de adoção. A utilização da plataforma é descrita, visando demonstrar as funcionalidades existentes e opções disponíveis.

Já o Capítulo 6 apresenta a avaliação da abordagem por meio de um estudo de observação. O objetivo, a definição e os procedimentos do estudo são descritos, destacando os cenários propostos, bem como as análises obtidas a partir dos resultados fornecidos por participantes.

Finalizando, o Capítulo 7 apresenta as considerações finais deste estudo, ressaltando as contribuições da pesquisa, as limitações identificadas, além de possibilidades de trabalhos futuros.

# **CAPÍTULO 2 - SENSIBILIDADE AO CONTEXTO**

## **2.1 Introdução**

Prover respostas adequadas para cada solicitação é um objetivo comum e importante para diferentes tipos de sistemas. Para determinados casos, o conjunto de requisições pode ser muito grande e variável, o que dificulta o mapeamento entre solicitações e resultados recomendáveis. Nesse sentido, abordagens especiais devem ser consideradas, de modo a viabilizar o gerenciamento de tais possibilidades de respostas.

Uma abordagem que utiliza metodologias para organizar as entradas reconhecidas e suas respectivas saídas aplica o conceito de sensibilidade ao contexto. Sistemas sensíveis ao contexto são caracterizados por extrair informações de ambientes e usuários, visando fornecer respostas customizadas de acordo com cada estado reconhecido (CHEN & KOTZ, 2000).

É importante ressaltar a propriedade extensível de uma abordagem sensível ao contexto, uma vez que é possível ampliar os estados mapeados, adicionando novos elementos ao contexto. No caso de sistemas de informação, pode-se editar o conjunto de entradas e saídas permitidas, o que pode tornar a aplicação do sistema mais eficiente.

Visando adotar o conceito de contexto para uma abordagem voltada para visualização de software, este capítulo reúne a fundamentação teórica sobre a área de sensibilidade ao contexto extraída da literatura. A Seção 2.2 apresenta os conceitos de sensibilidade ao contexto. A Seção 2.3 explica os elementos existentes em um modelo de contexto. As considerações finais acerca da utilização de sensibilidade ao contexto são apresentadas na Seção 2.4.

## **2.2 Conceitos**

Sensibilidade ao contexto é um conceito que, para ser compreendido com clareza, requer o conhecimento de outros fatores relacionados à área. Sua utilização não é restrita a apenas um domínio, por isso, diferentes abordagens podem adotar a sensibilidade ao contexto com uma perspectiva própria. Nesse sentido, é importante destacar a área de origem da sensibilidade ao contexto, bem como uma definição que seja compatível com diferentes aplicações.

### **2.2.1 Ubiquidade**

A sensibilidade ao contexto é uma propriedade presente em alguns sistemas e advém de uma área mais ampla, composta pelos chamados sistemas ubíquos (WEISER, 1991). A denominada “computação ubíqua” envolve o uso de computadores diante de sua disponibilidade constante em um ambiente, o que tornaria seu uso, de certa forma, transparente (ou seja, imperceptível) para as pessoas.

Por ser um campo bem amplo, é importante reconhecer quais propriedades são comuns em projetos de software ubíquo. Um sistema pode ser caracterizado como ubíquo quando atende a um subconjunto destas propriedades. SPÍNOLA *et al.* (2007) apresentam um levantamento de características de computação ubíqua, conforme listado abaixo:

- Onipresença de serviço,
- Invisibilidade,
- Sensibilidade ao contexto,
- Comportamento adaptável,
- Captura de experiência,
- Descoberta de serviço,
- Composição de funções,
- Interoperabilidade espontânea,
- Heterogeneidade de dispositivos, e
- Tolerância a falhas.

É importante destacar que muitas dessas características têm sua definição permeando as fronteiras das demais, ou até mesmo acaba sendo comum a aplicação de múltiplas características em conjunto. Um exemplo está na própria “sensibilidade ao contexto”. Sua função está relacionada à coleta de dados do ambiente, o que também pode ser interpretado pela característica “captura de experiência”, que registra informações para uso posterior. O resultado da aplicação de técnicas voltadas para sensibilidade ao contexto está no fornecimento de respostas apropriadas, o que também pode ser associado à característica “comportamento adaptável”.

### **2.2.2 Contexto**

Em ambientes dinâmicos, gerenciar o conhecimento adquirido ao longo de projetos legados pode colaborar na tomada de decisão, acelerando tal tarefa. Com esse objetivo, capturar e disponibilizar as circunstâncias que tiveram como resultado a tomada de decisão (ou a criação de algum artefato) pode permitir uma reutilização de tal

conhecimento de forma mais sistematizada em situações futuras (NUNES *et al.*, 2007). Essas circunstâncias em que determinado evento ocorre são consideradas informações potencialmente relevantes para o contexto.

CHEN & KOTZ (2000) destacam o sentido vago dado para o termo “contexto”, uma vez que qualquer coisa acontece em certo contexto, em alguma condição. Por isso, de acordo com o uso dado, considerando o domínio abordado, é importante definir como o contexto será caracterizado para tal aplicação.

Diversos autores buscam prover definições gerais para contexto. SCHILIT *et al.* (1994) dividem contexto em três categorias: contexto de computação, incluindo características relacionadas a conectividade e recursos como impressoras, monitores e estações de trabalho; contexto de usuário, considerando dados como perfil, localização e situação social; e contexto físico, como temperatura, condições de trânsito e níveis de ruído. Além disso, segundo os autores, ainda seria possível destacar um contexto de tempo e reunir as categorias anteriores ao longo do tempo, a fim de criar um histórico de contexto. Outra proposta (DEY & ABOWD, 1999) identifica contexto como qualquer informação que possa ser usada para caracterizar a situação de uma entidade, sendo esta entidade qualquer coisa de relevância para a interação entre usuários e sistemas, incluindo-os.

O conceito de contexto adotado neste trabalho segue a definição proposta por BRÉZILLON & POMEROL (1999), que consideram contexto como uma descrição complexa do conhecimento que não é necessariamente explícito, mas bem conhecido. Nesse sentido, o conhecimento contextual pode ser definido como todo conhecimento que é relevante e que pode ser aplicado para entender um determinado problema de decisão com suas circunstâncias. Ainda seria possível distinguir o contexto de acordo com o foco de atenção, uma vez que o foco determina o que deve ser considerado relevante no contexto.

### **2.2.3 Computação Sensível ao Contexto**

A partir de um contexto definido, é necessário investigar metodologias para se fazer um uso eficiente e eficaz de tais tipos de informações. A computação sensível ao contexto atua nessa linha, visando implementar alternativas para cada aplicação em que se busque atender requisitos de ubiquidade.

A fim de dotar uma aplicação de funcionalidades sensíveis ao contexto, ou seja, que forneçam resultados orientados à ocorrência de determinados eventos, quatro categorias de mecanismos podem ser escolhidas (SCHILIT *et al.*, 1994):

- Seleção por proximidade, apresentando opções mais próximas do contexto atual para o usuário interagir;
- Reconfiguração contextual automática, ou seja, um processo que visa adicionar, alterar ou remover componentes e suas conexões de acordo com mudanças no contexto;
- Informações e comandos contextuais, que produzem resultados variados de acordo com o contexto em que em ocorrem; e
- Ações disparadas por contexto, indicando as adaptações para o sistema por meio de regras IF-THEN.

Apesar das diferentes opções de se aplicar orientação ao contexto em sistemas, pode-se identificar objetivamente dois tipos de abordagens: sensibilidade ao contexto ativa e sensibilidade ao contexto passiva (CHEN & KOTZ, 2000). A abordagem ativa envolve a adaptação automática, modificando o comportamento de um sistema a partir do contexto descoberto. A sensibilidade ao contexto passiva parte do princípio de se apresentar o novo contexto para o usuário, ou ao menos guardar o estado para uso posterior. De forma geral, a abordagem passiva atua como um mecanismo de suporte para o usuário sob demanda, enquanto que a abordagem ativa atua de modo mais transparente.

Independente da abordagem escolhida, deve-se buscar uma forma adequada para se organizar o conjunto de informações presentes no contexto. Essa estrutura será importante para o processamento posterior durante o gerenciamento do contexto. Nesse sentido, deve-se planejar a adoção de um modelo de contexto e de seus elementos.

## **2.3 Modelo de Contexto**

Diversas alternativas podem ser adotadas para representar informações relevantes em contextos, como pares chave-valor, modelos orientados a objeto e ontologias, dentre outras (BALDAUF *et al.*, 2007). Um aspecto que deve ser considerado na modelagem de contexto envolve a identificação de técnicas de representação que sejam adequadas para as características da informação contextual (VIEIRA, 2008).

De modo a atender aos requisitos de representação de um contexto, ou seja, estruturar de forma completa e organizada todos os elementos relevantes de tal contexto, optou-se pelo uso da notação UbiFEX (FERNANDES *et al.*, 2008), desenvolvida no

grupo de Reutilização de Software do Programa de Engenharia de Sistemas e Computação (PESC) da COPPE/UFRJ<sup>1</sup>. Esta notação estende a notação Odyssey-FEX (BLOIS *et al.*, 2006), desenvolvida para representar características de domínio em um modelo, com todos os conceitos comuns e variáveis, além de seus relacionamentos, para um mesmo domínio (FERNANDES *et al.*, 2008). A modelagem de características é uma técnica voltada para a identificação de tais características. Esta técnica faz parte da Engenharia de Domínio, que visa à geração de artefatos reutilizáveis (PRIETO-DIAZ & ARANGO, 1991).

Na notação Odyssey-FEX, as características são dispostas como: (i) pontos de variação, em que a característica é especializada para atender aos requisitos; (ii) variantes, que representam funcionalidades especializando os pontos de variação; e (iii) invariantes, características obrigatórias na seleção final (BLOIS *et al.*, 2006). Os relacionamentos entre as características também são exibidos no modelo, apresentando relações de associação e hierarquia entre as características.

A modelagem de características respeita uma abordagem em três dimensões (FERNANDES *et al.*, 2008) (FERNANDES *et al.*, 2011): (i) categoria, em que uma característica pode ser classificada quanto ao seu papel no domínio, como conceitual, referente a conceitos, funcional, designando funcionalidades de tal domínio, ou tecnológica, representando tecnologias adotadas para atender um requisito específico; (ii) variabilidade, em que uma característica pode ser uma variante, uma invariante ou um ponto de variação, dando origem a outras variantes; e (iii) opcionalidade, designando uma característica como mandatória ou opcional. A Figura 2.1 ilustra a adoção da notação Odyssey-FEX em um domínio de aplicativo de dispositivos móveis para a prática de exercícios físicos.

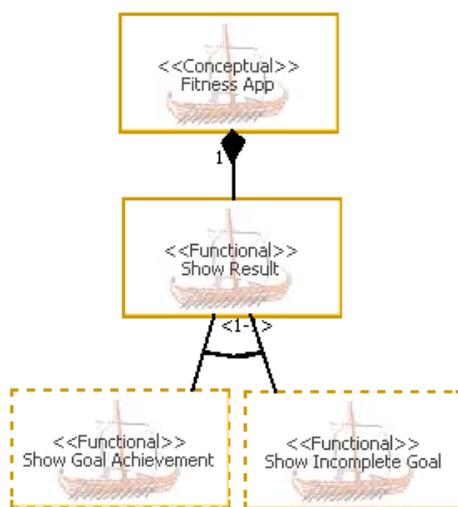
Além desses elementos, regras de composição podem ser especificadas, de forma a indicar a necessidade de incluir ou excluir a seleção de outra característica (LEE *et al.*, 2002). Uma regra de composição entre características é organizada de acordo com o padrão a seguir:

R\_X - <Característica> **requer** <Característica>

R\_Y - <Característica> **exclui** <Característica>

---

<sup>1</sup> Equipe de Reutilização de Software: <http://reuse.cos.ufrj.br>

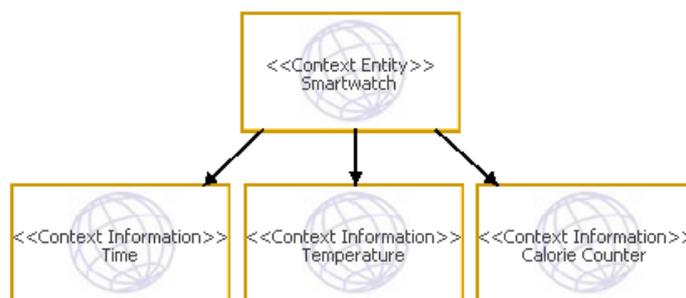


**Figura 2.1** - Exemplo de modelo de características com a notação Odyssey-FEX

A extensão realizada pela notação UbiFEX adiciona novas categorias, expressões e regras à Odyssey-FEX (FERNANDES *et al.*, 2011). Em termos de categorias, foram criados dois novos elementos, bem como propriedades para sua descrição. Tais elementos são:

- **Característica de Entidade de Contexto:** Representa entidades de contexto para o domínio em questão, atuando como dimensões em um modelo de contexto.
  - Propriedades: nome e descrição.
- **Característica de Informação de Contexto:** Representa dados relevantes que são necessários para descrever uma entidade de contexto.
  - Propriedades: nome, descrição, persistência, tipo base, composição e aquisição.

Os relacionamentos possíveis em diagramas com características de entidade e de informação de contexto são os mesmos relacionamentos UML permitidos pela notação Odyssey-FEX, como associação, herança e composição. A Figura 2.2 apresenta um exemplo de modelo de características de contexto para o domínio de relógios inteligentes, que ilustra relacionamentos cuja semântica é “has” (FERNANDES *et al.*, 2011).



**Figura 2.2** – Exemplo de modelo de características de contexto

No que se refere a expressões, foi incluído um novo tipo à Odyssey-FEX, através da notação UbiFEX. As chamadas definições de contexto descrevem situações de destaque ou propriamente contextos para um domínio. Sua representação envolve um nome e uma expressão. Expressões de definições de contexto podem ser formadas por uma característica de informação de contexto (<CIC>), um operador relacional, um valor ou até mesmo por uma composição de expressões usando operadores lógicos (FERNANDES *et al.*, 2011). A Figura 2.3 apresenta uma notação BNF (Backus–Naur Form) descrevendo a formação de uma expressão. Um exemplo de definição de contexto pode ser consultado na Figura 2.4.

```

<expressão> ::= <CIC><operador-relacional><valor>
              | <expressão><operador-lógico><expressão>
              | NOT<expressão>

<operador-relacional> ::= > | < | >= | <= | = | <>

<operador-lógico> ::= AND | OR

<valor> ::= <string> | <int> | <float> | <boolean> | <date> | <time>

```

**Figura 2.3** - Formação de uma expressão de definição de contexto (FERNANDES *et al.*, 2011)

```

Meta alcançada
Smartwatch.Calorie Counter >= 300
AND
Smartwatch.Time > 18:00

```

**Figura 2.4** - Exemplo de definição de contexto

De forma a definir como um sistema irá se comportar diante da ocorrência de uma definição de contexto específica, regras de contexto são propostas para determinar quais ações serão realizadas, seguindo o mecanismo de ações disparadas por contexto (SCHILIT *et al.*, 1994). Uma regra é composta por um nome e uma expressão.

Uma expressão de regra de contexto é formada por um campo antecedente, o operador *implies* e um campo consequente. O campo antecedente é composto por uma expressão que pode conter definições de contexto, características e operadores lógicos. O operador *implies* indica que, caso o campo antecedente seja verdadeiro, o consequente deverá ser executado. O campo consequente é uma expressão que pode conter

características e operadores lógicos. Dessa forma, uma expressão de regra de contexto determina quais características devem ser selecionadas ou removidas da configuração de um produto (FERNANDES *et al.*, 2011). A Figura 2.5 apresenta um exemplo de regra de contexto para apresentar um resultado satisfatório em um aplicativo para exercícios físicos.

|  |
|--|
| <p style="text-align: center;"><b>Regra de Contexto 1 (RC)</b><br/>Meta Alcançada <i>implies</i> Show Goal Achievement</p> |
|--|

**Figura 2.5** – Exemplo de regra de contexto

Através da notação UbiFEX, é possível descrever um modelo de contexto com todos os tipos de elementos necessários. É importante que as modelagens de domínio e de contexto sejam realizadas por especialistas, ou mesmo um único especialista para ambas as áreas, desde que seja capaz de identificar todas as informações relevantes, de forma a obter resultados positivos no sistema sensível ao contexto a ser projetado.

## 2.4 Considerações Finais

Diversos problemas podem ser abordados através de metodologias que envolvam o gerenciamento de contexto. Sistemas sensíveis ao contexto permitem resultados adaptados para as circunstâncias em que as demandas ocorrem. Um ponto que merece atenção é a composição do contexto, isto é, a definição dos elementos que devem fazer parte do modelo. Nessa etapa, especialistas de domínio e de contexto devem ser consultados, bem como potenciais usuários e outros atores pertinentes, uma vez que, dependendo do objetivo do sistema, ouvir a opinião dos futuros usuários pode ser uma alternativa interessante para se buscar um modelo de contexto mais completo.

Em domínios bem variados, cujo número de candidatos a entidades e informações de contexto é grande, um modelo bem definido com definições e regras de contexto pode ser um ponto de partida para estruturar o conhecimento e permitir o desenvolvimento de um sistema sensível ao contexto sem muitos obstáculos. Tal sistema, com resultados apropriados, pode servir como mecanismo de suporte para diversos cenários. Os resultados das regras de contexto podem auxiliar o desempenho de diferentes atividades, provendo funcionalidades customizadas para cada evento.

No entanto, com um grande número de informações e definições de contexto para gerenciar, mecanismos de suporte podem ser necessários. Representar graficamente os dados provenientes desses elementos de contexto pode ser uma forma de simplificar análises. Visualizações são conhecidas como interessantes propostas de apoio à tomada

de decisão, e poder aliar a sua apresentação de acordo com o contexto traz possibilidades de uso interessantes, com uma visualização mais adequada para cada momento.

No próximo capítulo, são apresentados os principais conceitos relacionados à área de visualização de informação. Diferentes elementos existentes em visualizações são identificados a fim de possibilitar a adoção de representações de apoio à decisão mais adequadas.

# CAPÍTULO 3 - VISUALIZAÇÃO DE INFORMAÇÃO

## 3.1 Introdução

O problema da falta de percepção sobre as diversas atividades sendo desenvolvidas em um cenário é decorrente da falta de mecanismos que permitam e estimulem a comunicação de informações entre entidades de tal cenário. De forma a destacar pontos relevantes e possibilitar análises sob diferentes perspectivas, abstrações visuais são alternativas eficientes para representação e interpretação de dados. A área de visualização de informação reúne o conhecimento em torno da aplicação de diferentes técnicas e paradigmas para representar informações. O objetivo da adoção de elementos de visualização está na ação de extrair ideias e gerar conhecimento a partir de dados caóticos e com ruído (CHEN, 2006).

Devido à sua ampla aplicabilidade e aos resultados favoráveis em mecanismos de suporte ao desempenho de diferentes atividades, a utilização de técnicas de visualização de informação para acompanhar tarefas relacionadas ao desenvolvimento de software culminou na especificação de um campo próprio, a visualização de software. Uma discussão sobre como a adoção de visualização pode colaborar em projetos de software se faz necessária para o entendimento da abordagem descrita neste trabalho.

A composição de uma visualização não pode ser considerada como uma tarefa simples, uma vez que a escolha de técnicas e metáforas de leiaute varia de acordo com os dados a serem representados graficamente. Como a interpretação da visualização é algo subjetivo, a adição ou remoção de determinada técnica pode colaborar ou prejudicar o processo de análise. De forma a entender melhor as possibilidades de aplicação de tais técnicas, visando à identificação dos elementos de visualização mais comuns, foram adotados como procedimentos uma revisão informal da literatura e a participação em uma *quasi*-revisão sistemática da literatura (SCHOTS *et al.*, 2015).

Buscando apresentar e discutir a importância da visualização de informação para uma abordagem de promoção de percepção em diferentes cenários, este capítulo reúne a fundamentação teórica sobre a área de visualização de informação bem como uma contribuição desenvolvida para este domínio, na forma de um levantamento dos diferentes elementos existentes. A Seção 3.2 apresenta alguns conceitos de percepção

(*awareness*). A Seção 3.3 descreve com mais detalhes a área de visualização de informação, incluindo uma breve caracterização de visualização de software. A Seção 3.4 reúne o levantamento realizado sobre os diferentes elementos de visualização e as considerações finais são apresentadas na Seção 3.5.

## 3.2 Percepção

Projetos desenvolvidos em organizações que envolvem muitas pessoas em equipes e locais separados apresentam desafios de alta complexidade. Não é fácil manter a percepção (ou *awareness*) das diferentes atividades desempenhadas por todos os membros, além da situação global do projeto e as tarefas com maior prioridade, pois isto inclui numerosas variáveis, não bastando apenas acompanhar uma simples métrica, ao passo que, dependendo das tarefas atuais, a granularidade necessária da percepção pode variar sensivelmente (TREUDE & STOREY, 2010). Investir atenções em mecanismos de percepção pode demandar muito tempo (LATOZA *et al.*, 2006), mas os resultados são consideravelmente positivos para o sucesso de diversos projetos (GUTWIN *et al.*, 2004).

A importância de se promover uma percepção geral em cenários complexos pode ser ressaltada pela definição dada por DOURISH & BELLOTTI (1992), onde percepção traduz-se como um entendimento das atividades de outros membros de uma equipe, o que fornece um contexto para a sua própria atividade. GUTWIN *et al.* (2004) destacam a necessidade de percepção do ambiente de trabalho para um conhecimento efetivo por parte das equipes em cenários colaborativos. De uma forma mais direta, as pessoas utilizam a percepção para realizar ações em um ambiente (GIBSON, 1979).

Considerando as diversas formas de promover percepção, duas categorias podem ser derivadas: (i) percepção ativa e (ii) percepção passiva. Em (i), os mecanismos de promoção adotados precisam de dados gerados explicitamente para promover a percepção, como mensagens sendo enviadas diretamente ao interessado (TREUDE & STOREY, 2010). Já em (ii), os mecanismos utilizam informações já disponíveis em um ambiente compartilhado por diferentes profissionais (DOURISH & BELLOTTI, 1992).

Em processos de desenvolvimento de software, por exemplo, que podem gerar uma grande quantidade de dados, há um consenso em termos da necessidade de estudos sobre as formas como recursos disponíveis podem apoiar a percepção e o entendimento dos dados gerados como um todo (SCHOTS *et al.*, 2012) (TREUDE & STOREY, 2010). Abstrações visuais que permitam destacar informações relevantes para um determinado cenário a partir dos dados podem se colocar como alternativas relevantes no objetivo

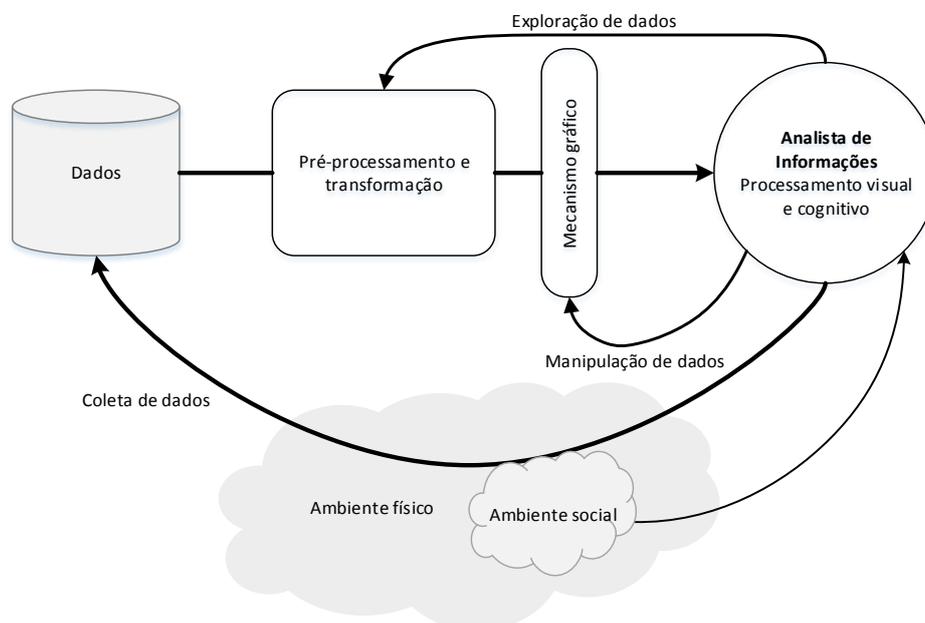
discutido de promover percepção, seja para ambientes colaborativos ou mesmo para um indivíduo trabalhando isoladamente, mas que precisa lidar com diferentes dados ao mesmo tempo.

### 3.3 Visualização

No passado, visualização era usualmente associada ao processo de construir mentalmente uma imagem visual (Shorter Oxford English Dictionary, 1972). Atualmente, ela está mais relacionada a representações gráficas de dados ou conceitos. Com isso, visualizações passaram de um construto interno da mente para um artefato externo, visando apoiar a tomada de decisão, atuando como uma ferramenta cognitiva (WARE, 2004). Como uma definição mais formal, a visualização de informação é composta por uma representação gráfica e interativa de dados abstratos, podendo ser suportada por computador, cujo objetivo é ampliar a cognição do observador (CARD *et al.*, 1999).

De modo a compreender todos os fatores envolvidos no processo de visualização, a Figura 3.1 apresenta quatro estágios básicos. De acordo com WARE (2004), tais estágios podem ser identificados como:

- Coleta e armazenamento de dados;
- Pré-processamento projetado para transformar dados em algo inteligível;
- Exibição em algum dispositivo e utilização de algoritmos gráficos que produzem uma imagem; e
- Estimulação do sistema perceptivo e cognitivo humano (observador).



**Figura 3.1** - Estágios de visualização (WARE, 2004)

A coleta de dados é realizada por um *stakeholder* com interesse em certos indicadores. O ambiente físico pode ser uma fonte de dados, enquanto o ambiente social pode influenciar na escolha dos dados e na sua forma de interpretação. Transformações são realizadas nos dados em uma fase anterior à visualização. A fase de visualização pode ser interativa, com o *stakeholder* manipulando parâmetros de forma a obter resultados mais relevantes.

As aplicações de visualização de informação são bem variadas. SHNEIDERMAN (1996) aponta as possibilidades em áreas como representações visuais, visualização científica e visualização de informação abstrata. Uma representação visual apresenta recursos como: fornecer orientação ou contexto, permitir seleção de regiões e prover *feedback* dinâmico para verificar mudanças nos dados. Visualizações científicas possibilitam ver e compreender fenômenos atômicos e cósmicos. Já para informações abstratas, visualizações possuem a capacidade de revelar padrões, agrupamentos ou *outliers* nos mais diferenciados tipos de conjuntos de dados.

Apesar das vantagens apresentadas, a visualização de informação por si só não pode ser considerada como a solução para todos os problemas relativos a dificuldades em analisar, explorar e interpretar dados. Uma aplicação de visualização de informação não garante respostas, comprovação de hipóteses ou descoberta de tendências. É necessário que o uso dos seus elementos, ou seja, das diferentes técnicas de visualização, seja feito de modo apropriado.

Dois critérios podem ser adotados para avaliar se a representação gráfica está adequada: expressividade e efetividade (MACKINLAY, 1986 *apud* SCHOTS *et al.*, 2014). O critério de expressividade identifica se uma linguagem gráfica expressa fielmente uma informação desejada. A efetividade avalia a capacidade da linguagem em explorar as capacidades do meio de exibição e do sistema visual do observador.

Se as visualizações não forem claras o suficiente e se os usuários tiverem que recorrer a outros meios, ou seja, se as visualizações não podem oferecer o suporte para o qual foram produzidas, sua função principal acaba sendo perdida (BERTINI, 2004). Nesse sentido, é importante conhecer e avaliar técnicas de visualização a serem compostas e implementadas em uma representação resultante, a fim de verificar se esta permite efetuar análises intuitivas.

### 3.3.1 Técnicas de Visualização

Não existe um padrão para a composição de visualizações, uma vez que cada representação gráfica atende a um objetivo específico e diferentes requisitos podem ser esperados. Apesar de parecer vantajoso oferecer muitas opções de técnicas de visualização para um observador, é importante ressaltar que uma grande diversidade de alternativas pode dificultar ainda mais a descoberta da técnica necessária para uma situação de exploração e análise.

Quando o volume de dados a ser explorado é muito grande, a sobrecarga de informação (*information overloading*) pode ser mitigada por meio de certas técnicas, como, por exemplo, filtragem (CHEN, 2006). Do mesmo modo, a aplicação de determinada técnica de visualização ou a escolha de alguma metáfora ou leiaute pode prejudicar a interpretação, trazendo sobrecarga cognitiva ao observador.

Apesar da ausência de padronização na utilização de técnicas de visualização, alguns arquétipos são propostos para objetivos comuns em representações gráficas. Como exemplo, um princípio básico é o mantra “Visão geral primeiro, *zoom* e filtragem, então detalhes sob demanda”, conhecido na literatura de visualização de informação como “*information seeking mantra*” (SCHNEIDERMAN, 1996).

Em termos de possibilidades para composição de visualizações, o número de técnicas é grande, com novas propostas sendo apresentadas na literatura constantemente. Considerando a popularidade de técnicas de visualização, a partir de uma classificação própria, um levantamento presente em (CHEN, 2006) ilustra a adoção de diferentes elementos em visualizações através de buscas em páginas *web*, com as menções a termos de visualizações nas páginas indexadas pelo mecanismo de buscas Google. A Tabela 3.1 apresenta os termos mais relevantes nos resultados com o número de menções. Alguns termos foram traduzidos para o português, para maior clareza.

Dentre as diferentes técnicas e classificações para elementos de visualização, SCHNEIDERMAN (1996) propõe uma taxonomia por tipo de dados, com categorias como: unidimensional, bidimensional, tridimensional, multidimensional, temporal, árvore e rede.

Dados unidimensionais referenciam registros organizados de maneira sequencial ou linear, como textos ou listas. A categoria bidimensional inclui atributos que podem ser relacionados a áreas de mapas ou leiautes semelhantes. Dados tridimensionais envolvem itens com volumes e relacionamentos complexos. A categoria multidimensional

referencia itens com n atributos que podem ser representados em espaços n-dimensionais. Dados temporais incluem itens cuja representação necessita de linhas temporais, que possuem início e fim, podendo haver sobreposição de itens. A categoria árvore envolve estruturas hierárquicas e de árvore, em que cada item tem um relacionamento com outro item de hierarquia acima ou abaixo. Dados de redes incluem itens com um número arbitrário de relacionamentos com outros itens.

**Tabela 3.1** - Popularidade de termos referentes a visualização (CHEN, 2006)

| <b>Termo de Visualização</b>                               | <b>Número de Menções</b> |
|--|--------------------------|
| Foco + contexto ( <i>Focus + Context</i> )                 | 6980                     |
| Evolução ( <i>Evolution</i> )                              | 4370                     |
| Grafo ( <i>Graph Drawing</i> )                             | 3200                     |
| Estudo empírico ( <i>Empirical Study</i> )                 | 2750                     |
| Visão olho de peixe ( <i>Fisheye</i> )                     | 1960                     |
| Hiperbólico ( <i>Hyperbolic</i> )                          | 1910                     |
| <i>Treemap</i>   | 934                      |
| <i>Spotfire</i>  | 808                      |
| SOM ( <i>Self-Organizing Map</i> )                         | 659                      |
| Semiótica ( <i>Semiotics</i> )                             | 563                      |
| Deteção de tendências ( <i>Detect Trend</i> )              | 356                      |
| <i>Pathfinder</i>  | 300                      |
| Deteção de mudança abrupta ( <i>Detect Abrupt Change</i> ) | 48                       |

Em um nível alto de abstração, técnicas de visualização, que podem ser especializadas em um número maior de técnicas, são identificadas como tarefas em (SCHNEIDERMAN, 1996), sendo apresentadas como:

- **Visão geral**
  - Destaca a visão geral de uma coleção inteira;
- **Zoom**
  - Amplia ou reduz o tamanho de uma determinada parte de uma coleção;
- **Filtragem**
  - Filtra itens de uma coleção nos quais não há interesse;
- **Detalhes sob demanda**
  - Seleciona um item ou conjunto e obtém detalhes se for necessário;
- **Relação**
  - Visualizar relacionamentos entre itens;
- **Histórico**
  - Mantém um histórico de interações como desfazer e refazer;
- **Extração**

- Permite a extração de grupos de itens de uma coleção.

A implementação de uma técnica de visualização pode variar de acordo com o domínio abordado, onde a representação gráfica de um item pode exibir detalhes adicionais para ressaltar especificidades dos registros pertencentes a tal domínio. Nesse sentido, é importante conhecer as oportunidades de adoção de visualização de informação para a situação objetivada. Para dados relativos especificamente ao desenvolvimento de software, a visualização de software pode servir como um mecanismo eficiente de suporte.

### **3.3.2 Visualização de Software**

Projetos de software costumam apresentar problemas relacionados à compreensão do sistema sendo desenvolvido (BALL & EICK, 1996) (TREUDE & STOREY, 2010), causados por diversos fatores, como documentação desatualizada e rotatividade na equipe de desenvolvedores. Devido à própria natureza do software - intangível, sem formato físico ou tamanho - ferramentas visuais se apresentam como uma alternativa para apoiar os membros da equipe de um projeto a compreenderem ou se lembrarem dos diferentes aspectos relacionados ao software em questão (BALL & EICK, 1996) (SCHOTS & WERNER, 2012).

A visualização de software, pertencente à área mais ampla de visualização de informação, era inicialmente conhecida como a visualização de algoritmos e programas. Tal definição pode ser considerada restrita, uma vez que muitos aspectos da engenharia de software são desconsiderados nessa afirmação. Com isso, uma forma mais adequada é considerar a visualização de software como a visualização de artefatos relacionados ao software e ao seu processo de desenvolvimento (DIEHL, 2007).

BALL & EICK (1996) destacam três propriedades básicas que costumam ser consideradas para visualização, como:

- **Estrutura**
  - Relacionamentos entre entidades de software são representados comumente por grafos, e disponibilizadas em diversas ferramentas de análise;
- **Comportamento**
  - Estruturas e animações podem representar comportamento de algoritmos (em alto nível) (BROWN, 1988) ou revelar problemas referentes a desempenho e defeitos (quando em baixo nível) (CHILIMBI *et al.*, 1995);
- **Código**

- Técnicas para exibição diferenciada de trechos de código, destacando elementos (BAECKER & MARCUS, 1990).

Além de estrutura e comportamento, DIEHL (2007) ainda destaca o aspecto da evolução, que foca a atenção para o desenvolvimento de software e as mudanças ocorridas no código-fonte ao longo de diferentes etapas do processo.

É importante notar que a utilização de visualização em processos de desenvolvimento de software pode atender a diferentes perspectivas. Um ambiente de desenvolvimento pode envolver diferentes *stakeholders* e cada um deles tem seu objetivo próprio no contexto de um projeto, o que definirá requisitos diferenciados para cada visualização adotada como apoio. TELEA *et al.* (2010) destacam os valores principais buscados em uma visualização para três tipos de *stakeholders*, como:

- **Desenvolvedor**
  - Obter visualizações customizadas com poucas operações necessárias;
- **Gerente/Arquiteto**
  - Descobrir problemas de evolução de software e gerenciar diferentes aspectos do projeto, como equipe, prazos e qualidade;
- **Consultor**
  - Obter visualizações simples, facilmente configuradas, sem necessidade de produzir resultados exatos e detalhados.

Dadas as possibilidades de uso de visualizações de forma geral, além das aplicações voltadas para visualização de software, é importante conhecer as diferentes customizações disponíveis, ou seja, os diferentes paradigmas e técnicas existentes. Nesse sentido, uma análise da literatura foi escolhida para identificar os elementos de visualização já propostos em outros estudos.

### **3.4 Elementos de Visualização**

Visando um melhor entendimento e mapeamento da extensão do domínio de visualização, buscou-se realizar um levantamento dos elementos comuns em representações gráficas (SCHOTS *et al.*, 2015). Neste trabalho, um elemento de visualização é interpretado como um conceito que pode ser representado por um paradigma ou uma técnica de visualização. O levantamento foi realizado para identificar candidatos a elementos para o domínio de visualização.

A metodologia adotada foi baseada em duas etapas: (i) uma revisão informal da literatura, inicialmente identificando diferentes elementos presentes em visualizações, e

(ii) uma *quasi*-revisão sistemática da literatura (SCHOTS *et al.*, 2014), que serviu para confirmar o uso dos elementos anteriormente identificados e para completar o resultado com novos candidatos. Apesar do foco de (ii) ter sido voltado para abordagens de visualização de software propostas para apoiar reutilização de software, o alcance da revisão considera um levantamento inicial em termos de estudos em engenharia de software. As duas etapas da metodologia foram realizadas em conjunto com um aluno de doutorado, dado o interesse comum no objeto de estudo.

A partir da *quasi*-revisão sistemática da literatura, os resultados obtidos podem ser utilizados para uma análise geral. Em termos de popularidade dos elementos de visualização, das 34 abordagens apresentadas nas 36 publicações selecionadas, 6 elementos foram citados em mais de 10 abordagens, a saber: seleção, navegação, detalhamento (*drill-down*), agrupamento, realce e rotulagem (*labeling*). A maioria destes elementos está diretamente relacionada a interações sobre representações gráficas.

Com as duas etapas da metodologia completas, o resultado obtido no levantamento aponta um total de 62 elementos de visualização. A Tabela 3.2 apresenta os elementos resultantes de tal levantamento.

É importante ressaltar que não se considera o resultado desse levantamento como completo e definitivo, mas que necessita ser completado com novos elementos no decorrer do tempo, ainda que o resultado atual apresente uma abrangência razoável no domínio de visualização. As descrições detalhadas, com imagens e possíveis restrições identificadas no uso dos elementos, além de suas classificações, encontram-se em (SCHOTS *et al.*, 2015).

**Tabela 3.2** - Elementos de visualização

| <b>Elementos de Visualização</b>                       |  |
|--|--|
| Foco + Contexto ( <i>Focus + Context</i> )             | Árvore ( <i>Tree</i> )                                       |
| Lente Mágica ( <i>Magic Lens</i> )                     | Partição ( <i>Partition</i> )                                |
| Distorção ( <i>Distortion</i> )                        | Cluster  |
| Visão Olho de Peixe ( <i>Fisheye View</i> )            | Gráfico de Pacote ( <i>Pack</i> )                            |
| Exibição Polifocal ( <i>Polyfocal Display</i> )        | Detalhe sob Demanda ( <i>Details on Demand</i> )             |
| Visão Geral ( <i>Overview</i> )                        | Detalhamento ( <i>Drill-Down</i> )                           |
| Visão Geral + Detalhe ( <i>Overview + Detail</i> )     | Rotulagem ( <i>Labeling</i> )                                |
| Mini Mapa ( <i>Minimap</i> )                           | Texto Indicativo ( <i>Tooltip</i> )                          |
| Visão Geral de Miniatura ( <i>Thumbnail Overview</i> ) | Binning  |
| Sobreposição ( <i>Overlap</i> )                        | Apresentação Sequencial ( <i>Sequential Presentation</i> )   |
| Transparência ( <i>Transparency</i> )                  | Apresentação Simultânea ( <i>Simultaneous Presentation</i> ) |
| Visão Estéreo ( <i>Stereovision</i> )                  | Filtragem ( <i>Filtering</i> )                               |
| Flipping   | Filtragem Dinâmica ( <i>Dynamic Filtering</i> )              |
| Agrupamento ( <i>Clustering</i> )                      | Filtragem Estática ( <i>Static Filtering</i> )               |
| Animação ( <i>Animation</i> )                          | Realce/Mitigação ( <i>Highlighting/Mitigation</i> )          |
| Leiaute ( <i>Layout</i> )                              | Inclusão/Remoção ( <i>Inclusion/Removal</i> )                |
| Perspectiva 2D ( <i>2D Perspective</i> )               | Colapso/Expansão ( <i>Collapse/Expand</i> )                  |
| Perspectiva 3D ( <i>3D Perspective</i> )               | Amostragem ( <i>Sampling</i> )                               |
| Matriz de Dispersão ( <i>Scatter Plot</i> )            | Sintonia/Ajuste ( <i>Tuning/Tweaking</i> )                   |
| Coordenadas Paralelas ( <i>Parallel Coordinates</i> )  | Segmentação ( <i>Segmentation</i> )                          |
| Tabela ( <i>Table</i> )                                | Deslocamento ( <i>Panning</i> )                              |
| Lista ( <i>List</i> )                                  | Arrastar e Soltar ( <i>Drag &amp; Drop</i> )                 |
| Chord  | Ordenação ( <i>Sorting</i> )                                 |
| Leiaute de Força ( <i>Force</i> )                      | Relacionamento ( <i>Linking</i> )                            |
| Histograma ( <i>Histogram</i> )                        | Browsing   |
| Gráfico de Torta ( <i>Pie</i> )                        | Navegação ( <i>Navigation</i> )                              |
| Leiaute de Pilha ( <i>Stack</i> )                      | Consulta ( <i>Querying</i> )                                 |
| Bundle   | Seleção ( <i>Selection</i> )                                 |
| Leiaute Hierárquico ( <i>Hierarchical</i> )            | Agregação ( <i>Aggregation</i> )                             |
| Treemap  | Rotação ( <i>Rotate</i> )                                    |
| Árvore em Cone ( <i>Cone Tree</i> )                    | Zoom Geométrico ( <i>Geometric Zooming</i> )                 |
| Radial ( <i>Radial</i> )                               | Zoom Semântico ( <i>Semantic Zooming</i> )                   |

### 3.5 Considerações Finais

Mecanismos de suporte baseados em visualização podem fornecer diversas vantagens aos seus usuários, como novas ideias, indicações de problemas e soluções para uma questão e compreensão de uma determinada situação representada pelos dados. As possibilidades de exploração e análise dos dados por meio de visualizações tornam a sua adoção indicada para diferentes domínios.

Identificadas as necessidades de um cenário, visualizações podem ser produzidas com os elementos adequados para representar determinada situação de interesse. A partir de um contexto mapeado, conforme abordado no Capítulo 2, elementos de visualização

podem ser selecionados para atender aos requisitos expostos. Uma abordagem que reúna tais conceitos tornará a adoção de visualizações mais apropriada, com o valor dos resultados mais adequado para análises necessárias. Tal abordagem, proposta no presente trabalho, é apresentada no capítulo a seguir.

# CAPÍTULO 4 - VISUALIZAÇÕES SENSÍVEIS AO CONTEXTO

## 4.1 Introdução

Conforme discutido no Capítulo 2, dada a importância de se conhecer o contexto de um cenário, independente do domínio abordado, de forma a se verificar informações relevantes, mitigar riscos e propor soluções para problemas, metodologias que considerem esse fator podem ter ganhos para aumentar a eficiência de sua aplicação. Para ambientes de desenvolvimento de software, onde grande quantidade de variáveis está envolvida, como artefatos, papéis e atividades, gerenciar esse volume de dados de forma estruturada pode ser uma tarefa complexa.

Técnicas de suporte ao gerenciamento de conhecimento costumam ser aplicadas quando uma organização precisa estruturar aprendizados individuais e compartilhá-los entre seus membros (GRAMBOW *et al.*, 2012). Caracterizar o contexto de um cenário de desenvolvimento pode ser considerado como uma etapa inicial para analisar dados das diferentes tarefas em um projeto de software e derivar conhecimento a partir do progresso atual e lições aprendidas de projetos anteriores. Com esse conhecimento destacado, devem ser aplicados meios para comunicá-lo aos membros de tal organização.

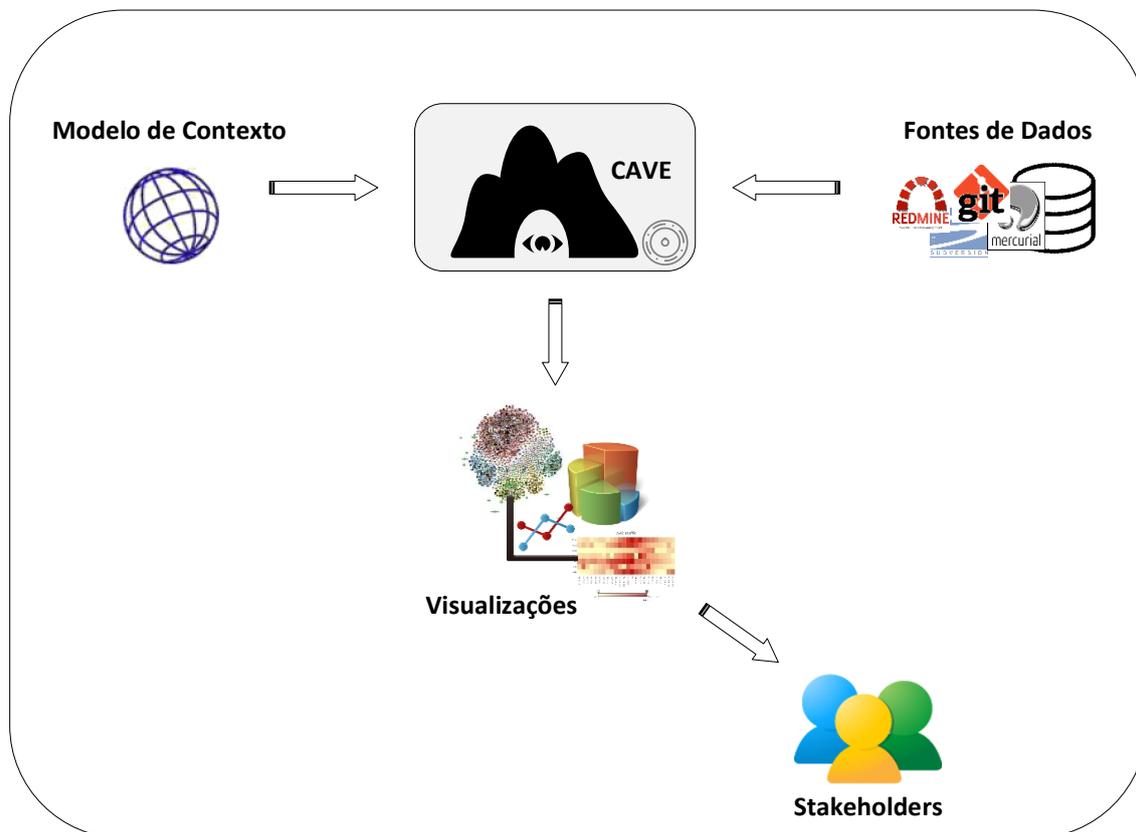
Como discutido no Capítulo 3, a visualização de informação se posiciona como uma alternativa adequada para esse fim. Visualizações apropriadas para cada situação podem colaborar na identificação de pontos importantes e interpretação de indicadores de forma mais rápida. Nesse sentido, visualizações orientadas ao contexto são exploradas na abordagem proposta neste trabalho, denominada CAVE (Context-Aware Visualization Engine).

Além desta seção introdutória, este capítulo está organizado da seguinte forma: a Seção 4.2 traz uma visão geral da abordagem proposta; a Seção 4.3 apresenta uma análise comparativa com outras abordagens encontradas na literatura. Por fim, na Seção 4.4, as considerações finais deste capítulo são apresentadas.

## 4.2 Visão Geral da Abordagem

Visando prover visualizações de software mais recomendadas de acordo com as necessidades atuais de uma organização, a abordagem CAVE apresenta uma infraestrutura para especificar um mecanismo de promoção de percepção (*awareness*),

considerado essencial para um processo de desenvolvimento de software (TREUDE & STOREY, 2010). Realizando uma análise de contexto, verificando dados gerados pela organização e apresentando visualizações para diferentes *stakeholders*, a visão geral da abordagem é representada na Figura 4.1.



**Figura 4.1** - Visão geral da abordagem CAVE

A abordagem considera um modelo de contexto construído com a notação UbiFEX, associando-o a fontes de dados, que possibilitam a verificação dos valores dos elementos dispostos no modelo. Adotando uma abordagem de sensibilidade ao contexto passiva (CHEN & KOTZ, 2000), notificações sobre mudanças no contexto são apresentadas para os diferentes *stakeholders*. Com uma concordância para o alerta apresentado, é viabilizada a apresentação de visualizações para apoiar as situações em destaque no novo contexto.

Como a abordagem inclui dois conceitos de áreas separadamente bem consolidadas, é importante definir as especificidades que são propostas para tais conceitos em sua aplicação em conjunto. Assim, as Seções 4.2.1 e 4.2.2 apresentam como é a adoção de visualização e sensibilidade ao contexto, respectivamente, para a definição da CAVE, enquanto a Seção 4.2.3 apresenta uma esquematização da abordagem completa, com os conceitos apresentados.

## 4.2.1 Características de Visualização

Existem diversas visualizações de informação disponíveis na literatura, representando vários tipos de dados e sob diferentes focos. E como os objetivos de cada visualização são diferentes, não existe uma representação que seja melhor ou mais eficiente que outra (MANN, 1999). Nesse sentido, é importante que a escolha da visualização seja feita seguindo critérios, sendo apoiada de forma sistemática (ROBERTSON, 1991).

Para viabilizar a escolha de uma visualização propícia para uma dada situação, é necessário utilizar um método que permita customizar uma visualização desde a seleção de sua metáfora até a escolha de suas técnicas. Dessa forma, uma abordagem de engenharia de domínio<sup>2</sup> para mapeamento de conceitos sobre visualizações (VASCONCELOS *et al.*, 2014) pode se colocar como uma alternativa adequada, permitindo a seleção de características necessárias para atender o objetivo de uma visualização.

### 4.2.1.1 Análise de Domínio

Conforme mencionado no Capítulo 3, algumas técnicas de visualização foram identificadas como candidatas a elementos para o domínio de visualização, compondo pontos de variação, variantes e invariantes (CLEMENTS & NORTHROP, 2001). Assim, optou-se por efetuar uma análise de domínio de forma a organizar os referidos elementos para compor um modelo de características para o domínio de visualização (VASCONCELOS *et al.*, 2014). A partir disto, um recorte desse modelo implicaria na composição de uma visualização que atenderia a determinadas necessidades de análise.

É importante ressaltar que a análise de domínio, feita em colaboração com um aluno de doutorado, foi realizada a partir dos resultados obtidos no levantamento de elementos de visualização descrito na Seção 3.5. Tal estudo encontra-se descrito em (SCHOTS *et al.*, 2015).

As características que, para a Análise de Domínio, representam as necessidades de compreensão para as atividades em determinado domínio, são dispostas, conforme mencionado no Capítulo 2, como: pontos de variação, variantes e invariantes. O modelo inclui propriedades como: categoria, variabilidade e opcionalidade. Como a aplicação de determinadas técnicas de visualização só faz sentido para um conjunto restrito de

---

<sup>2</sup> A engenharia de domínio visa oferecer ativos reutilizáveis representativos de um domínio específico (CLEMENTS & NORTHROP, 2001)

metáforas, é esperado que um modelo de características para o domínio de visualização apresente algumas regras de composição, restringindo as seleções.

A estruturação do modelo foi realizada utilizando-se a notação Odyssey-FEX (BLOIS *et al.*, 2006), através do ambiente Odyssey (REUSE, 2014). Tanto a notação como o ambiente foram propostos pelo grupo de Reutilização de Software do Programa de Engenharia de Sistemas e Computação (PESC) da COPPE/UFRJ. O modelo de características para o domínio de visualização também foi resultado de um trabalho em conjunto com um aluno de doutorado. A aplicação do modelo também faz parte de uma pesquisa voltada para o uso de visualizações visando apoiar atividades de reutilização de software, por meio da abordagem APPRAiSER (OLIVEIRA, 2014) (SCHOTS, 2014). O modelo é um trabalho em evolução, logo novas características podem ser adicionadas em trabalhos futuros. A Figura 4.2 e a Figura 4.3 apresentam a versão mais recente do modelo de características. A descrição de cada elemento de visualização presente no modelo pode ser consultada em (SCHOTS *et al.*, 2015).

#### 4.2.1.2 Categorias de Características

No domínio de visualização, alguns elementos estão relacionados a uma mesma categoria. Ações do usuário em uma visualização claramente fazem uso de técnicas de interação, o que pode promover mudanças na exibição dos elementos gráficos, customizando a apresentação.

Características que referenciem ações automatizadas ou controladas pelo usuário em uma visualização proposta denotam elementos pertencentes à categoria **Interação**. Representada pela característica conceitual de mesmo nome no modelo, apresentada na Figura 4.2 (a), essa categoria inclui características opcionais relacionadas a técnicas de interação.

Relacionadas não apenas a uma visualização mas a várias composições de visualizações, características da categoria **Apresentação**, apresentadas na Figura 4.2 (b) definem o modo como múltiplas visualizações serão dispostas na representação gráfica. Com apenas duas características funcionais, a seleção de um desses elementos irá apresentar diferentes visualizações de forma sequencial ou simultaneamente.

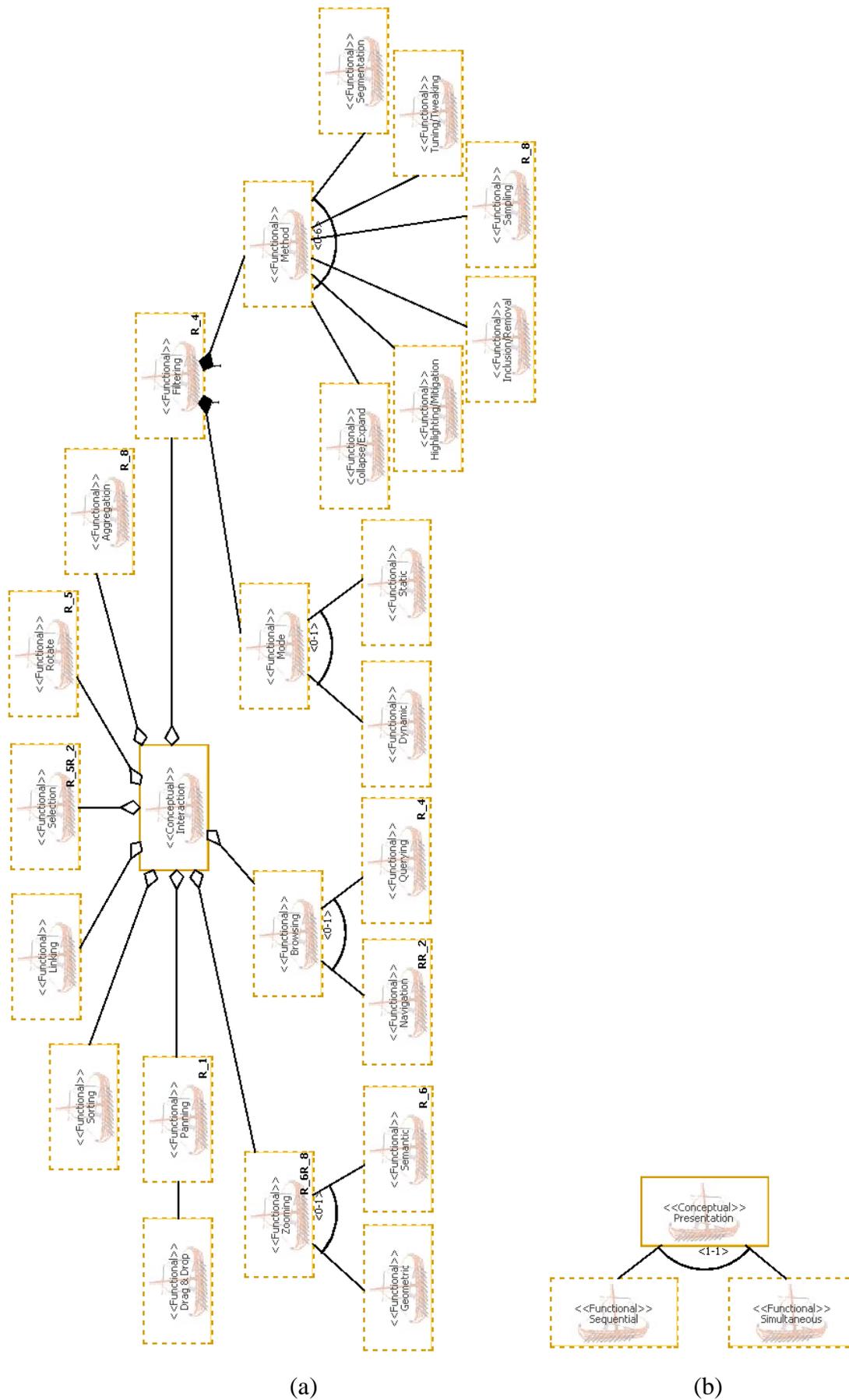


Figura 4.2 - Modelo de características para (a) interação e (b) apresentação

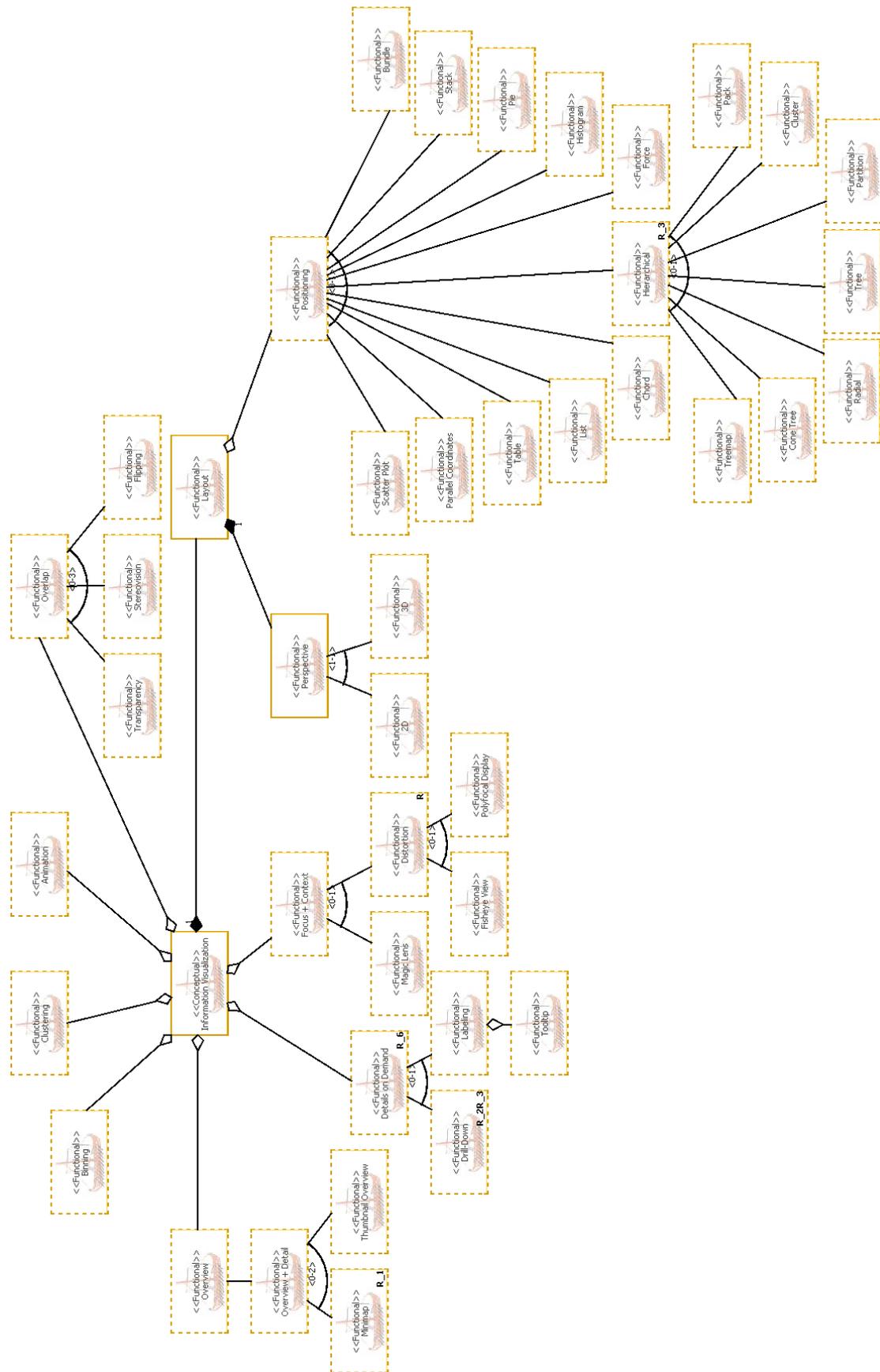


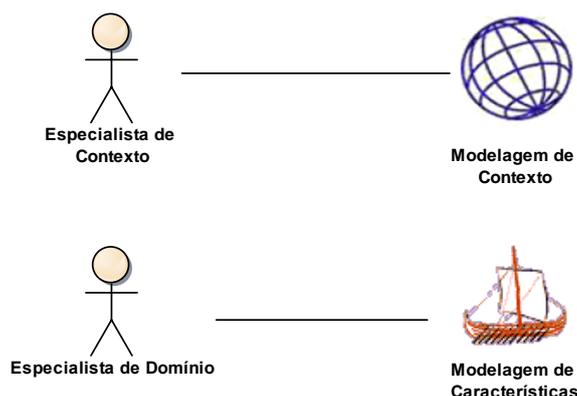
Figura 4.3 - Modelo de características para visualização

Em termos de técnicas que irão adicionar recursos visuais a uma visualização, a categoria **Visualização da Informação** agrupa essas características. Para esta característica conceitual, apresentada na Figura 4.3, uma característica funcional acaba sendo obrigatória, no caso, a relacionada ao leiaute da visualização, uma vez que essa definição é necessária para a organização dos dados na representação (HURLBURT, 1986).

O modelo de características colabora na organização dos elementos do domínio de visualização e possibilita a customização de uma representação gráfica, compondo uma visualização de acordo com requisitos específicos. É importante destacar que a análise de domínio realizada não visa ser uma referência completa de prescrição de características de visualização para qualquer cenário, uma vez que diferentes aspectos deveriam ser contemplados, como mapeamento entre a aplicação de uma técnica de visualização e o conjunto de dados. Uma análise necessária é definir os fatores que irão estabelecer tais requisitos. Nesse caso, adotar uma perspectiva de contexto para as questões envolvidas na escolha de uma visualização pode trazer benefícios ao processo.

#### 4.2.2 Contexto de Visualização

A abordagem distingue dois atores, relevantes para a fase de modelagem, que podem ser classificados como: especialista de domínio e especialista de contexto. Um especialista de domínio é capaz de determinar os pontos mais relevantes de tal domínio, de forma a destacá-los em um modelo de características. Um especialista de contexto, de forma equivalente, deve ser capaz de apontar as informações mais importantes, cujos valores devem ser observados de forma a se verificar o estado atual do contexto de um cenário. A Figura 4.4 explicita as responsabilidades dos atores identificados. É importante destacar que nada impede que ambos os papéis sejam realizados por um único especialista.



**Figura 4.4** - Atores da abordagem

Como o objetivo final é produzir uma visualização a partir do contexto, deve-se ter um cuidado especial na seleção de informações relevantes. Não apenas propriedades relacionadas a atividades ou a pessoas em uma organização devem ser listadas, mas também é necessário avaliar os aspectos que influenciam na escolha da visualização.

Conforme acontece na engenharia de domínio, com a formalização de um modelo de características de forma a possibilitar uma abordagem de linha de produtos (LEE *et al.*, 2002), as informações relevantes para um contexto devem ser concretizadas em um modelo de contexto.

#### 4.2.2.1 Modelo de Contexto

Alguns aspectos devem ser considerados ao se buscar informações candidatas para o contexto de um cenário sendo avaliado. Como o foco está voltado para visualizações de informação, deve-se realizar um levantamento para verificar quais fatores exercem maior influência nas propriedades de uma representação gráfica. No entanto, em uma aplicação genérica de sensibilidade ao contexto, buscando visualizações como resultados, a escolha dos fatores ocorreria de forma livre, visando apenas destacar situações de contexto que fossem necessárias para monitoramento, sem atenção especial a características próprias de visualizações que pudessem ser incorporadas ao modelo de contexto.

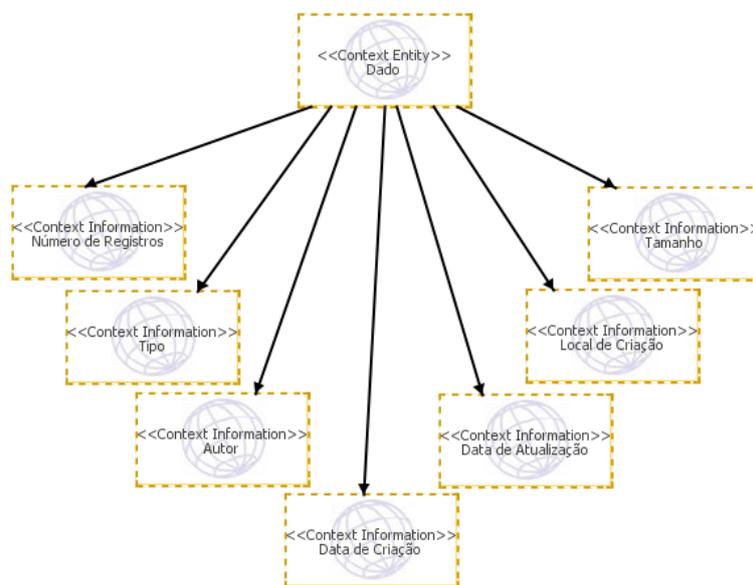
Conforme mencionado no Capítulo 2, com a notação UbiFEX, as duas categorias para características de contexto são Entidade e Informação de Contexto (FERNANDES *et al.*, 2011). Com o objetivo de compor um modelo com as informações relevantes para um contexto, elementos importantes para visualizações devem ser classificados nessas categorias.

Um elemento fundamental para o resultado de uma visualização diz respeito aos valores sendo representados, ao conjunto de dados selecionado para visualização. Mudanças nos valores dos dados podem promover mudanças em visualizações. Tais modificações nem sempre podem ser previstas, por isso, visualizações baseadas em eventos precisam de mecanismos de precaução para tratar de possíveis restrições (MÜLLER & SCHUMANN, 2003).

A relevância dos dados para visualizações pode ser confirmada pela existência de diferentes taxonomias para visualização de informação que são baseadas no tipo de dado usado como entrada (CHI & RIEDL, 1998) (MÜLLER & SCHUMANN, 2003). Não apenas o tipo, mas metadados em geral podem ser considerados úteis para avaliar requisitos para uma visualização. Com isso, uma entidade de contexto denominada **Dado**

poderia ser destacada. A Figura 4.5 apresenta um trecho de modelo de contexto referente à entidade Dado. Exemplos de informações de contexto relacionadas a dados podem ser:

- Número de registros,
- Tipo de cada registro,
- Autor de cada registro,
- Data de criação de cada registro,
- Data de atualização de cada registro,
- Local de criação de cada registro, e
- Tamanho de cada registro.



**Figura 4.5** - Trecho de modelo de contexto para entidade Dado

Outro elemento comum e importante em visualizações diz respeito à interação. De forma geral, interações permitem a comunicação entre usuários e sistemas (DIX *et al.*, 2004) e, nas visualizações, permitem mudanças nas representações.

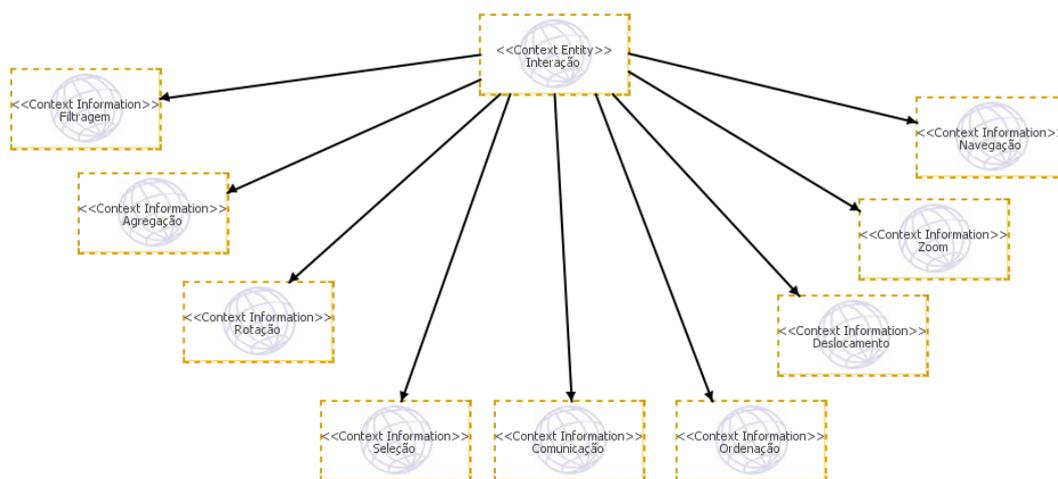
Em termos de contexto, a interação pode ser utilizada como evento, ou seja, na ocorrência de uma interação, uma mudança sensível ao contexto seria registrada. Além disso, interações podem servir como informação relevante em termos históricos. Baseando-se em ações passadas de um usuário, uma análise de contexto pode recomendar determinadas mudanças nas visualizações, promovendo interações automatizadas previamente à ação de um usuário na visualização atual.

Diversas interações foram identificadas na literatura para o campo de visualizações, como as apresentadas em (YI *et al.*, 2007), bem como aquelas explicitadas no Capítulo 3. Além de considerar **Interação** como outra entidade de contexto, em um

nível mais alto na hierarquia de interações proposta no modelo de características da Figura 4.2 (a), é possível listar como informações de contexto os seguintes elementos (SCHOTS *et al.*, 2015):

- Filtragem,
- Agregação,
- Rotação,
- Seleção,
- Comunicação,
- Ordenação,
- Deslocamento,
- Zoom, e
- Navegação.

A Figura 4.6 apresenta um trecho de modelo de contexto referente à entidade Interação.



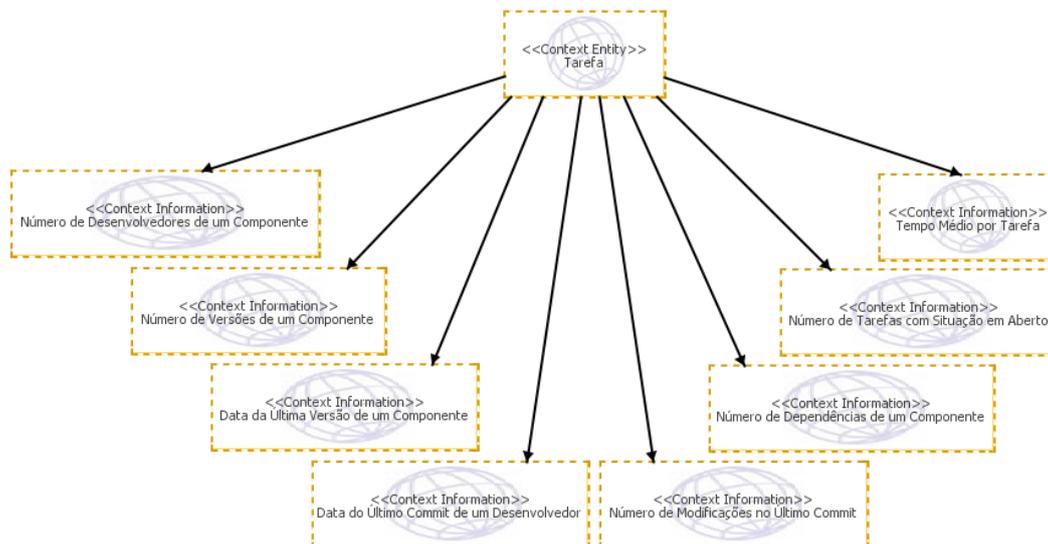
**Figura 4.6** - Trecho de modelo de contexto para entidade Interação

Além de dados e interações, uma visualização precisa de um objetivo, um escopo que serve como motivação para a adoção de uma visualização como mecanismo de suporte. Tal objetivo vem das tarefas que poderão ser apoiadas por representações gráficas.

Visando tratar tarefas como elemento de contexto, é necessário identificar as propriedades que caracterizam tais tarefas. Nesse sentido, SCHULZ *et al.* (2013) propõem dimensões para descrever tarefas, extraindo informações sobre o usuário, como o momento em que a tarefa é realizada, a ordem de uma sequência de tarefas e quem irá realizar tal tarefa.

Com **Tarefa** sendo outra entidade de contexto, as possíveis informações de contexto irão variar de acordo com o domínio das tarefas sendo apoiadas por visualizações. É importante destacar essa customização necessária para a entidade de tarefas, que devem contemplar informações suficientes para os objetivos da visualização. A Figura 4.7 ilustra um trecho de modelo de contexto referente à entidade Tarefa. Para tarefas relativas a desenvolvimento de software, um conjunto de informações de contexto pode ser listado, como por exemplo:

- Número de desenvolvedores de um componente,
- Número de versões de um componente,
- Data da última versão de um componente,
- Data do último *commit* de um desenvolvedor,
- Número de modificações no último *commit*,
- Número de dependências de um componente,
- Número de tarefas com situação em aberto, e
- Tempo médio por tarefa.



**Figura 4.7** - Trecho de modelo de contexto para entidade Tarefa

Por fim, diferentes visualizações podem utilizar o mesmo conjunto de dados e apoiar as mesmas tarefas, mas com representações dispostas de formas diferentes. Mudanças no leiaute de uma visualização visam apoiar diferentes perspectivas, o que pode apoiar *stakeholders* diferentes.

De forma a verificar ou determinar o foco de uma visualização, pode-se considerar **Foco de Representação** como uma entidade de contexto para o modelo pretendido. De

acordo com o foco selecionado, determinado conjunto de layouts são mais adequados. Em (SCHOTS *et al.*, 2015), é proposta uma análise desse mapeamento, entre focos e gráficos mais apropriados de forma a atender aos requisitos do foco.

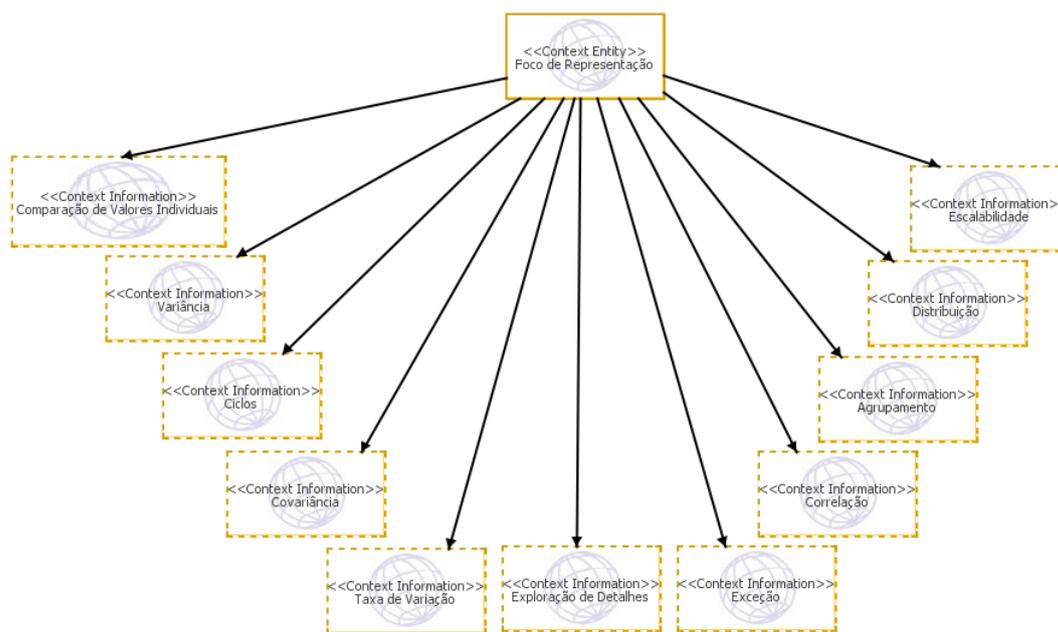
Alguns focos de representação podem ser observados na literatura. FEW (2009) identifica alguns desses focos e aponta elementos de visualização que realçam seus requisitos nas representações gráficas. Comparações permitem observar padrões e interpretar cenários. A capacidade de realçar variações de valores em uma visualização é representada pelo foco voltado para variância. Ciclos são interessantes para representação de registros em períodos cíclicos. O foco de covariância permite apresentar mais de um conjunto de dados e verificar a correlação existente. Uma seleção voltada para taxa de variação permitiria analisar alterações percentuais em um gráfico.

AMAR *et al.* (2005) colaboram apontando outros possíveis componentes para um conjunto de focos de representação. Nesse sentido, visando explorar detalhes, uma visualização permitiria verificar detalhes dos dados exibidos. Para identificar anomalias, um foco voltado para exceções representaria dados de forma a facilitar a interpretação das exceções. Focos específicos também seriam importantes para analisar a correlação entre variáveis, interpretar agrupamentos de dados e explorar distribuições de valores. Por fim, EICK & KARR (2002) destacam a importância da escalabilidade para representações, permitindo visualizar grandes conjuntos de dados.

Com as diferentes possibilidades de análise para uma visualização, um passo inicial importante é selecionar o foco adequado. A partir dos componentes propostos na literatura, a Figura 4.8 apresenta um trecho do modelo de contexto para a entidade de foco de representação, onde algumas informações de contexto são (SCHOTS *et al.*, 2015):

- Comparação de Valores Individuais,
- Variância,
- Ciclos,
- Covariância,
- Taxa de Variação,
- Exploração de Detalhes,
- Exceção,
- Correlação,
- Agrupamento,
- Distribuição, e

- Escalabilidade.



**Figura 4.8** - Trecho de modelo de contexto para entidade Foco de Representação

#### 4.2.2.2 Situações de Contexto

A partir das entidades e informações de contexto especificadas para um cenário, é importante destacar as ocorrências de interesse. Conforme explicado no Capítulo 2, situações de contexto reúnem definições relevantes que devem ser observadas constantemente durante o acompanhamento do contexto.

Levando em consideração que os resultados esperados para essa análise de contexto são visualizações que irão apoiar as atividades sendo desenvolvidas em uma organização, as situações devem envolver os elementos identificados na seção anterior, uma vez que são informações que consideram aspectos de interesse para definir uma visualização. Tais situações devem refletir questões cujas ocorrências se deseja notar para o contexto da organização.

Uma aplicação da abordagem, utilizando os recursos apresentados para condições específicas, pode colaborar na compreensão de sua utilidade. A partir dos conceitos discutidos, um exemplo com situações de contexto para um cenário hipotético é apresentado. Focando em uma área específica do desenvolvimento de software, a reutilização de software serve como inspiração para este exemplo, levando em consideração algumas informações previstas na abordagem APPRAiSER, descritas em (OLIVEIRA, 2014) (SCHOTS, 2014). Dada a dificuldade que certas organizações têm em adotar práticas voltadas para reúso de ativos de software (SHERIF & VINZE, 2003),

a busca por mecanismos de suporte voltados para atividades de reutilização é indicada, como abstrações provenientes de visualizações de software. Com as dificuldades devidas à falta de variadas ferramentas de apoio para reutilização (SCHOTS, 2014), uma abordagem de visualização sensível ao contexto pode ser uma alternativa interessante para uma organização diante de tal problemática.

Em meio a um ambiente de reuso, supõe-se que uma organização tem por objetivo identificar a necessidade de novos ativos reutilizáveis. Dado um repositório de reutilização, onde ativos ficam disponíveis para serem encontrados e adotados em novos projetos, a fonte de dados para o objetivo mencionado estaria no próprio repositório, mais especificamente nos termos buscados nele que não retornaram nenhum resultado. Pretende-se, com isso, identificar os termos mais buscados por desenvolvedores no repositório sem nenhum ativo relacionado, ou seja, talvez seja interessante implementar e disponibilizar ativos relacionados a tais termos, uma vez que haveria demanda comprovada.

Como para um conjunto reduzido de termos buscados no repositório, conforme ilustrado na Figura 4.9 (a), talvez uma visualização não fosse nem necessária ou indicada, uma vez que a frequência das palavras-chave poderia ser explicitada em uma lista ou tabela, uma possível condição para ser considerada determinaria que apenas seriam analisadas ocorrências em que o número de registros buscados ao todo, incluindo com e sem resultados associados, sendo maior do que 30, caso ilustrado na Figura 4.9 (b). Os tipos dos dados sendo utilizados tratam-se de *strings*, já que são os próprios termos buscados. Com isso, duas informações de contexto referentes a entidade de dados com valores atribuídos já podem ser organizadas como a estrutura apresentada na Figura 4.10.

android apache api beans build  
 build-system bytecode cache client  
 cloud codehaus config container database  
 directory eclipse ejb esb framework

(a)

admin agent akka amazon amqp analysis android angularjs annotations ant aop apache api appengine aspect asynchronous authentication aws batch beans benchmark binding bpm build build-system builder  
 bytecode cache cassandra client clojure cloud cloudstack cluster cms code-coverage codecodehaus collections command-line compiler compression concurrency config console container couchdb oss database  
 debugger decoder dependency-injection deployment directory discovery distributed dns doc docker driver dsl eclipse editor ejb embedded encoder esb etconf facebook factory file-system foursquare framework generator git gradle  
 graphais graph graphics groovy gwt hadoop hazelcast hbase hibernate html http i18n ide imap interface io javascript javax jboss jdbc jms jmx jndi jpa jsf json jsp language levels library linkedin logging machine-learning mail  
 math maven memcached metadata metrics mock model module mongodb monitoring native neo4j net netty oauth openid openstack orientdb osgi parser pdf performance persistence play-framework plugin pooling  
 portal portlet profiling protocol query queue rabbitmq redis regexp repository rest riak rmi rpc rss rule-engine sbt scala scheduling scm scripting sdk search security serializer server servlet shell smtp snmp soa soap social  
 socket spdy spring sql sqlite ssh streaming svg swing svn taglib template testing transaction transport twitter typesafe ui validation vertx web web-framework webdav webservice websocket wicket  
 wiki workflow xml xmpp xquery xslt yaml yam zeromq zookeeper

(b)

**Figura 4.9** - Termos buscados em um repositório de reuso

|   |
|---|
| <p>Número de registros &gt; 30</p> <p>Tipo de dados = <i>String</i></p> |
|---|

**Figura 4.10** - Informações referentes a dados com valores

Para organizações com um número pequeno de ativos reutilizáveis disponíveis, considerados como componentes neste caso, pode-se esperar que haja interesse em aumentar a disponibilidade de ativos, identificando novas necessidades para reuso. Nesse sentido, ocorrências em que o número de ativos disponíveis na organização seja menor do que 10 passariam a ser observadas. Visando restringir apenas a casos em que a organização tiver um número mínimo de projetos legados, uma informação explicitando que o número de projetos legados deve ser maior do que 5 poderia ser adotada. Tal condição remete a organizações que tenham um conjunto de projetos já desenvolvidos para justificar uma busca por novos ativos.

Como foi estabelecida uma condição para o número total de termos buscados nas informações referentes a dados, uma restrição em que o número de termos buscados no repositório sem resultados deve ser maior do que 7 impõe que poucos termos buscados sem resultados não seriam suficientes para caracterizar um indicador de novo ativo necessário. A partir dessas novas colocações, três novas informações de contexto referentes a tarefas com seus respectivos valores podem ser vistas na Figura 4.11.

|   |
|---|
| Número de componentes disponíveis < 10                      |
| Número de projetos legados > 5                              |
| Número de termos buscados no repositório sem resultados > 7 |

**Figura 4.11** - Informações referentes a tarefas com valores

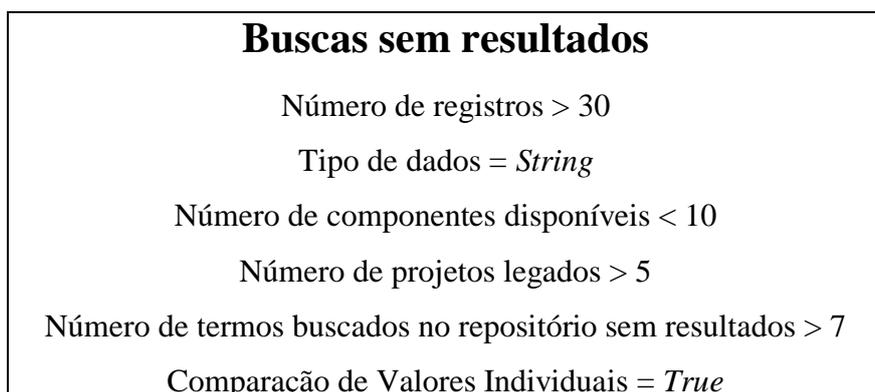
Como pretende-se utilizar o número de ocorrências de palavras-chave buscadas em um repositório de reuso de software, comparações entre os números de ocorrências para cada termo são esperadas de forma a verificar candidatos que possam inspirar a implementação de novos ativos. Dessa forma, a perspectiva de análise em uma visualização para apoiar esse objetivo deveria ser para comparar valores individuais. Com isso, uma nova informação de contexto relacionada a foco de representação é listada na Figura 4.12.

|   |
|---|
| Comparação de Valores Individuais = <i>True</i> |
|---|

**Figura 4.12** - Informação referente a foco de representação com valor

Com todas as informações relevantes para o objetivo definido e valores especificados para cada uma, já seria possível configurar a versão completa da situação de contexto almejada. Como a ocorrência dessa situação está relacionada a buscas de termos sem resultados no repositório de ativos reutilizáveis, pode-se denominar tal

situação como **Buscas sem resultados**. A Figura 4.13 apresenta a versão final dessa situação de contexto.



**Figura 4.13** - Situação de contexto para necessidade de ativos reutilizáveis

Diferentes situações de contexto devem ser especificadas conforme a necessidade de observação para um cenário. Com a ocorrência de cada situação, ou seja, a ocorrência de valores respeitando todas as condições das respectivas informações de contexto, o procedimento seguinte é estabelecer o resultado, quais elementos de visualização irão compor uma representação gráfica.

#### 4.2.2.3 Regras de Contexto

Como apresentado no Capítulo 2, regras de contexto definem quais funcionalidades ou características serão aplicadas ou selecionadas para um produto final, em geral um software. Com a abordagem proposta, buscando estruturar visualizações sensíveis ao contexto, o resultado de uma regra de contexto será um conjunto de técnicas e metáforas de visualização, que irão compor uma visualização apropriada para a dada situação de contexto prevista na regra.

Adaptando o padrão de regras de contextos para uma aplicação no domínio de visualização de informação, um modelo de regra de contexto pode ser verificado na Figura 4.14.

|  |
|--|
| <p>&lt;Situação de Contexto&gt; <b>implica</b> &lt;Característica(s) de Visualização&gt;</p> |
|--|

**Figura 4.14** - Padrão de regra de contexto para visualizações

Visando escolher elementos de visualização que atendam às necessidades do *stakeholder* de visualização, é importante analisar alguns componentes da situação sob foco. Usando como exemplo a situação **Buscas sem resultados** exposta na Figura 4.13, pode-se reconhecer que as condições envolvendo as informações relativas a tarefas (*Número de ativos disponíveis, Número de projetos legados e Número de termos*

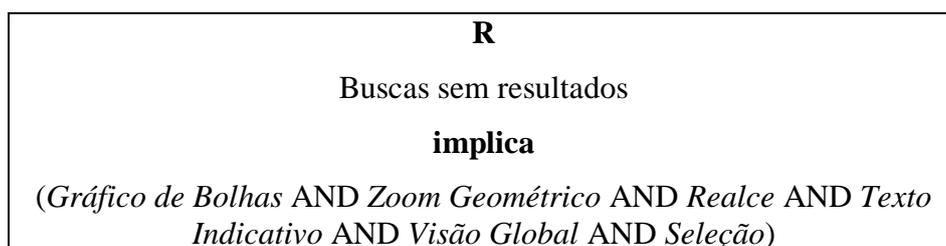
*buscados no repositório sem resultados*) indicam que novos ativos são necessários no repositório da organização, de acordo com o número de buscas em tal repositório.

Com a informação *Número de registros* possuindo valor maior do que 30 e a informação referente ao foco de representação *Comparação de Valores Individuais* sendo verdadeira para o contexto atual, pode-se buscar elementos de visualização apropriados para essas condições. De forma a representar um número elevado de registros (acima de 30) e tendo em conta o mapeamento apresentado em (SCHOTS *et al.*, 2015), em que um gráfico de bolhas seria um leiaute de visualização aceitável para comparação de valores, pode-se confirmar a adoção de tal leiaute para uma representação resultante.

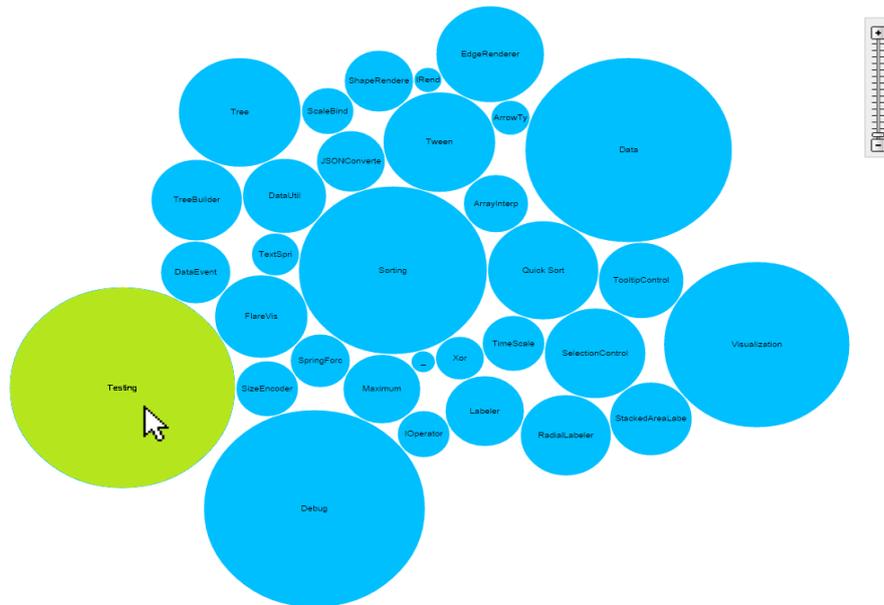
Com um gráfico de bolhas, uma perspectiva de visão global é proporcionada de forma direta, além da possível disponibilidade de mecanismos de seleção e realce de modo intuitivo para cada bolha, representando um termo buscado sem resultado relacionado. Como o tamanho de uma bolha indicaria o número de pesquisas a um mesmo termo, aqueles menos buscados acabarão com bolhas de tamanho reduzido, o que pode dificultar uma desejável identificação de cada bolha, ou seja, de um termo. Neste caso, mecanismos de *zoom* geométrico podem ser empregados, possibilitando ampliar diretamente os elementos sendo visualizados.

Como o tipo de dados é referente aos termos buscados, ou seja, *strings*, é possível disponibilizar textos indicativos exibidos como *tooltips* sobre cada bolha, apresentando a palavra-chave e algum metadado, como o número de buscas para a palavra cuja bolha está sob foco.

A partir da análise realizada para a situação hipotética **Buscas sem resultados**, é possível definir as características de visualização que serão resultantes da regra de contexto. A Figura 4.15 apresenta a regra **R** que inclui todos os elementos discutidos no exemplo. O resultado de tal regra de contexto está representado na Figura 4.16, com a visualização de todos os termos buscados sem resultados associados, destacando o mecanismo de seleção para uma bolha individual.



**Figura 4.15** - Regra de contexto para identificar a necessidade de novos ativos

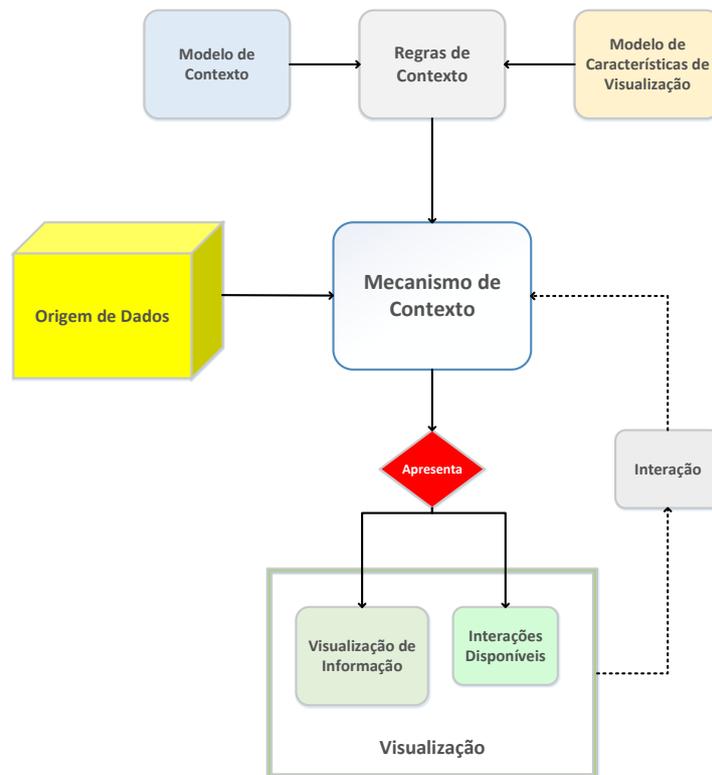


**Figura 4.16** - Visualização para identificar necessidade de novos ativos

Regras de contexto nesta abordagem são os elementos responsáveis pela integração direta entre conceitos relacionados à sensibilidade ao contexto e visualização de informação. Como, para cada situação, um conjunto específico de características de visualização pode ser designado como resultante, é importante uma seleção adequada das técnicas e metáforas necessárias para uma representação gráfica apropriada. Nesse sentido, os elementos mencionados no resultado de regras de contexto podem ser interpretados como requisitos de visualização para representar situações.

### 4.2.3 Visualização Orientada ao Contexto

Com as características de visualização disponíveis para seleção e os diferentes elementos de contexto especificados, como entidades, informações, situações e regras, já é possível unir os conceitos discutidos. Nesse sentido, a abordagem CAVE, buscando apresentar visualizações orientadas ao contexto, pode ser sintetizada pelo esquema exibido na Figura 4.17.



**Figura 4.17** - Visão conceitual da abordagem CAVE

Através do esquema, verifica-se a utilização dos modelos de contexto e de características de visualização para a formulação das regras de contexto. Tais regras serão analisadas pelo mecanismo de contexto, que gerencia a ocorrência das situações e efetivação das regras por meio de consultas aos valores dos dados na fonte selecionada para a abordagem. Na efetivação de uma regra, o mecanismo de gerenciamento de contexto irá selecionar as características de tal regra e então apresentar a visualização composta por tais elementos, como características de visualização, incluindo mecanismos de interação disponíveis. Além dos valores dos dados, se uma situação estiver prevendo a utilização de uma interação como condição para sua ocorrência, interações realizadas sobre uma visualização também podem promover mudanças na representação gráfica a ser apresentada ou atualizada ao final.

### 4.3 Análise Comparativa

De forma a verificar os diferenciais entre o presente estudo e outros trabalhos propostos na literatura, é importante realizar uma comparação, mencionando características comuns ou divergentes. Por isto, nesta seção, é feita uma análise comparativa entre os trabalhos semelhantes identificados e a presente abordagem.

A abordagem proposta por BECK *et al.* (2013) se assemelha à abordagem CAVE por utilizar classificações referentes a dados e tarefas em uma aplicação para escolher visualizações. No entanto, alguns pontos diferem do objetivo da abordagem CAVE.

A abordagem de BECK *et al.* (2013) se propõe a apoiar especialistas em visualização a selecionar técnicas de visualização para determinados perfis de aplicação, além de considerar apenas visualizações cujo leiaute é baseado em grafos. Por sua vez, a abordagem CAVE busca auxiliar diferentes *stakeholders* com a apresentação de visualizações, após o modelo de contexto ser estabelecido por especialistas. Além disso, não há uma limitação para o tipo de visualizações a serem propostas. No entanto, as abordagens podem ser complementares, em uma situação em que um especialista de contexto utilize a abordagem de BECK *et al.* (2013) para definir quais características de visualização serão selecionadas para certas regras de contexto.

O trabalho de VOIGT *et al.* (2012) apresenta uma abordagem para recomendação de visualizações baseada em conhecimento e sensível ao contexto. Uma ontologia denominada VISO é adotada para descrever componentes de visualização e fontes de dados, incluindo sete módulos referentes ao domínio de visualização, como: dados, gráfico, atividade, usuário, sistema, domínio e fatos. Com tal ontologia, além dos componentes de visualização, conhecimentos sobre o contexto também são detalhados. Um algoritmo de recomendação é empregado para listar possíveis mapeamentos para os dados selecionados, considerando o conteúdo disposto na ontologia.

Tal abordagem se assemelha à CAVE, remetendo-se a diferentes entidades de contexto para prover visualizações apropriadas. No entanto, apesar de verificar módulos como atividade, domínio e fatos, tal análise acaba sendo centrada na visualização, com verificações como atividades relacionadas a ações de interação na representação gráfica, tipos de visualizações mais apropriadas para determinado domínio ou conhecimentos consensuais na comunidade de visualização, como afirmações explicitando que posições são mais precisas para representar quantidades do que cores. Nesse sentido, a entidade Tarefa existente na CAVE acaba não tendo uma equivalência na ontologia VISO.

Levando em consideração um aspecto de contexto, o trabalho de SCHULZ *et al.* (2013) menciona o uso de tarefas de forma a apontar visualizações recomendadas. Nesse aspecto, tal abordagem se relaciona com a CAVE, porém sem abranger dimensões mais amplas como contexto.

Também focando em um aspecto de contexto, como dados, a abordagem de CHI (2000) permite analisar requisitos para a aplicação de diferentes técnicas de visualização.

Nesse sentido, é possível fazer um paralelo com as regras de contexto existentes na abordagem CAVE, em que as características de visualização presentes em uma regra podem ser consideradas como requisitos para representação de uma situação de contexto.

Comparando com a proposta do modelo de características de visualização, a abordagem de TORY & MOLLER (2004) utiliza como critérios aspectos relativos aos dados a serem representados para estruturar uma taxonomia de visualização. Com isso, é possível destacar a diferença para o insumo utilizado pela abordagem CAVE, que aplica um modelo de características baseado em uma análise de domínio.

A partir dos estudos relacionados, pode-se perceber que nenhum dos mecanismos identificados adota a mesma perspectiva da abordagem CAVE, reunindo sensibilidade ao contexto para apresentar visualizações, em que o contexto considerado reúne dimensões amplas como dados, interações e tarefas. É importante ressaltar que o foco de cada trabalho difere do foco da abordagem CAVE, o que explica as diferenças adotadas por cada abordagem, ou, pelo menos, por parte de cada abordagem.

#### **4.4 Considerações Finais**

Visualizações proveem abstrações úteis para apoiar e até mesmo permitir a realização de diferentes tarefas. Um dos desafios que permeia essa alternativa diz respeito à dúvida de quando uma visualização pode ser necessária e, além disso, qual visualização é a mais adequada em cada situação.

De forma a abordar esses obstáculos, visualizações sensíveis ao contexto reúnem conceitos de diferentes áreas visando obter resultados mais adequados. Com visualizações apropriadas para cada situação, considera-se que a realização das demais tarefas, apoiadas por tais visualizações, também pode ser impactada positivamente.

A fim de aplicar a abordagem em situações comuns e verificar vantagens e limitações, um protótipo permite uma análise mais apropriada. A implementação da abordagem em uma plataforma é apresentada no capítulo a seguir.

# CAPÍTULO 5 – IMPLEMENTAÇÃO DA ABORDAGEM

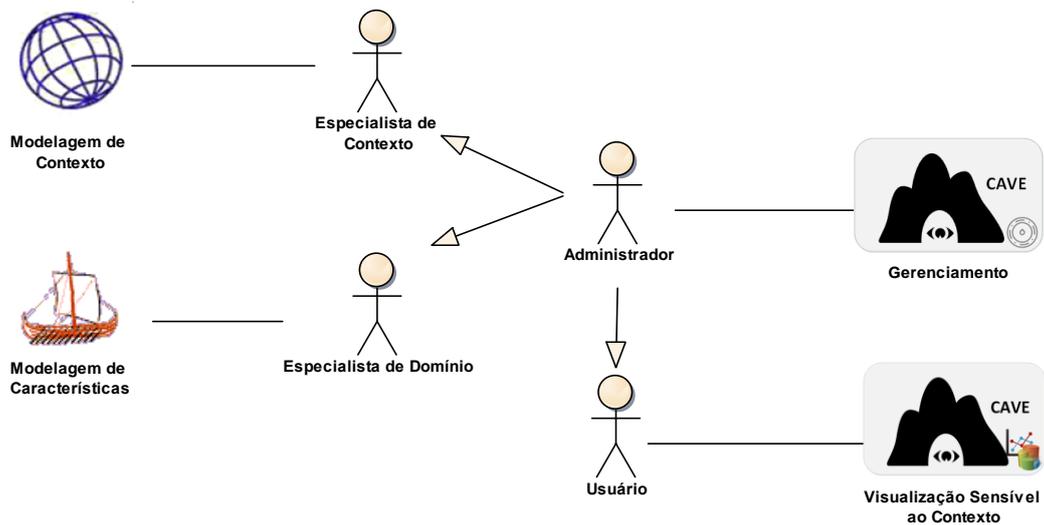
## 5.1 Introdução

Com a abordagem detalhada, é importante descrever e desenvolver meios para sua real adoção. Nesse sentido, a implementação da abordagem permite verificar a viabilidade da abordagem bem como a constatação da necessidade de novos requisitos ou de limitações não previstas. Com um protótipo desenvolvido, pode-se seguir todos os passos de utilização permitidos, além de analisar se sua adoção poderia ocorrer de forma intuitiva.

Além desta seção introdutória, este capítulo apresenta a arquitetura da implementação na Seção 5.2. As alternativas de obtenção de dados a serem utilizados na CAVE são destacadas na Seção 5.3. O mecanismo de monitoramento de contexto é detalhado na Seção 5.4. A Seção 5.5 explica como ocorre a composição de diferentes elementos em uma visualização resultante. Já a Seção 5.6 traz exemplos de utilização das diferentes funcionalidades disponíveis no protótipo. Considerações finais sobre a implementação são apresentadas na Seção 5.7.

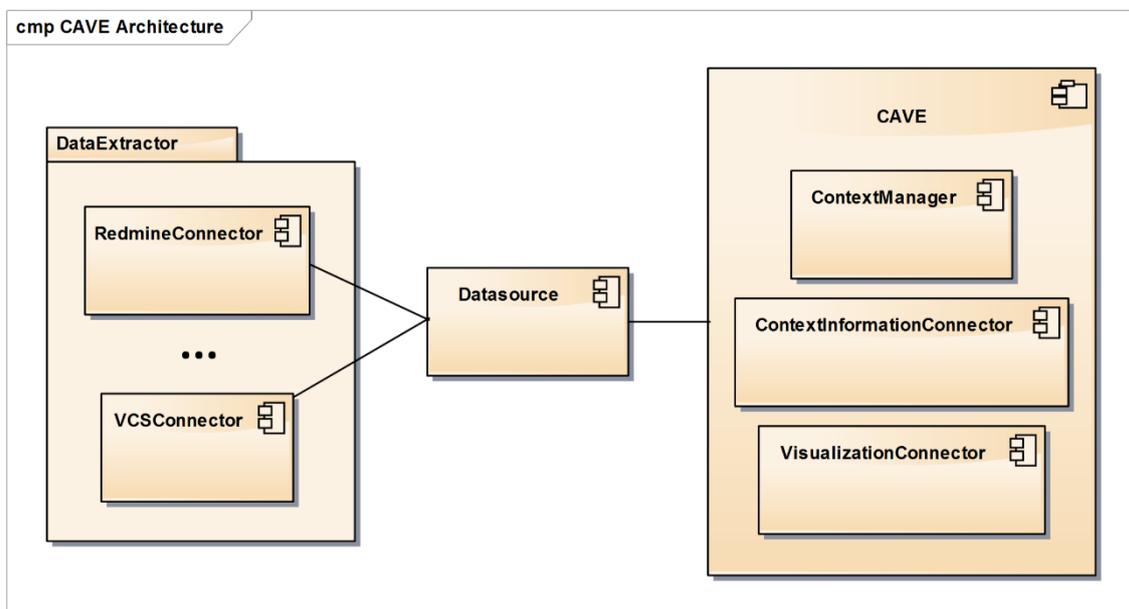
## 5.2 Arquitetura

No que diz respeito à implementação da abordagem de visualizações sensíveis ao contexto, optou-se pelo desenvolvimento de uma aplicação *web* de mesmo nome, CAVE, responsável por avaliar situações de contexto e apresentar visualizações seguindo regras de contexto. Para tal aplicação, estão previstos dois papéis específicos: um administrador, responsável pelo gerenciamento do modelo de contexto e fornecimento de acesso a usuários, e um usuário comum, que, sendo cadastrado no sistema, poderá acompanhar a apresentação de visualizações orientadas ao contexto ou selecionar visualizações diretamente. Remetendo a papéis mencionados para a presente abordagem, não há obrigação que o administrador ocupe outros papéis, porém é interessante que este possa realizar as atividades relativas aos especialistas de contexto e de domínio. Qualquer alteração nos modelos só poderá ser refletida na plataforma ao se submeter o modelo de contexto através do painel de gerenciamento. A Figura 5.1 ilustra o relacionamento indicado entre papéis para a abordagem como um todo e na plataforma.



**Figura 5.1** - Papéis previstos na plataforma CAVE

A aplicação é composta por pelo menos seis componentes, sendo um deles a base de dados utilizada como fonte. A fim de mostrar a sua estrutura, o diagrama de componentes para a plataforma CAVE pode ser verificado na Figura 5.2.



**Figura 5.2** - Diagrama de componentes para a plataforma CAVE

### 5.3 Obtenção dos Dados

Um módulo responsável pela extração de dados se fez necessário de forma a obter os valores dos dados, referentes a sistemas de gerenciamento de tarefas (*issue tracker*) e de controle de versão, adotados nesta etapa da pesquisa, mas não limitando-se a apenas esses tipos de sistemas. Diferentes sistemas podem ser utilizados como fontes de dados, sendo necessário apenas implementar um conector compatível. O componente

RedmineConnector, uma extensão do conector homônimo apresentado em (QUEIROZ *et al.*, 2012), implementa a extração de dados do sistema Redmine, com seus recursos de gerenciamento de tarefas. Os dados extraídos incluem características de projetos, tarefas e usuários envolvidos no desenvolvimento das atividades de tais projetos.

Com relação aos sistemas de controle de versão, o componente VCSCConnector foi criado com base no conector EvolTrack-VCS (SILVA, 2012), que foi utilizado de forma desacoplada da abordagem PREViA (OLIVEIRA, 2011). O conector usa a API Maven SCM (APACHE SOFTWARE FOUNDATION, 2014), oferecendo suporte a diferentes tipos de repositório, bastando apenas a sua configuração de acesso, como a identificação do repositório objetivado. São coletados dados referentes a revisões e arquivos presentes em *commits*.

Tais conectores extraem dados referentes a projetos de software e alimentam uma base de dados comum. Tal base serve de fonte de dados para a aplicação CAVE.

## 5.4 Monitoramento do Contexto

A plataforma CAVE, composta por três módulos, foi desenvolvida de forma a verificar valores de dados de interesse ao modelo de contexto sendo utilizado e apresentar visualizações apropriadas, conforme as regras de contexto inseridas no modelo.

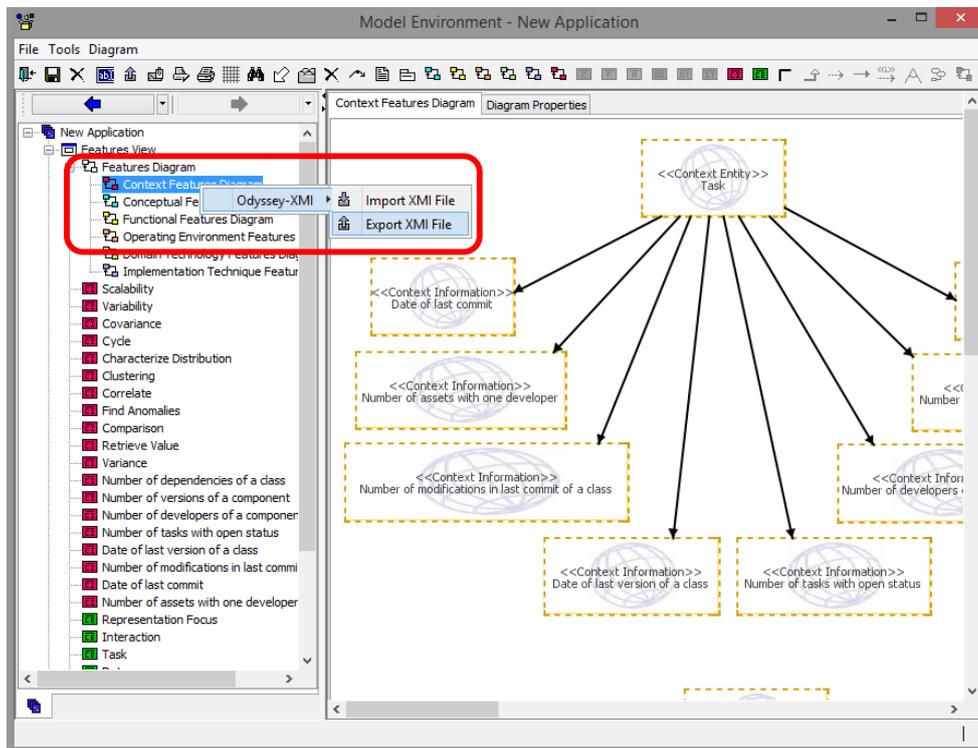
O gerenciador de contexto (*ContextManager*) verifica as informações, situações e regras do modelo de contexto usado como entrada para a plataforma e, de acordo com os dados presentes na fonte de dados, aponta as situações em ocorrência em determinado momento.

O conector de informações de contexto (*ContextInformationConnector*) apresenta as informações de contexto suportadas pela plataforma, com a consulta responsável por determinar o valor da informação. Todas as informações suportadas implementam a interface *Information*. Esse componente é extensível, sendo possível adicionar novas informações de contexto que podem ser adicionadas em modelos de contexto.

Por fim, o conector de visualização (*VisualizationConnector*) gerencia a seleção dos elementos de visualização presentes nas regras de contexto em ocorrência, para posterior apresentação na plataforma, com a visualização completa. Como as visualizações são compostas utilizando a biblioteca D3.js (BOSTOCK, 2013), novos elementos a serem disponibilizados para regras de visualização devem ser implementados em JavaScript usando a biblioteca D3.js.

Uma vez que o modelo de contexto é utilizado como entrada para o processo de monitoramento do contexto, é necessário preparar o arquivo para ser importado na plataforma CAVE. Dessa forma, no ambiente Odyssey, na visão de características, é importante criar o diagrama de contexto, com as informações e entidades, as situações e regras.

Ao final da modelagem, deve-se exportar os diferentes elementos em um formato suportado pela plataforma CAVE. Nesse sentido, de forma a viabilizar um formato padrão, optou-se por expandir o *plugin* Odyssey-XMI, existente no Odyssey para exportar modelos em arquivos XMI (*XML Metadata Interchange*), adotado de forma geral para integração de sistemas. A modificação no *plugin* envolveu a adição de suporte à exportação de modelos de contexto, com diagramas, situações e regras de contexto. A Figura 5.3 ilustra essa atividade de geração do modelo de contexto para exportação.



**Figura 5.3** - Exportação do modelo de contexto em formato XMI

Os arquivos em formato XMI devem ser importados na plataforma CAVE. Por isso, o administrador da instância da plataforma deve realizar esse procedimento de importação na plataforma. A Figura 5.4 apresenta um trecho do arquivo XMI gerado e a Figura 5.5 apresenta a parte da tela após a importação de um modelo de contexto.

```

<?xml version = '1.0' encoding = 'windows-1252' ?>
<XMI xmi.version = '1.2' xmlns:UML = 'org.omg.xmi.namespace.UML' timestamp = 'Sun Jan 18 20:40:04 BRST 2015'>
  <XMI.header>
    <XMI.documentation>
      <XMI.exporter>Netbeans XMI Writer</XMI.exporter>
      <XMI.exporterVersion>1.0</XMI.exporterVersion>
    </XMI.documentation>
  </XMI.header>
  <XMI.content>
    <UML:TagDefinition xmi.id = 'a1' name = 'Priority' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a2' name = 'Category' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a3' name = 'Optionality' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a4' name = 'Description' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a5' name = 'Type' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a6' name = 'Persistence' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a7' name = 'Composition' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a8' name = 'Acquisition' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a9' name = 'Values' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a10' name = 'Interval1' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a11' name = 'Interval2' isSpecification = 'false' />
    <UML:TagDefinition xmi.id = 'a12' name = 'Step' isSpecification = 'false' />
    <UML:Node xmi.id = 'a13' name = 'Context Features Diagram' isSpecification = 'false'
      isRoot = 'false' isLeaf = 'false' isAbstract = 'false'>
      <UML:ModelElement.taggedValue>
        <UML:TaggedValue xmi.id = 'a14' name = 'isDiagram' isSpecification = 'false'>
          <UML:TaggedValue.dataValue>true</UML:TaggedValue.dataValue>
        </UML:TaggedValue>
        <UML:TaggedValue xmi.id = 'a15' name = 'contextNode' isSpecification = 'false'>
          <UML:TaggedValue.dataValue>Date of last commit</UML:TaggedValue.dataValue>
          <UML:ModelElement.taggedValue>
            <UML:TaggedValue xmi.id = 'a16' isSpecification = 'false'>
              <UML:TaggedValue.dataValue>High</UML:TaggedValue.dataValue>
              <UML:TaggedValue.type>
                <UML:TagDefinition xmi.idref = 'a1' />
              </UML:TaggedValue.type>
            </UML:TaggedValue>
            <UML:TaggedValue xmi.id = 'a17' isSpecification = 'false'>
              <UML:TaggedValue.dataValue>Context Information</UML:TaggedValue.dataValue>
              <UML:TaggedValue.type>
                <UML:TagDefinition xmi.idref = 'a2' />
              </UML:TaggedValue.type>
            </UML:TaggedValue>
          </UML:ModelElement.taggedValue>
        </UML:ModelElement.taggedValue>
      </UML:Node>
  </XMI.content>
</XMI>

```

Figura 5.4 - Trecho de arquivo XMI gerado na exportação do modelo de contexto

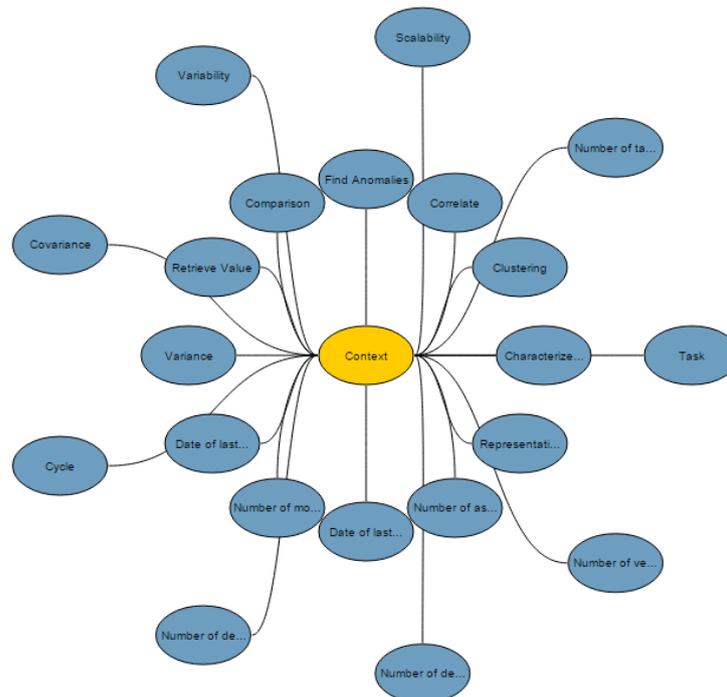


Figura 5.5 - Representação visual do modelo de contexto importado, gerada pela CAVE

De forma a mapear as informações de contexto presentes no modelo importado com as informações suportadas pela plataforma, o administrador da instância deve editar o modelo. Como não necessariamente os nomes das informações de contexto presentes no modelo são idênticos aos nomes das informações suportadas, um formulário para fazer as associações entre informações do modelo e informações suportadas é apresentado para o administrador. A Figura 5.6 apresenta a tela em que tal mapeamento pode ser realizado.

The screenshot shows the CAVE web application interface. At the top, there is a navigation bar with the CAVE logo and a user greeting: "Welcome back, Renan" with a "logout" button. Below the navigation bar is a menu with options: "New Model", "Edit Model", "Edit Definitions", "Edit Rules", and "Information Access". The main content area is titled "Map modeled information with supported information:" and contains a table with two columns: "Modeled Information" and "Supported Information". The table lists various modeled information items and their corresponding supported information items. A "Next" button is located at the bottom right of the table. At the bottom of the page, there is a footer with links for "CAVE About Contact", "Services Visualizations Projects", and "Documentation Help", along with "Terms of use Privacy" and a copyright notice: "© 2014 COPPE/UFRJ. All rights reserved".

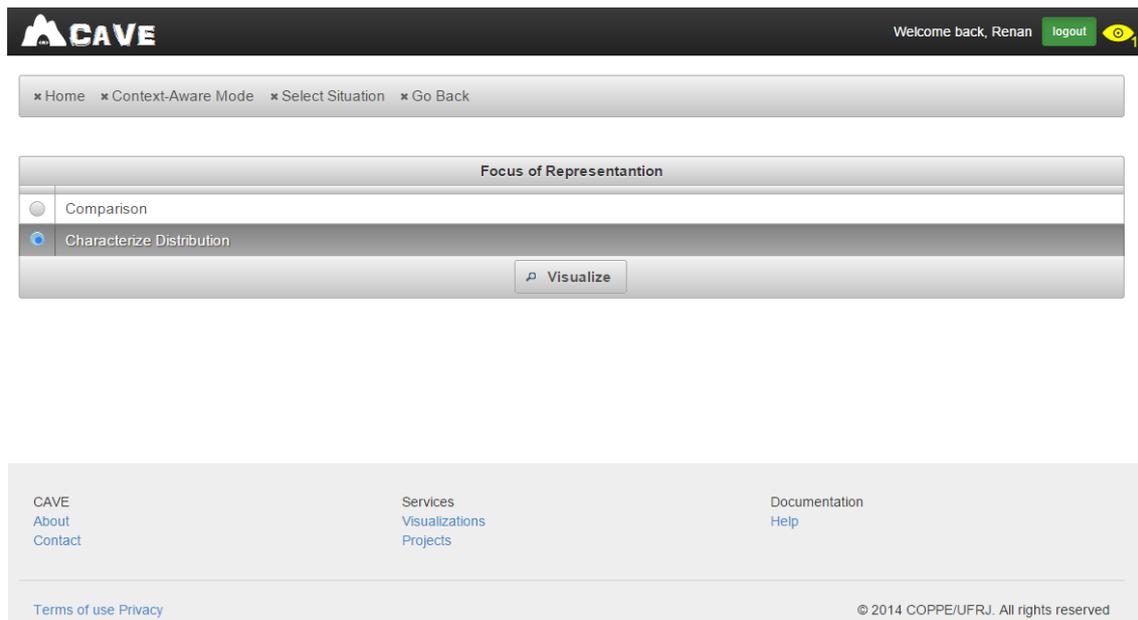
| Modeled Information                               | Supported Information                              |   |
|---|--|---|
| Representation Focus                              |  | / |
| Number of assets with one developer               | Number of Assets with One Developer Within a Month | / |
| Date of last commit                               | Date of an User's Last Commit                      | / |
| Number of modifications in last commit of a class | Number of Modifications in Last Commit             | / |
| Date of last version of a class                   | Date of an User's Last Commit                      | / |
| Variance  |  | / |
| Retrieve Value                                    |  | / |
| Comparison  |  | / |
| Find Anomalies                                    |  | / |
| Correlate   |  | / |
| Clustering  |  | / |
| Characterize Distribution                         |  | / |
| Cycle   |  | / |
| Covariance  |  | / |
| Variability                                       |  | / |
| Scalability                                       |  | / |
| Number of tasks with open status                  | Number of Tasks With Opened Status                 | / |
| Task  |  | / |
| Number of versions of a component                 | Number of Versions of a Component                  | / |
| Number of developers of a component               | Number of Developers of a Component                | / |
| Number of dependencies of a class                 | Number of Dependencies of a Component              | / |

**Figura 5.6** - Tela para realizar o mapeamento entre informações de contexto importadas e informações suportadas

Opções semelhantes de edição do modelo de contexto também são oferecidas para outros elementos. O administrador pode editar os valores e os símbolos de expressão para cada informação em uma situação de contexto. Além disso, é possível também editar as regras de contexto, indicando os elementos de visualização que serão selecionados na ocorrência de uma situação. Visando dar suporte a projetos com múltiplos *stakeholders* e até mesmo considerando a existência de informações sensíveis em termos de segurança,

é oferecido ao administrador um mecanismo de concessão de acesso às situações de contexto. Nesse sentido, pode-se indicar quais membros poderão obter visualizações referentes a determinadas situações de contexto.

Na ocorrência de uma situação e efetivação de uma regra de contexto, antes da apresentação de uma visualização propriamente dita, o *stakeholder* visualizador deve selecionar o foco de representação desejado. A Figura 5.7 apresenta a tela de seleção do foco. Após tal escolha, a visualização, composta pelos elementos de visualização existentes na regra de contexto efetivada, é finalmente apresentada. Tal apresentação pode ser vista na Figura 5.8.



**Figura 5.7** - Tela com as opções de foco de representação para uma visualização

Além do modo sensível ao contexto, o usuário da plataforma CAVE também dispõe de um mecanismo de seleção direta da situação de contexto que se deseja apoiar por meio de visualizações. Com isso, é possível selecionar qualquer uma das situações de contexto que o usuário tenha acesso para obter como retorno a visualização proposta na regra de contexto com tal situação. É importante ressaltar que tal visualização também será apresentada somente após a seleção do foco de representação.

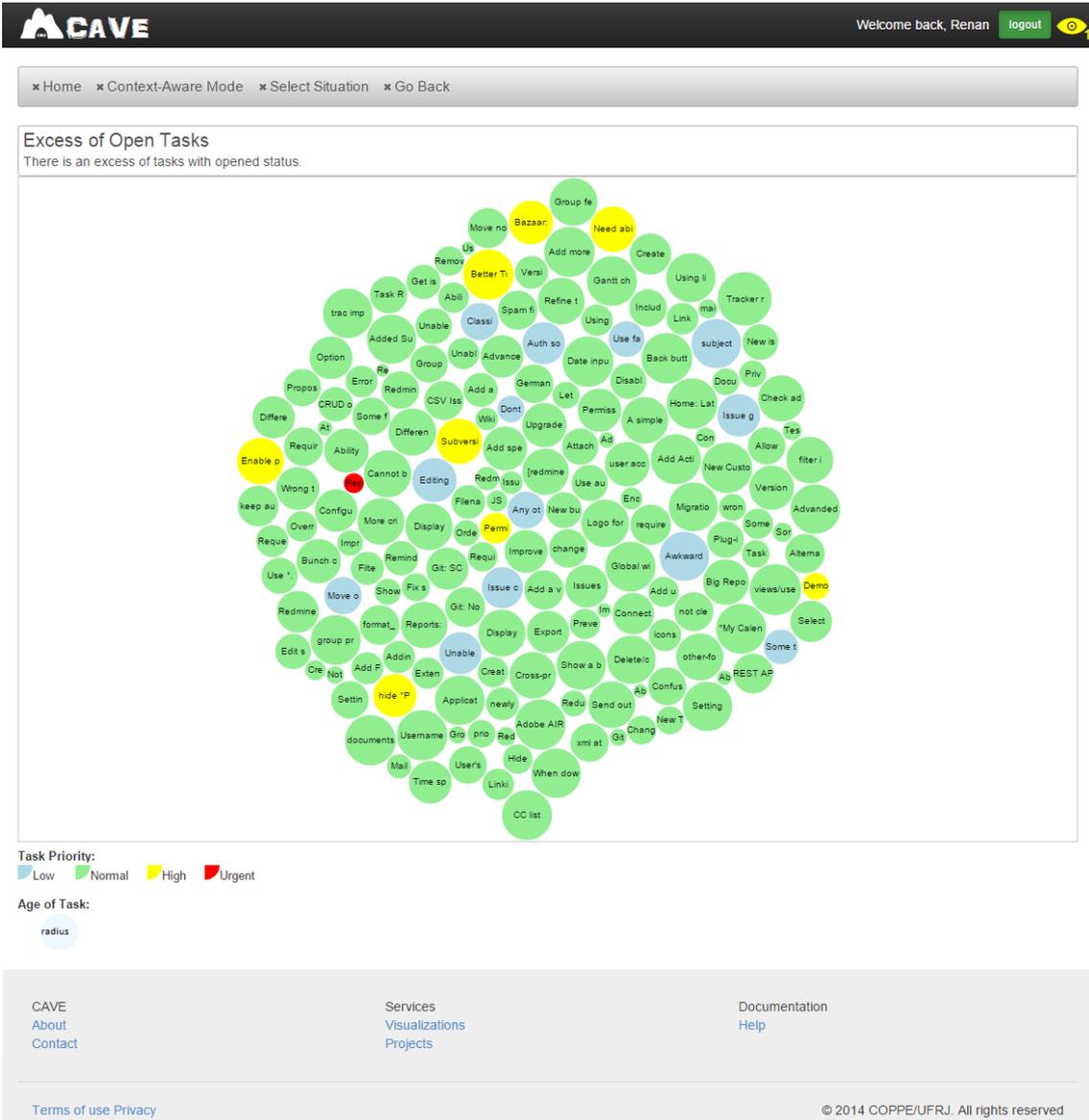
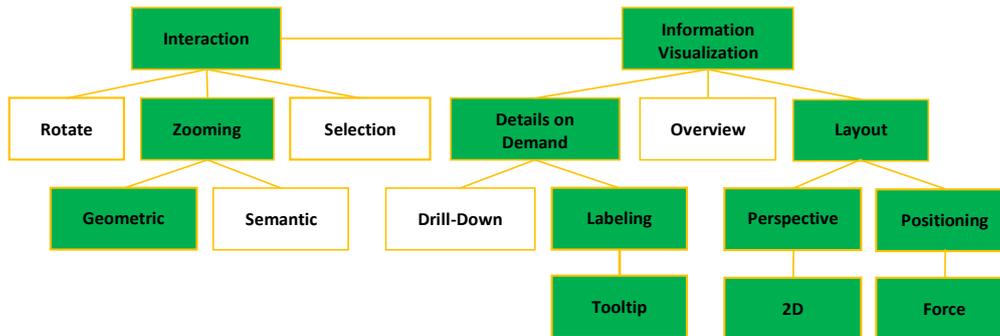


Figura 5.8 - Tela com a visualização orientada ao contexto

## 5.5 Composição de Visualizações

Uma vez que as regras de contexto apenas listam os elementos de visualização que serão selecionados em sua efetivação, a composição da visualização a ser apresentada com tais elementos fica a cargo da implementação com a biblioteca D3.js. De forma a possibilitar a combinação de diferentes técnicas de visualização com variados layouts, foram criadas funções em JavaScript para a adição de cada elemento de visualização na representação gráfica final.

Visando exemplificar essa composição de elementos, a Figura 5.9 apresenta um diagrama de características, na forma de um restrito trecho do modelo de características proposto na Figura 4.2 e na Figura 4.3, com a indicação de características selecionadas, de forma a proceder com o recorte do modelo e a composição da visualização.



**Figura 5.9** - Recorte do diagrama de características de visualização

Com isso, a visualização a ser apresentada deverá prover uma representação com leiaute de grafo (*force layout*) bidimensional, textos indicativos como *tooltips* e um mecanismo de *zoom* geométrico. Nesse caso, três funções devem ser executadas para obter o resultado desejado. A Figura 5.10 apresenta o trecho de código referente à função para adicionar uma metáfora de visualização baseada em grafo. A Figura 5.11 exhibe o trecho da função para adicionar textos indicativos em cada nó de um grafo na visualização. Já a Figura 5.12 apresenta a função destinada a adicionar um mecanismo de *zoom* geométrico na representação gráfica. Por fim, a Figura 5.13 exhibe a visualização resultante da execução das funções apresentadas.

```

1 function addGraph(e, source) {
2   //Initialize a default force layout, using the nodes and edges in dataset
3   var force = d3.layout.force()
4     .nodes(source.nodes)
5     .links(source.edges)
6     .size([w, h])
7     .linkDistance([50])
8     .charge([-100])
9     .start();
10
11  //Create edges as lines
12  edges = e.selectAll("line")
13    .data(source.edges)
14    .enter()
15    .append("line")
16    .style("stroke", "#ccc")
17    .style("stroke-width", 1);
18
19  //Create nodes as circles
20  nodes = e.selectAll("circle")
21    .data(source.nodes)
22    .enter()
23    .append("circle")
24    .attr("r", 10)
25    .style("fill", function(d, i) {
26      return colors(i);
27    })
28    .call(force.drag);
29
30  //Every time the simulation "ticks", this will be called
31  force.on("tick", function() {
32    edges.attr("x1", function(d) { return d.source.x; })
33      .attr("y1", function(d) { return d.source.y; })
34      .attr("x2", function(d) { return d.target.x; })
35      .attr("y2", function(d) { return d.target.y; });
36
37    nodes.attr("cx", function(d) { return d.x; })
38      .attr("cy", function(d) { return d.y; });
39  });
40 }

```

**Figura 5.10** - Função para adicionar leiaute de grafo em uma visualização

```

1 function addTooltip(e, arg) {
2   e.append("title").text(function(d) {return eval(arg);})
3 }

```

**Figura 5.11** - Função para adicionar texto indicativo em uma visualização

```

1 function zooming() {
2   svg.attr("transform", "translate(" + d3.event.translate + ")scale(" + d3.event.scale + ")");
3   nodes.attr("transform", "translate(" + d3.event.translate + ")scale(" + d3.event.scale + ")");
4   edges.attr("transform", "translate(" + d3.event.translate + ")scale(" + d3.event.scale + ")");
5 }
6
7 function addZooming(e) {
8   e.call(d3.behavior.zoom().scaleExtent([1, 8]).on("zoom", zooming));
9 }

```

**Figura 5.12** - Função para adicionar mecanismo de *zoom* geométrico em uma visualização



**Figura 5.13** - Visualização composta por diferentes elementos

A disponibilidade de elementos de visualização pode ser aumentada implementando novas funções para sua adição nas representações gráficas. Considera-se essa abordagem flexível o suficiente para customizar visualizações e permitir sua extensão com o desenvolvimento de novos elementos para representação gráfica.

## 5.6 Exemplos de Utilização

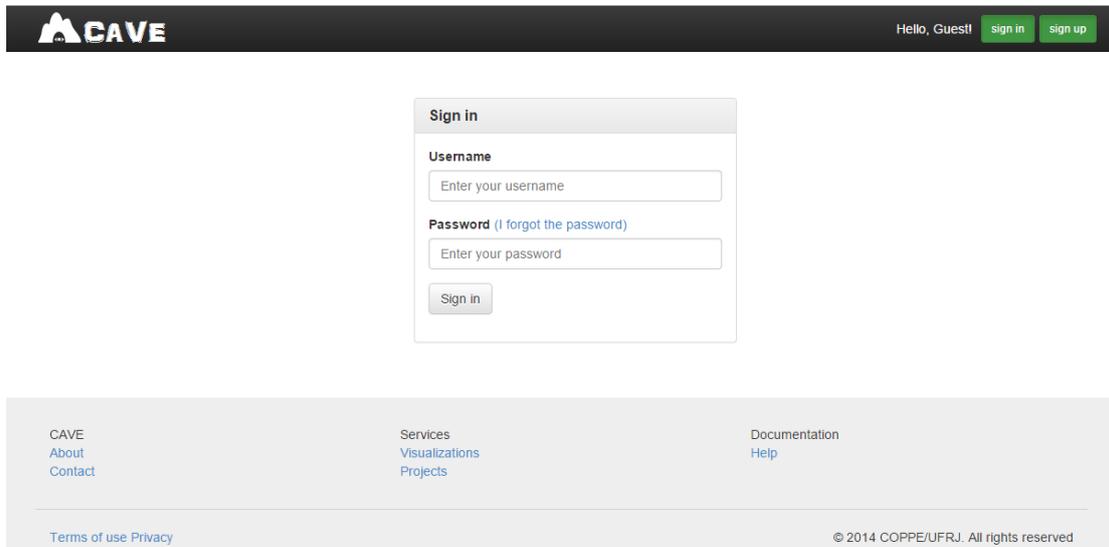
Conforme mencionado na Seção 5.2, a plataforma dispõe de dois perfis de usuários: um administrador e um usuário comum. A fim de compreender todas as formas de utilização das funcionalidades, nesta seção serão apresentados todos os passos que ambos os perfis de usuários podem seguir.

De forma a realizar a autenticação ou cadastro, o usuário deve acessar a página inicial da plataforma e escolher umas das opções na barra superior exibida. A Figura 5.14 apresenta esta barra superior. A página para autenticação de um usuário já cadastrado é

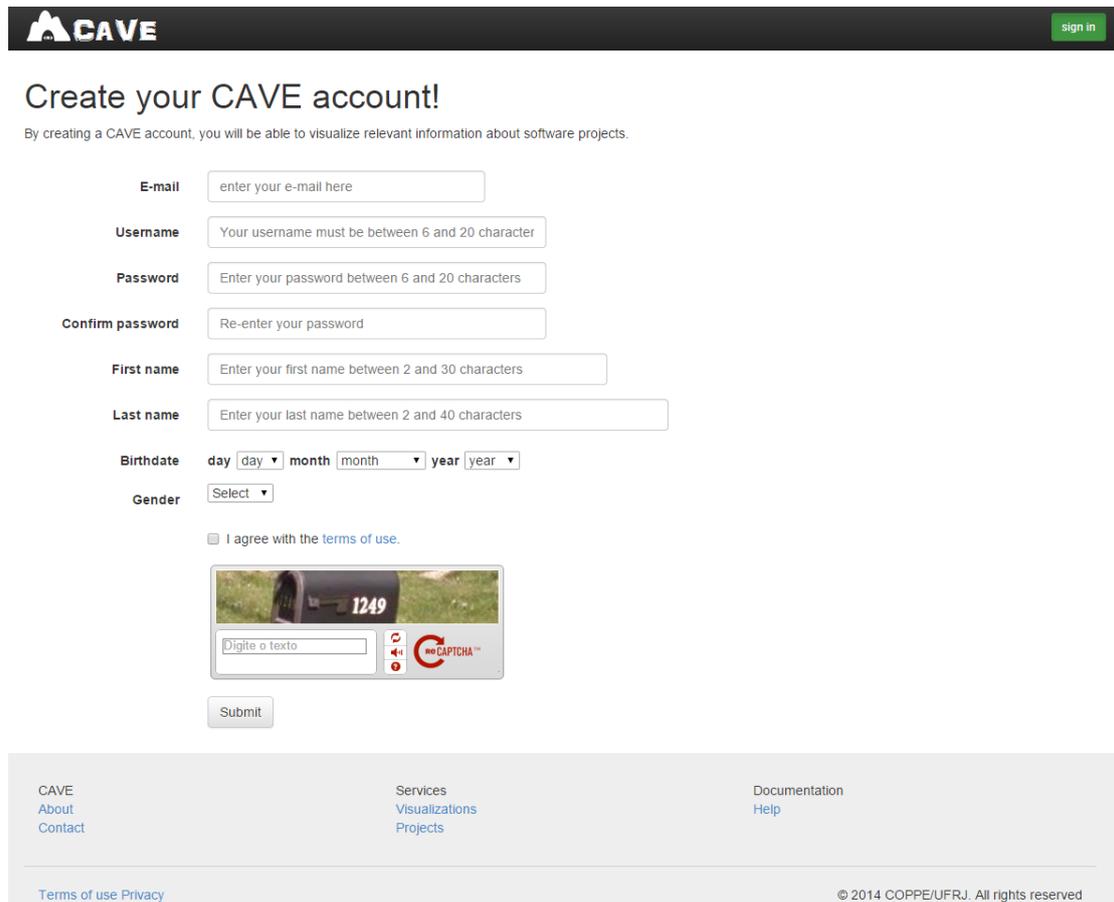
exibida na Figura 5.15, enquanto que a Figura 5.16 apresenta a página para cadastro de um novo usuário.



**Figura 5.14** - Barra superior para autenticação ou cadastro



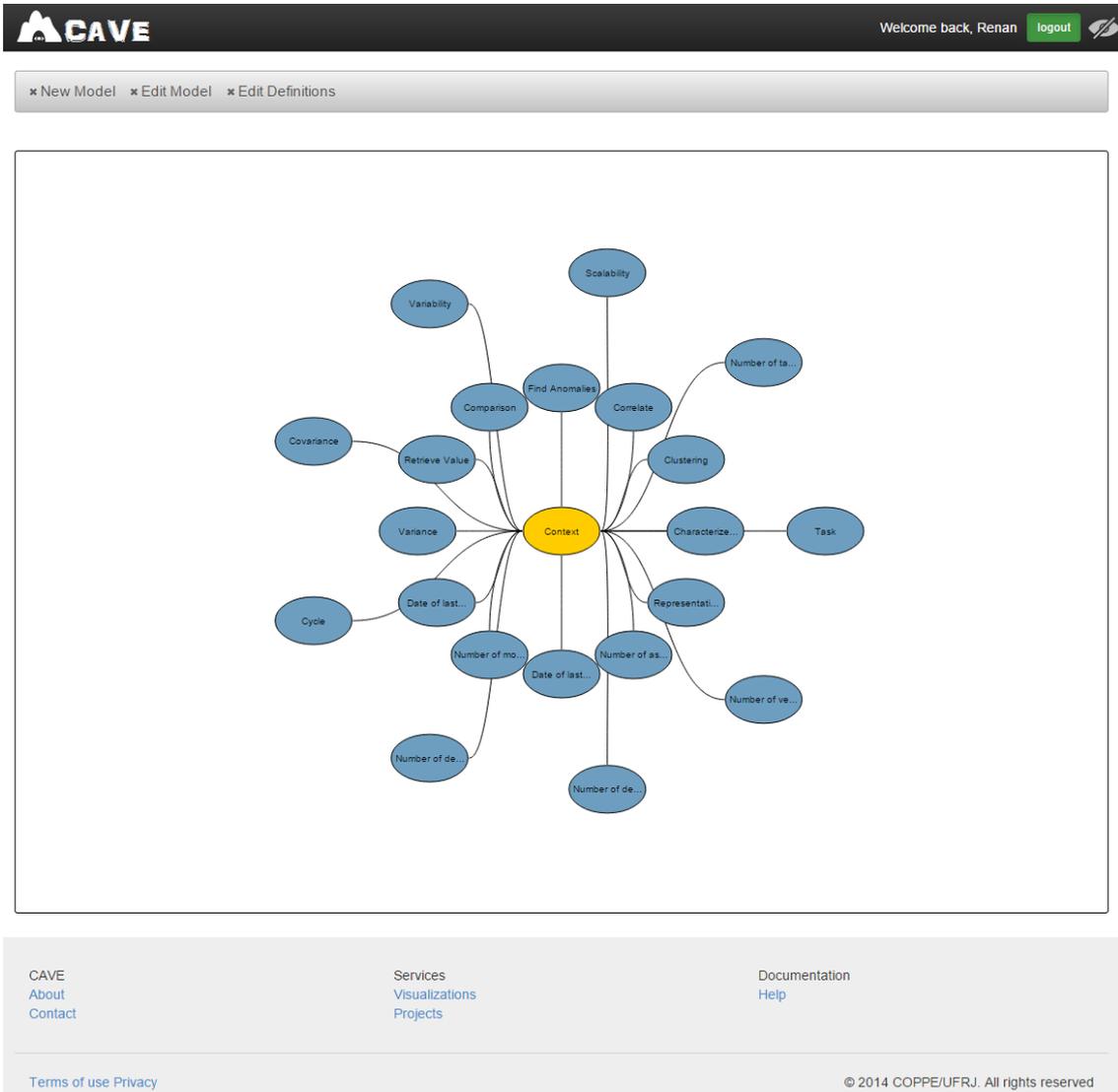
**Figura 5.15** - Página para autenticação do usuário



**Figura 5.16** - Página para cadastro de novo usuário

## 5.6.1 Administrador

Para um usuário com privilégios de administrador, ao se realizar a autenticação, este será encaminhado para a página principal de gerenciamento da plataforma. Tal página exibe um menu com opções específicas de administração e uma representação do modelo de contexto, se algum modelo já tiver sido previamente importado. A Figura 5.17 apresenta esta página principal para gerenciamento da plataforma CAVE.



The screenshot displays the CAVE administrator interface. At the top, there is a header with the CAVE logo on the left, a user greeting "Welcome back, Renan" and a "logout" button on the right. Below the header is a navigation menu with three items: "x New Model", "x Edit Model", and "x Edit Definitions". The main content area features a central yellow node labeled "Context" connected to numerous other nodes in a radial pattern. These nodes include: Scalability, Find Anomalies, Complete, Clustering, Characterize..., Task, Representati..., Number of ve..., Number of de..., Number of as..., Date of last..., Number of mo..., Cycle, Date of last..., Variance, Retrieve Value, Covariance, Comparison, and Variability. The bottom of the page contains a footer with links for "CAVE About Contact", "Services Visualizations Projects", and "Documentation Help", along with "Terms of use Privacy" and a copyright notice "© 2014 COPPE/UFRJ. All rights reserved".

**Figura 5.17** - Página de gerenciamento da plataforma

Caso o administrador deseje importar um novo modelo de contexto na plataforma, este deve selecionar a opção “New Model” no menu de gerenciamento. Na página a ser carregada, o administrador deve selecionar três arquivos XMI exportados pelo ambiente Odyssey: um com as entidades e informações de contexto, outro com as definições e um terceiro arquivo com as regras de contexto. Tais arquivos selecionados deverão ser

enviados ao servidor da plataforma por meio das opções de *upload* na página. A Figura 5.18 exibe esta página para inserir o novo modelo de contexto.

CAVE

Welcome back, Renan [logout](#)

✕ New Model ✕ Edit Model ✕ Edit Definitions

Insert XMI files containing context model elements created using the UbiFEX notation with the Odyssey environment.

Context Information XMI File:

+ Choose Upload Cancel

Context Definitions XMI File:

+ Choose Upload Cancel

Context Rules XMI File:

+ Choose Upload Cancel

Next

CAVE  
About  
Contact

Services  
Visualizations  
Projects

Documentation  
Help

Terms of use Privacy

© 2014 COPPE/UFRJ. All rights reserved

**Figura 5.18** - Página para inserir um novo modelo de contexto na plataforma

Após submeter um novo modelo de contexto, o administrador pode selecionar a opção “Next” para ir para a próxima etapa, mapeando as informações de contexto presentes no modelo com as informações suportadas pela plataforma. Se um modelo já estiver previamente registrado, o administrador também pode editar esse mapeamento através da opção “Edit Model” no menu de gerenciamento. A Figura 5.19 apresenta a página para realizar o mapeamento entre informações na plataforma.

Com as informações mapeadas, o administrador precisa relacionar definições de contexto presentes no modelo de contexto e aquelas suportadas pela plataforma CAVE. Para efetuar esta operação, o administrador pode selecionar a opção “Next” na página de mapeamento de informações, ou mesmo escolher a opção “Edit Definitions” no menu de gerenciamento. A Figura 5.20 exibe essa página para mapear as definições de contexto na plataforma.

[New Model](#)
[Edit Model](#)
[Edit Definitions](#)

| Map modeled information with supported information: |  |  |
|---|--|--|
| Modeled Information                                 | Supported Information                              |  |
| Representation Focus                                |  |  |
| Number of assets with one developer                 | Number of Assets with One Developer Within a Month |  |
| Date of last commit                                 | Date of an User's Last Commit                      |  |
| Number of modifications in last commit of a class   | Number of Issues with Open Status                  |  |
| Date of last version of a class                     | Number of Users Assigned to Low Priority Issues    |  |
| Variance  | Date of an User's Last Commit                      |  |
| Retrieve Value                                      | Number of Assets with One Developer Within a Month |  |
| Comparison  | Number of Modifications in Last Commit             |  |
| Find Anomalies                                      | Number of Tasks With Opened Status                 |  |
| Correlate   | Number of Developers of a Component                |  |
| Clustering  | Number of Versions of a Component                  |  |
| Characterize Distribution                           | Number of Dependencies of a Component              |  |
| Cycle   |  |  |
| Covariance  |  |  |
| Variability   |  |  |
| Scalability   |  |  |
| Number of tasks with open status                    | Number of Tasks With Opened Status                 |  |
| Task  |  |  |
| Number of versions of a component                   | Number of Versions of a Component                  |  |
| Number of developers of a component                 | Number of Developers of a Component                |  |
| Number of dependencies of a class                   | Number of Dependencies of a Component              |  |

[Next](#)

[CAVE](#)  
[About](#)  
[Contact](#)

[Services](#)  
[Visualizations](#)  
[Projects](#)

[Documentation](#)  
[Help](#)

[Terms of use](#) [Privacy](#)

© 2014 COPPE/UFRJ. All rights reserved

**Figura 5.19** - Página para mapear informações modeladas e informações suportadas

[New Model](#)
[Edit Model](#)
[Edit Definitions](#)

| Map modeled definitions with supported definitions: |                                    |  |
|---|------------------------------------|--|
| Modeled Definition                                  | Supported Definition               |  |
| High Dependency Between Classes                     | High Dependency Between Classes    |  |
| Concurrent Development                              | Concurrent Development             |  |
| Modified Project                                    | Modified Project                   |  |
| Modified Project With Details                       | Modified Project With Details      |  |
| Excess of Open Tasks                                | Excess of Open Tasks               |  |
| Excess of Open Tasks per Developer                  | Excess of Open Tasks per Developer |  |
| Low Collaboration                                   | Low Collaboration                  |  |

[Next](#)

[CAVE](#)  
[About](#)  
[Contact](#)

[Services](#)  
[Visualizations](#)  
[Projects](#)

[Documentation](#)  
[Help](#)

[Terms of use](#) [Privacy](#)

© 2014 COPPE/UFRJ. All rights reserved

**Figura 5.20** - Página para mapear definições de contexto modeladas e suportadas

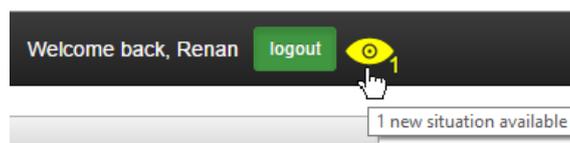
## 5.6.2 Usuário

Um usuário comum, após realizado seu processo de autenticação, pode utilizar a plataforma CAVE através de suas funcionalidades disponíveis. Um menu de usuário é apresentado, destacando, além das opções de carregar a página inicial da plataforma (“Home”) e voltar para a página anterior (“Go Back”), dois modos de utilização: sensível ao contexto (“Context-Aware Mode”) ou manual (“Select Situation”). A Figura 5.21 apresenta o menu do usuário com suas diferentes opções.



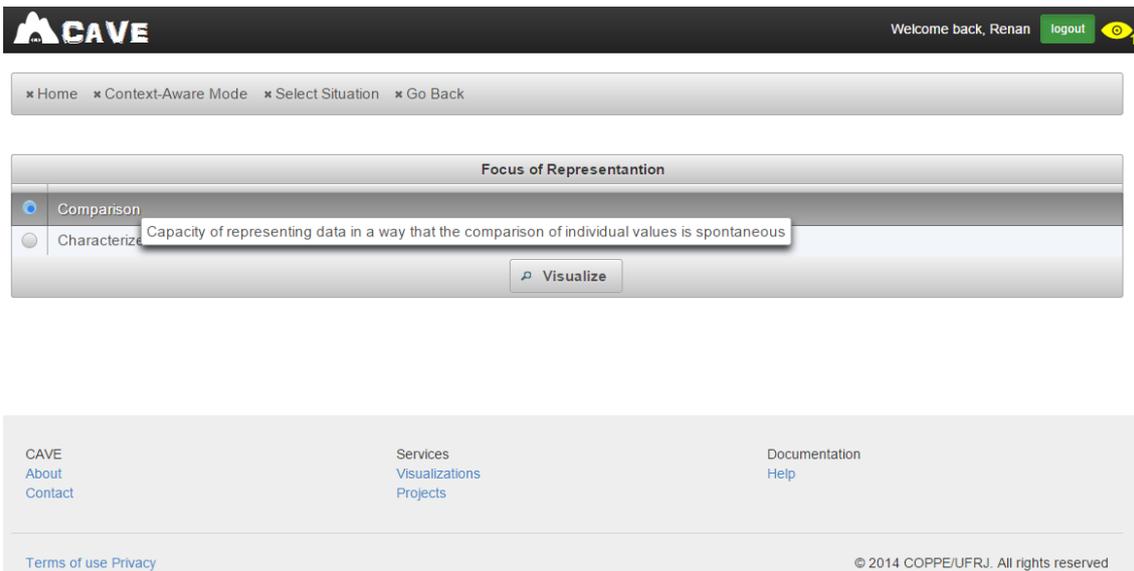
**Figura 5.21** - Menu de usuário

Para o modo sensível ao contexto, o usuário receberá notificações de definições de contexto ativas no momento. Através de um ícone na barra superior da plataforma, é indicado ao usuário o número de definições de contexto ativas, ou situações em ocorrência. A Figura 5.22 exibe a forma de notificação na plataforma CAVE. A fim de seguir uma notificação, o usuário pode proceder com um clique sobre o ícone de notificação, ou selecionar a opção “Context-Aware Mode” no menu de usuário.



**Figura 5.22** - Notificação de definição de contexto ativa

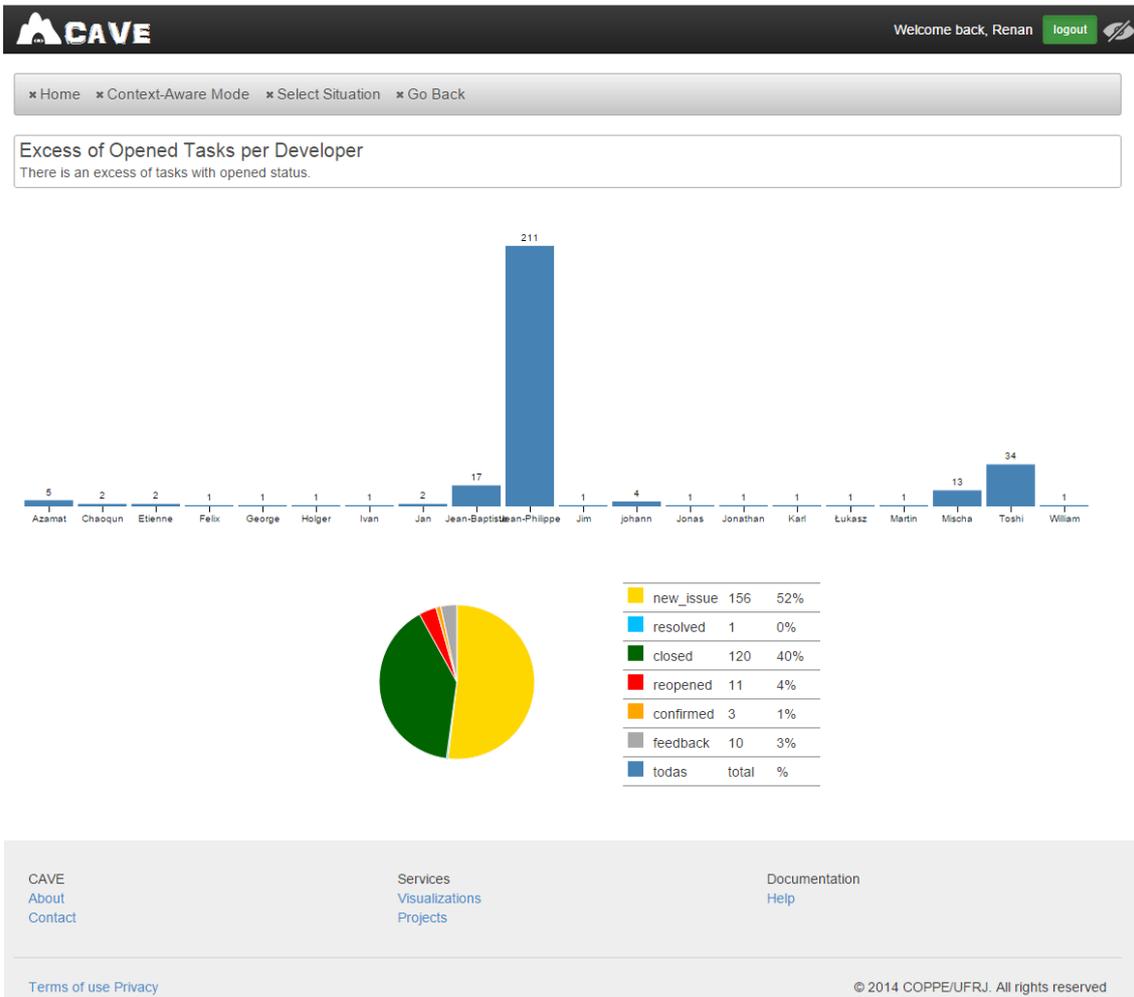
Se mais de uma definição de contexto estiver ativa no momento, ao seguir a notificação, o usuário irá para uma página em que poderá escolher para qual definição deseja obter uma visualização de apoio no momento. Caso apenas uma definição esteja ativa, a página a ser carregada será a de seleção do foco de representação, permitindo ao usuário visualizar dados sob diferentes perspectivas para tratar uma mesma situação. A Figura 5.23 apresenta esta página para seleção do foco de representação.



**Figura 5.23** - Página para escolher o foco de representação desejado

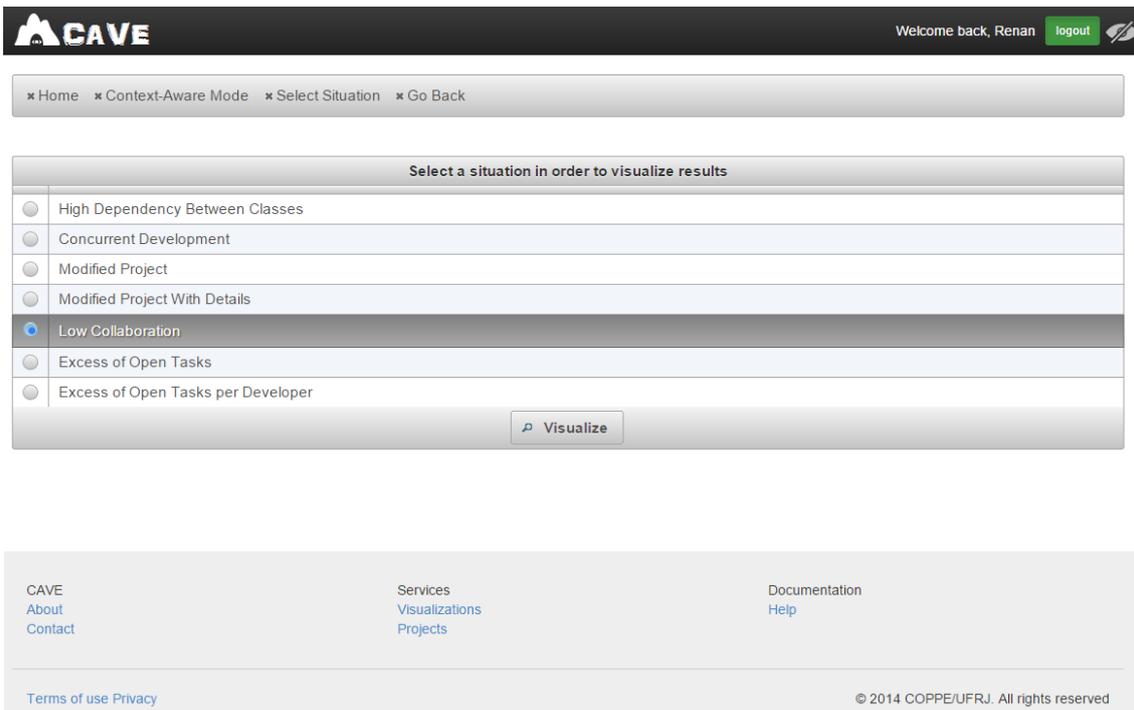
Com o foco selecionado, o usuário pode selecionar a opção “Visualize” para então ser direcionado para uma visualização. Na página seguinte, uma visualização é exibida, permitindo ao usuário interagir e analisar todos os dados representados. A Figura 5.24 apresenta esta página com a visualização. Cabe ressaltar que o usuário pode voltar à página anterior através da opção “Go Back” no menu para selecionar outro foco de representação e analisar outra visualização para a mesma definição de contexto ativa.

Após o direcionamento para uma visualização, a plataforma continua verificando valores para o contexto mapeado. Na ocorrência de outras definições de contexto, novas notificações serão destacadas para o usuário. Como os valores que caracterizam uma definição de contexto como ativa permanecem durante a análise da respectiva visualização ou mesmo após, caso modificações não sejam efetuadas rapidamente nas atividades que resultam em tais valores, para uma mesma sessão de um usuário na plataforma, notificações referentes a uma definição de contexto previamente visualizada são desconsideradas. Somente em uma sessão futura tal definição passaria a ser novamente considerada para ser notificada em caso de sua ocorrência.



**Figura 5.24** - Página com a visualização

Para o modo manual de utilização da plataforma CAVE, o usuário pode selecionar a opção “Select Situation”. Dessa forma, este pode visualizar dados referentes a qualquer definição de contexto a qualquer momento, independente se tal definição encontra-se ativa ou não. A Figura 5.25 apresenta a página para escolher uma definição de contexto manualmente para visualização. Como esta listagem de definições já reúne todas as definições modeladas, onde cada definição está relacionada a um foco de representação, com a definição selecionada na página, basta ao usuário proceder com a opção “Visualize” para ser redirecionado diretamente para a página com uma visualização, sem ter que selecionar um foco de representação.



**Figura 5.25** - Página para seleção de definição de contexto

## 5.7 Considerações Finais

Com a plataforma CAVE implementada, pode-se constatar a forma como as funcionalidades disponibilizadas atendem aos conceitos expostos na abordagem. Com o protótipo desenvolvido como uma aplicação *web*, busca-se uma facilidade de acesso em diferentes ambientes e dispositivos, a fim de garantir compatibilidade.

A implementação da plataforma, seguindo a infraestrutura proposta na abordagem, possui algumas limitações. Com esse reconhecimento, em versões futuras do protótipo, seria possível planejar alternativas para superar tais limitações. Como restrições reconhecidas, podem ser destacadas as seguintes limitações:

- A modelagem de contexto é feita externamente à plataforma, dependendo de sua construção por meio do ambiente Odyssey, com posterior exportação em formato XMI;
- A seleção de um elemento de visualização em uma regra de contexto depende que tal elemento esteja disponível na plataforma, que sua implementação tenha sido realizada.

A partir da plataforma, pode-se analisar de forma mais aprofundada como ocorreria a adoção da abordagem. Nesse sentido, uma avaliação conduzida por meio da plataforma CAVE poderia reunir indicadores destacando benefícios e dificuldades de sua utilização.

# CAPÍTULO 6 - ESTUDO DE OBSERVAÇÃO

## 6.1 Introdução

Com a abordagem CAVE propondo uma discussão com relação à adoção de conceitos de sensibilidade ao contexto para visualizações de software, é importante verificar como essa adoção se comporta quando envolvida com cenários detalhados e participantes reais, de diferentes perfis. Uma avaliação é uma forma importante de ponderar os resultados que a abordagem pode prover quando colocada em prática.

Um estudo de observação, avaliando a abordagem por meio da utilização da plataforma implementada, é desenvolvido neste capítulo. Visando estruturar a descrição do estudo, a Seção 6.2 apresenta o objetivo, enquanto a Seção 6.3 destaca a hipótese considerada. Uma definição do estudo de observação é descrita na Seção 6.4. Uma descrição geral da apresentação dos cenários é exposta na Seção 6.5. Os procedimentos seguidos durante a execução do estudo são apresentados na Seção 6.6. Já na Seção 6.7, os resultados coletados são apresentados. Observações sobre a validade do estudo são feitas na Seção 6.8, enquanto as considerações finais são expostas na Seção 6.9.

## 6.2 Objetivo

O objetivo deste estudo de observação, segundo a estrutura do GQM (BASILI *et al.*, 1994), é **analisar** o uso da visualização sensível ao contexto **com o propósito de** caracterizar, **com respeito à** adequação das visualizações **do ponto de vista** de gerentes de projeto **no contexto de** compreensão de atividades relacionadas ao desenvolvimento de software.

## 6.3 Hipótese

A hipótese deste estudo é que visualizações orientadas ao contexto de uma organização podem auxiliar na percepção de questões importantes, como alocação de tarefas e colaboração entre desenvolvedores, bem como na ocorrência de problemas, no que diz respeito a atividades comuns à gerência de projetos de desenvolvimento de software.

## 6.4 Definição

Espera-se observar como gerentes de projeto interagem com a plataforma CAVE, de forma a utilizar as visualizações como mecanismo de suporte à realização de atividades de desenvolvimento de software. Mais especificamente, buscou-se verificar,

qualitativamente, se com a utilização da plataforma CAVE, os participantes conseguiam identificar os problemas em destaque no processo de desenvolvimento de software através de visualizações indicadas para os mesmos ou selecionadas manualmente.

A aplicação deste estudo de viabilidade da abordagem envolve a observação da utilização da plataforma CAVE por participantes diante da análise de visualizações de software e mudanças de contexto, o que promove a exibição de notificações sobre novas visualizações disponíveis para consulta. Nesse sentido, visando à realização da avaliação, os seguintes pontos foram observados:

- Os participantes conseguiram identificar questões/problemas apontados em cada visualização?
- Os participantes conseguiram identificar questões/problemas após a mudança de contexto?
- Quanto tempo levou para que os participantes conseguissem interpretar as visualizações antes da mudança de contexto?
- Quanto tempo levou para que os participantes conseguissem interpretar as visualizações após a mudança de contexto?
- Quais dificuldades os participantes tiveram e/ou observaram no uso da plataforma durante as tarefas realizadas?
- Mudanças de visualizações ajudaram o usuário a entender melhor o problema atual?

Para ser possível realizar tais observações, o participante deveria se colocar como membro de uma organização que realiza desenvolvimento de software. Para que o participante tivesse consciência dos possíveis problemas com que tal organização pudesse se deparar, antes do estudo de observação ser iniciado, foi apresentado ao participante, através de definições de contexto, o conjunto de problemas possíveis que ele, como gerente de projeto, teria a responsabilidade de tratar na organização. Alguns desses problemas foram repassados ao participante por meio de mudanças propositas nos valores de contexto.

## **6.5 Cenários**

Como foi analisada a execução de cada participante de forma individual, estes foram separados em sessões diferentes. Em cada sessão, foi apresentado o conjunto de questões pelas quais o gerente de projeto ficaria responsável por analisar.

Como a instância atual da plataforma CAVE conta com informações provenientes de projetos de software oriundos de sistemas de gerenciamento de tarefas e controle de versão, quatro cenários foram apresentados para cada participante, em conjunto com uma descrição geral das tarefas a serem realizadas. Cada cenário pode ser apresentado como um ambiente em que o participante ocupa um papel de gerente de projeto e é responsável por gerenciar a execução das tarefas a serem desempenhadas durante o desenvolvimento.

Para cada cenário, informações referentes a um projeto de exemplo provenientes do sistema de gerenciamento de tarefas Redmine foram utilizadas para compor as definições de contexto em conjunto com informações referentes a controle de versão, provenientes do sistema SVN, desde que servissem para compor indicadores de problemas de interesse de um gerente de projeto.

## **6.6 Procedimentos**

Para cada sessão, o participante deveria ler e preencher um Formulário de Consentimento (Apêndice A). Concordando com o conteúdo, deveria ser preenchido um Questionário de Caracterização (Apêndice B), de forma a avaliar o nível de conhecimento e experiência do participante. Com isso, poderia ser feita uma análise do perfil de cada participante e possíveis impactos para os resultados obtidos de sua participação ao final. Como possibilidades de impacto a partir do perfil do participante, espera-se que uma maior experiência na área de visualização de informação possa promover uma facilidade maior para utilizar as representações gráficas disponíveis na plataforma CAVE, bem como uma maior experiência em gerência de projetos possa contribuir para interpretações dos dados nas visualizações.

Um breve treinamento foi realizado com cada participante, demonstrando as opções disponíveis para uso na plataforma CAVE. De forma geral, foram indicados: (i) o acesso à opção de seleção manual de uma definição de contexto para sua respectiva visualização; e (ii) a seleção de focos de representação, quando disponíveis, para uma dada definição de contexto.

Após os formulários, o participante tomou conhecimento de cada um dos cenários (Apêndice C) e questões sob rastreamento pela plataforma CAVE. Com isso, poderia efetivamente utilizar a plataforma, verificando as visualizações apresentadas para diferentes definições de contexto, propositalmente destacadas por mudanças diretas no contexto da organização hipotética. Durante a execução desta parte, o participante deveria relatar oralmente os pontos de destaque apresentados pelas visualizações por ele

percebidos, interpretando a representação gráfica e mencionando como os dados ressaltam um problema para o processo de desenvolvimento de software da organização.

Foi encorajado ao participante falar durante todo o processo, de forma a descrever todo o raciocínio, seguindo o protocolo *think aloud* (LEWIS, 1982). A fim de facilitar a análise dos resultados posteriormente, esta etapa do processo de avaliação foi acompanhada por gravação em áudio, capturando todas as reações expressas oralmente por cada participante.

Ao final de cada cenário, o participante responderia questões presentes em um Questionário de Observação (Apêndice D). Tal documento serviu para identificar os problemas associados às tarefas de desenvolvimento, por meio das visualizações apresentadas. Através das respostas e da fala dos participantes durante a execução de cada cenário, foi possível verificar se foram alcançados alguns dos resultados esperados nos gabaritos ou mesmo outras interpretações adequadas não previstas. É importante destacar que a solução dada pelo participante não é o mais importante. O principal da análise dos cenários diz respeito à identificação dos problemas. A solução sugerida pelo participante ajuda a entender se este compreendeu o problema atual na organização.

Após a conclusão de todos os cenários, o participante iria preencher o Questionário de Observação Geral (Apêndice E), que serviu para avaliar dificuldades em utilizar a plataforma CAVE. Sugestões e críticas também poderiam ser anotadas para aperfeiçoar a plataforma e a abordagem.

Cada cenário tem seu roteiro detalhado no Apêndice F, servindo para a indicação dos passos apresentados e seguidos pelo participante durante seu uso da plataforma CAVE. No Apêndice G, o gabarito apresenta alguns dos possíveis resultados esperados ao final de cada tarefa, sendo importante destacar que diferentes respostas corretas poderiam ser dadas pelos participantes. Pelos roteiros e gabaritos listados, percebe-se que existe uma ordem planejada para execução dos cenários. Inicialmente, o usuário iria apenas analisar uma visualização e procurar utilizá-la para apoiar na resolução de problemas do Cenário A. No Cenário B, a visualização apresentada para o participante não seria a mais adequada para apoiar o gerente no cenário atual. No Cenário C, haveria uma notificação de contexto ativo, indicando uma visualização mais apropriada para apoiar o problema do cenário que é o mesmo do Cenário B, destacando a importância de uma visualização mais adequada para cada contexto. Já no Cenário D, ocorreria uma notificação de contexto ativo com seleção de foco de representação, em que cada

visualização consequente de um foco apresenta uma perspectiva diferente, indicando aspectos diferentes do cenário.

Antes da realização dos estudos com os participantes selecionados, foi efetuado um piloto do estudo de observação em um dia anterior. Um participante convidado realizou todos os procedimentos mencionados, a fim de que fosse possível verificar a adequação do estudo para efetivação com os demais participantes. Não houve nenhuma modificação dos procedimentos a partir do piloto, sendo útil, no entanto, para estimar o tempo necessário em cada cenário, bem como deixar o ambiente configurado.

## **6.7 Resultados**

Os resultados foram analisados levando em consideração tanto o perfil dos participantes como as respostas dadas por eles durante a execução do estudo. Os resultados obtidos por cada participante, com relação à identificação dos problemas a partir dos dados presentes nas visualizações, servem para avaliar a disposição dos elementos de visualização selecionados para a visualização por meio da regra de contexto previamente estabelecida. Já com relação à troca de contexto e sua percepção por parte do participante, os resultados foram extraídos de forma qualitativa através das respostas e críticas que este forneceu, além de observações referentes ao raciocínio, coletadas com o protocolo *think aloud*.

### **6.7.1 Caracterização dos Participantes**

O estudo foi realizado com 6 participantes, selecionados por conveniência. Foram procurados participantes que tivessem experiência com visualização de informação e/ou análise de *dashboards*. A aplicação de tais conhecimentos na indústria foi considerada como um diferencial importante a fim de avaliar a experiência de cada participante. De todos os participantes, 2 haviam concluído a graduação e estavam trabalhando com desenvolvimento na indústria e 5 são alunos de mestrado em Engenharia de Sistemas na COPPE/UFRJ, com linhas de pesquisa variando entre Banco de Dados, Engenharia de Software e Redes de Computadores. A Tabela 6.1 reúne os dados referentes à caracterização dos participantes.

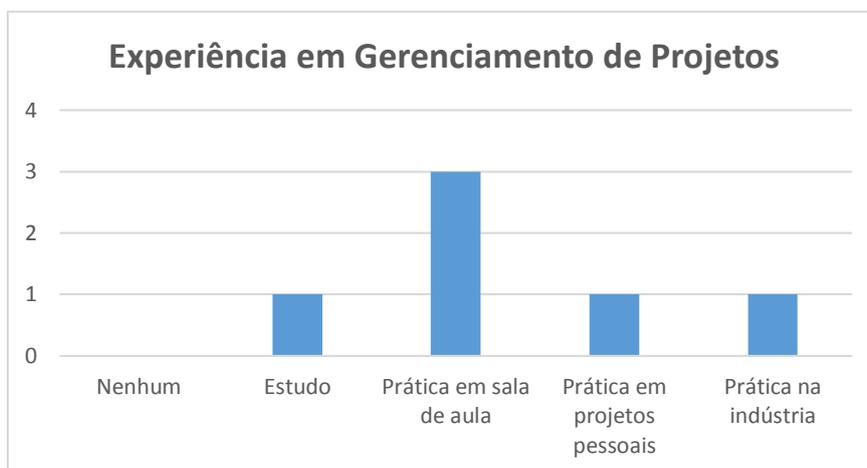
A caracterização quanto ao conhecimento dos participantes com relação a gerenciamento de projetos e visualização de informação ou análise de *dashboards* utilizava uma escala de valores variando dos níveis 0 até 4, com as seguintes descrições: 0 significa nenhum conhecimento adquirido; 1 representa ter estudado em aula ou livro;

2 significa ter praticado em projetos de sala de aula ou curso; 3 representa o uso em projetos pessoais; e 4 significa a prática em tais áreas na indústria.

**Tabela 6.1** - Caracterização dos participantes

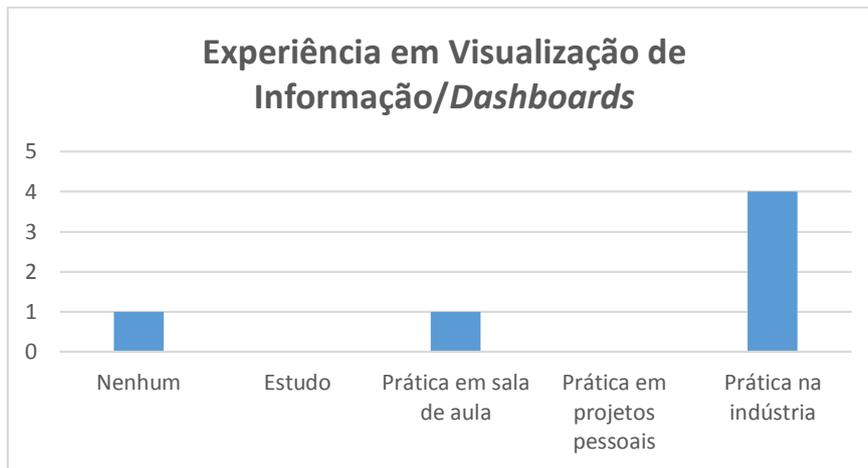
|                                    | Participantes  |                        |           |                       |           |                        |
|------------------------------------|----------------|------------------------|-----------|-----------------------|-----------|------------------------|
|                                    | P1             | P2                     | P3        | P4                    | P5        | P6                     |
| <b>Formação</b>                    | Mestrando      | Mestrando              | Graduado  | Mestrando             | Graduado  | Mestrando              |
| <b>Linha de Pesquisa</b>           | Banco de Dados | Engenharia de Software | -         | Redes de Computadores | -         | Engenharia de Software |
| <b>Gerenciamento</b>               | 2              | 3                      | 1         | 2                     | 2         | 4                      |
| <b>Visualização</b>                | 4              | 4                      | 4         | 0                     | 2         | 4                      |
| <b>Experiência – Tempo (anos)</b>  | 10             | 4                      | 1         | 3                     | 5         | 3                      |
| <b>Experiência – Ambiente</b>      | Indústria      | Indústria              | Indústria | Indústria             | Indústria | Indústria              |
| <b>Experiência - Classificação</b> | Sênior         | Pleno                  | Júnior    | Pleno                 | Sênior    | Pleno                  |

A Figura 6.1 apresenta um gráfico com o número de participantes em cada nível de experiência em gerenciamento de projetos de software. Pelos resultados, observa-se que a maioria dos participantes praticou de alguma forma o gerenciamento de projetos, apesar de que apenas um (P6) utilizou tal conhecimento na indústria.



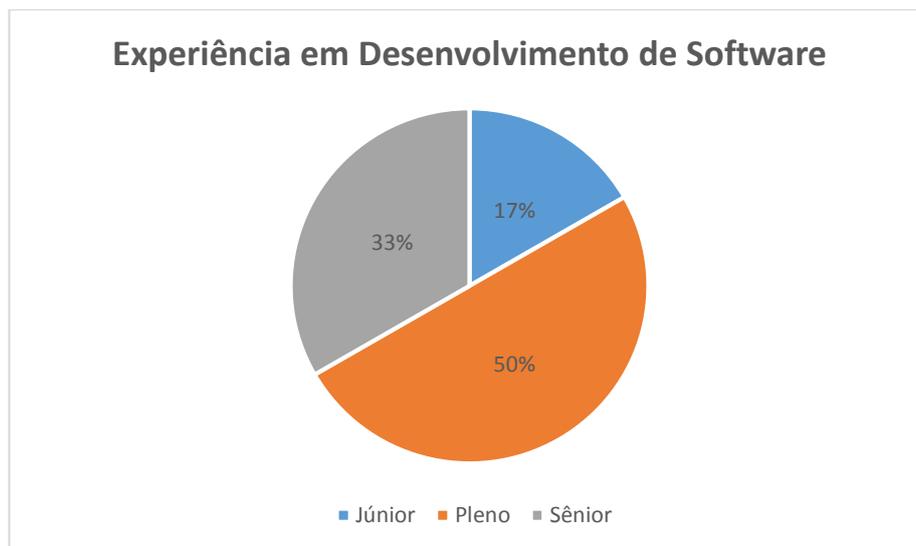
**Figura 6.1** - Número de participantes para cada nível de experiência em gerenciamento de projetos de software

A Figura 6.2 apresenta um gráfico com o número de participantes em cada nível de experiência em visualização de informação ou análise de *dashboards*. Pelos resultados, observa-se que a maioria dos participantes teve contato com essa área na indústria, sendo que apenas um participante (P4) declarou não ter nenhum conhecimento na área. Apesar deste fato, tal participante não foi excluído do estudo pois este possui conhecimento relevante na área de redes, estando acostumado a lidar com modelos de grafos, cuja metáfora de rede possui aplicação em um dos cenários, além das observações pertinentes fornecidas de forma geral.



**Figura 6.2** - Número de participantes para cada nível de experiência em visualização de informação/*dashboards*

No que diz respeito à experiência em desenvolvimento de software, três categorias podem ser identificadas: (i) júnior, em que o participante tem menos de 2 anos de experiência na indústria; (ii) pleno, em que o participante tem entre 2 e 4 anos de experiência; e (iii) sênior, em que o participante possui mais do que 4 anos de experiência na indústria. A Figura 6.3 apresenta a porcentagem de perfis identificados durante o estudo. Percebe-se que a grande maioria possui mais de 2 anos de experiência, em que os perfis estariam associados às categorias pleno e sênior. Apenas o participante P3 possui 1 ano de experiência na indústria.



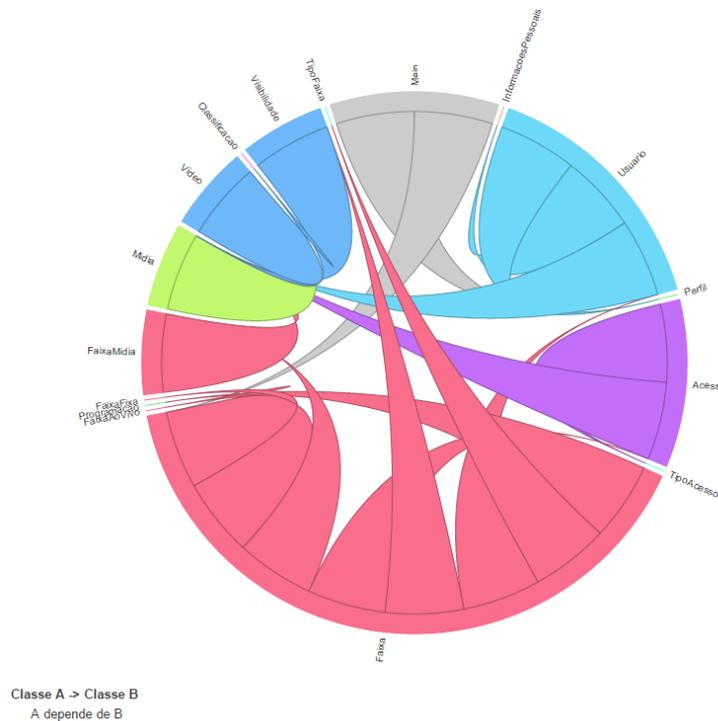
**Figura 6.3** - Porcentagem de participantes em cada categoria de experiência em desenvolvimento de software

Todos os participantes do estudo de observação têm alguma participação na indústria. Tal caracterização é importante por verificar que os participantes têm por hábito lidar com problemas reais, em cenários multivariados e dinâmicos.

## 6.7.2 Cenário A

Conforme apresentado no Apêndice C, o Cenário A descreve um problema de identificação de dependências entre classes. No entanto, apesar do destaque para a importância de alocar desenvolvedores para a manutenção de classes com um número grande de dependências, a questão colocada ao final da descrição do cenário atenta para a necessidade de se identificar a classe que era alvo de dependências do maior número de outras classes.

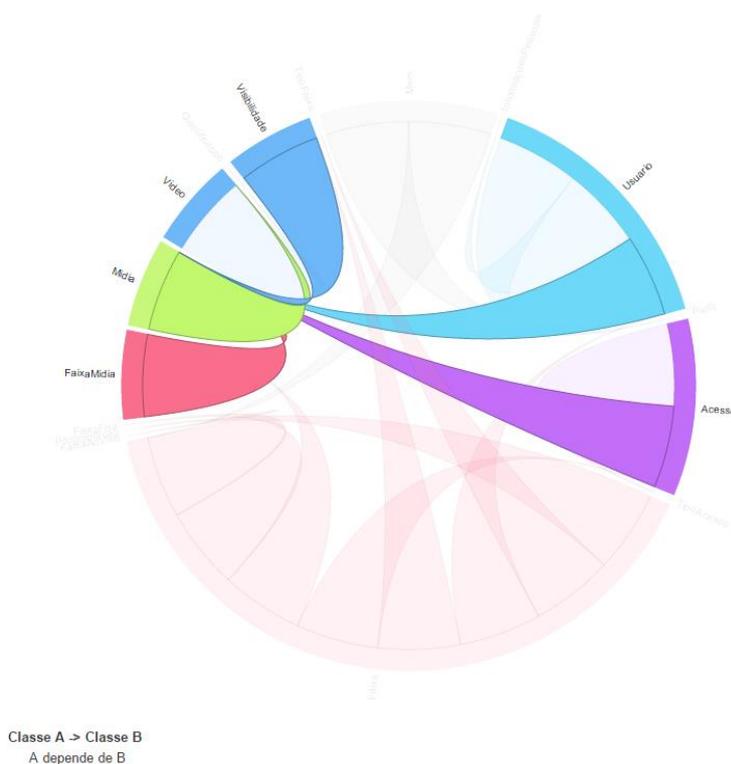
A visualização diretamente apresentada para os participantes analisarem é a indicada na Figura 6.4, também conhecida como *Dependency Wheel*. Tal visualização destaca os relacionamentos que determinada classe possui ao se mover o cursor do *mouse* sobre tal classe. Os dados visualizados são baseados em um projeto de Web TV desenvolvido no Grupo de Reutilização de Software da COPPE/UFRJ, no entanto, nenhum conhecimento sobre tal projeto seria necessário para realizar a tarefa indicada.



**Figura 6.4** - Visualização para analisar dependências entre classes

Houve uma dificuldade por parte dos participantes em entender rapidamente o problema, talvez por esta representação em particular não ser muito adotada para diferentes propósitos, ficando mais restrita a problemas de dependências entre elementos. Para os dados utilizados na representação gráfica, era possível perceber rapidamente qual classe possuía o maior número de dependências, pela extensão do arco relativo a cada classe, mais especificamente à classe Faixa, uma vez que quanto maior o número de

dependências de uma classe maior o seu arco na circunferência da visualização. Porém, a fim de identificar a classe que era alvo do maior número de dependências, seria necessário interagir destacando cada classe e verificar o número de classes que dependiam da classe sob foco. A Figura 6.5 apresenta a visualização modificada pela interação sobre a classe Midia. Neste caso, é possível notar que Midia é alvo de dependências de outras 4 classes. A Tabela 6.2 lista os resultados obtidos para todos os participantes ao longo deste cenário.



**Figura 6.5** - Classe Midia destacada e as dependências de outras classes

**Tabela 6.2** - Resultado da execução da tarefa do Cenário A

|                            | Participantes |       |       |       |       |       |
|----------------------------|---------------|-------|-------|-------|-------|-------|
|                            | P1            | P2    | P3    | P4    | P5    | P6    |
| <b>Tempo (minutos)</b>     | 10:50         | 12:56 | 16:56 | 17:40 | 18:45 | 17:44 |
| <b>Houve dificuldades?</b> | Sim           | Sim   | Sim   | Não   | Sim   | Sim   |
| <b>Resposta Inicial</b>    | Faixa         | Faixa | -     | Midia | Faixa | Faixa |
| <b>Resposta Final</b>      | Faixa         | Faixa | -     | Midia | Midia | Midia |

A média de tempo utilizado pelos participantes para a realização da tarefa do Cenário A foi de 15:48. Os três participantes que utilizaram mais tempo para análise da visualização e interpretação dos resultados foram aqueles que conseguiram a resposta mais adequada e requisitada na descrição do cenário.

Através das respostas dadas pelos participantes no questionário de observação em conjunto com o que foi relatado oralmente durante a análise de visualizações, algumas observações interessantes de cada participante foram relatadas.

O participante P3 ressaltou que conseguia facilmente identificar qual classe tem mais dependências (classe Faixa), mas qual classe é alvo de mais dependências ficou confuso através da visualização. Por isso, não foi alcançado o resultado requisitado para o cenário, apesar de entender o que foi solicitado. Neste caso específico, um tempo maior de utilização e interação na representação gráfica poderia ser benéfico para identificar a forma de consultar de quais classes uma determinada classe é alvo de dependência.

Os participantes P4 e P5 identificaram que a representação gráfica tratava-se de uma forma de grafo direcionado. O participante P4 destacou que na representação tradicional de grafo direcionado seria mais fácil identificar qual classe é mais alvo de dependências de outras classes, enquanto que na representação *Dependency Wheel* encontra-se quem tem mais dependências diretamente buscando a classe com maior arco no círculo. Neste ponto, é importante ressaltar que, pela formação do participante P4 na área de Redes de Computadores, sua experiência em lidar com modelos representados por grafos influenciaria a seu favor na análise de uma visualização neste formato.

O participante P5 não percebeu inicialmente a legenda abaixo da visualização, o que prejudicou inicialmente para identificar qual classe dependia de qual, apesar de reconhecer um relacionamento entre duas classes. Após a descoberta da legenda e o sentido das dependências, conseguiu iterar por todas as classes para identificar Mídia como a classe alvo do maior número de dependências. Este participante destacou a importância de haver uma largura mínima para as faixas que representavam dependências, a fim de possibilitar a utilização da interação com o cursor sobre a faixa para verificar o sentido da dependência. A fim de superar essa dificuldade, utilizou o *zoom* do navegador para conseguir realizar a interação para visualizar a dica (*tooltip*). Nesse sentido, uma alteração no estilo de exibição das faixas poderia solucionar o problema relatado, bem como a disponibilização de um mecanismo de *zoom* geométrico. Além disso, um destaque maior para a área de legenda poderia agilizar o seu reconhecimento por parte dos participantes.

O participante P6 teve dificuldades para identificar o sentido da dependência entre duas classes que estavam lado a lado. Já os participantes P2, P3 e P4 gostariam que houvesse uma forma de interação que permitisse inverter o sentido das dependências, a fim de verificar mais facilmente a classe que fosse alvo de dependências.

O participante P4 destacou que talvez para muitas classes seja complicado identificar qual classe possui mais dependências apenas comparando a extensão do arco referente a cada classe. A plataforma CAVE teria flexibilidade com relação a isso,

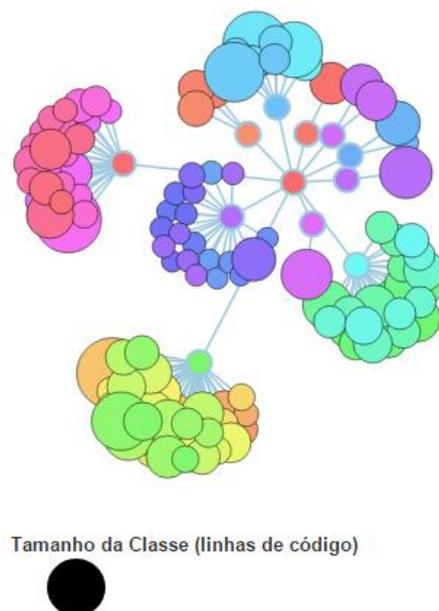
bastando uma regra de contexto para adotar um leiaute diferente para a visualização de acordo com o número de classes no conjunto de dados.

De forma geral, os participantes relataram que a visualização adotada era bem eficiente para identificar as classes que possuíam mais dependências, mas a fim de verificar qual classe era mais alvo de dependências de outras classes tornava-se mais complicado. Neste sentido, devido ao uso principal da visualização, alguns participantes não entenderam a solução solicitada pelo cenário, que demandava interações adicionais para ser obtida.

### 6.7.3 Cenário B

Também listado no Apêndice C, o Cenário B descreve um problema de baixa colaboração entre membros de uma equipe de desenvolvedores alocados para a manutenção de um projeto.

A visualização diretamente apresentada para os participantes analisarem é a indicada na Figura 6.6, uma árvore com a estrutura do projeto, em que as folhas representam arquivos e os nós intermediários são pacotes. Tal visualização é interessante para analisar a organização de um projeto, com a hierarquia entre pacotes, classes e arquivos. Os dados visualizados são baseados em um recorte do projeto de desenvolvimento do sistema de gerenciamento de tarefas Redmine.



**Figura 6.6** - Visualização com a estrutura de um projeto

Neste cenário, já era esperado que os participantes tivessem dificuldades para analisar o problema de acordo com a visualização apresentada, uma vez que esta não era adequada para o problema relatado na descrição de tal cenário. Verificou-se, no entanto,

se os participantes tiveram dificuldades em compreender o que a visualização representava.

As cores de cada arquivo na árvore não tinham um significado próprio, o que provocou confusão nos participantes, tentando fazer suposições sobre as cores e necessidades de informações que estes tinham para poder propor soluções para o cenário. Nesse sentido, de forma geral, os participantes indagavam sobre o significado das cores, uma vez que isto não constava na legenda, e sempre era informado que não havia nenhuma associação, sendo necessário utilizar para análise apenas o que estava disponível visualmente ou através de interações. A Tabela 6.3 lista os resultados obtidos para todos os participantes ao longo deste cenário.

**Tabela 6.3** - Resultado da execução da tarefa do Cenário B

|  | Participantes |       |       |       |       |       |
|--|---------------|-------|-------|-------|-------|-------|
|  | P1            | P2    | P3    | P4    | P5    | P6    |
| <b>Tempo (minutos)</b>                     | 05:02         | 13:03 | 13:50 | 18:50 | 31:16 | 10:51 |
| <b>Houve dificuldades na visualização?</b> | Não           | Não   | Sim   | Sim   | Sim   | Sim   |
| <b>Propôs solução?</b>                     | Não           | Sim   | Não   | Sim   | Sim   | Não   |

O tempo médio de execução da tarefa deste cenário foi de 15:29. Nesta questão, esperava-se que os resultados individuais fossem bem variados, uma vez que alguns participantes já não perceberiam nenhuma utilidade na visualização rapidamente, enquanto outros, imaginando que tal visualização fosse de fato útil, tentariam por um longo período desenvolver ideias que fossem aplicáveis ao cenário com as informações disponíveis na representação gráfica.

O participante P2 não destacou possibilidades de interpretação para as cores, uma vez que não constavam na legenda, mas propôs uma solução que consideraria apenas informações estruturais do projeto, alocando desenvolvedores a fim de evitar trabalhos concorrentes. Nesse caso, também há o destaque para a necessidade de se conhecer os relacionamentos entre os membros da equipe, visando alocações em módulos com maior dependência entre si.

Os participantes P1 e P3 indicaram a necessidade de informações adicionais, talvez incluindo na legenda, a fim de conseguir utilizar a visualização para apoiar o cenário. Com isso, não conseguiram pensar em uma solução para o problema relatado no cenário, considerando a visualização insuficiente.

O participante P4 conseguiu entender a estrutura existente em cada pacote por meio da interação de agregação possível ao se selecionar o nó referente a um pacote. Ele fez suposições com relação à cor dos nós tendo referência à complexidade associada à

manutenção de cada arquivo. Posteriormente, tal participante sugeriu a adoção de uma visualização com leiaute de grafo bipartido, formado por um conjunto de nós representando os componentes e outro conjunto com nós referentes a desenvolvedores, onde o relacionamento entre os nós dos dois conjuntos representaria que tal desenvolvedor trabalhou no componente em questão. Dessa forma, talvez fosse possível identificar quais desenvolvedores trabalharam em componentes em comum.

O participante P5 propôs uma solução assumindo que a cor de cada nó na árvore indicaria um desenvolvedor, baseando sua alocação de acordo com a cor no nó. Mas tal participante ressaltou que teve dificuldades para interpretar uma forma adequada de solução para o cenário a partir do que estava disponível na visualização.

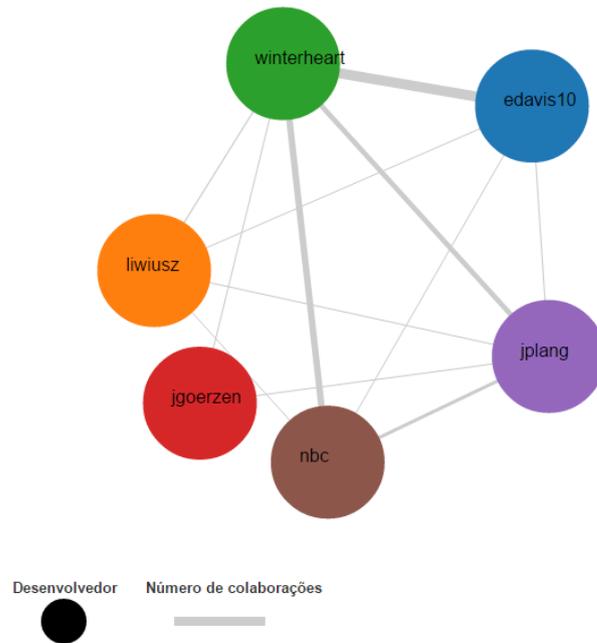
Com as cores dos nós funcionando como um fator de confusão, o que promoveu a busca por soluções através de suposições de seu significado, talvez com a adoção de uma única cor para todos os nós na árvore, os participantes pensariam em outras possibilidades. A forma de agrupamento na árvore poderia surgir como nova suposição, ou mesmo não haveria espaço para novas interpretações, tornando a resposta final para o cenário mais rápida, sem conseguir apoiar através da representação adotada. De qualquer forma, todos os participantes não consideraram a visualização apresentada como mecanismo de suporte eficiente para o cenário proposto, o que promoveu a sugestão de novas formas de representação gráfica que fossem realmente úteis.

#### **6.7.4 Cenário C**

Assim como consta no Apêndice C, o Cenário C descreve um problema de baixa colaboração, idêntico ao Cenário B. A execução da tarefa relativa a este cenário incluía continuar a análise dos dados por meio da visualização disponível na Figura 6.6 e, após breve período, de poucos segundos, uma notificação de nova definição de contexto ativa seria disponibilizada.

Com a definição de contexto ativa sendo notificada, o resultado seria uma representação gráfica que efetivamente exibisse informações sobre os membros da equipe do projeto em questão, de forma a apoiar decisões referentes à colaboração entre os desenvolvedores. A Figura 6.7 apresenta essa visualização mais adequada para o cenário analisado. Na representação, as colaborações são traduzidas por atividades realizadas em artefatos em comum, ou seja, dois desenvolvedores colaboraram entre si se trabalharam em um mesmo artefato, mesmo que em momentos diferentes na organização. Os dados

visualizados são baseados no projeto de desenvolvimento do sistema de gerenciamento de tarefas Redmine, exibindo o nome de usuário de desenvolvedores no sistema SVN.



**Figura 6.7** - Visualização com as colaborações entre os desenvolvedores

Na visualização apresentada após a notificação de definição de contexto ativa, os participantes dispunham de interações como *zoom* geométrico, deslocamento dos nós, e *tooltip* das arestas, com o número de colaborações em cada aresta. A Tabela 6.4 lista os resultados obtidos para todos os participantes ao longo deste cenário.

**Tabela 6.4** - Resultado da execução da tarefa do Cenário C

|  | Participantes |       |       |       |       |       |
|--|---------------|-------|-------|-------|-------|-------|
|  | P1            | P2    | P3    | P4    | P5    | P6    |
| <b>Tempo (minutos)</b>                     | 11:00         | 9:20  | 16:20 | 14:50 | 13:40 | 17:30 |
| <b>Tempo para perceber notificação</b>     | 04:47         | 00:07 | 3:17  | 3:22  | 00:07 | 1:02  |
| <b>Houve dificuldades na visualização?</b> | Não           | Não   | Não   | Não   | Não   | Não   |
| <b>Propôs solução?</b>                     | Sim           | Sim   | Sim   | Sim   | Sim   | Sim   |
| <b>Quantas soluções diferentes?</b>        | 1             | 1     | 1     | 2     | 1     | 1     |

O tempo médio de execução da tarefa deste cenário foi de 13:47. Para esta execução, era também importante verificar o tempo que os participantes demoraram para perceber a notificação de definição de contexto ativa. Através dos *logs* de utilização, foi possível verificar os tempos individuais e com isso calcular o tempo médio de 02:07. Tal tempo poderia ser bem reduzido se durante o treinamento antes da execução das tarefas fosse apresentado exatamente como ocorreria uma notificação, no entanto optou-se por apenas mencionar a existência de indicações de definições de contexto ativas e deixar para o estudo de observação verificar como uma notificação chamaria a atenção dos participantes.

Para os participantes P1, P3 e P4 o ícone de notificação para nova definição não foi de fácil percepção, demorando a notar sua presença. Isso pode ser verificado pelos tempos que estes participantes levaram para acessar a definição de contexto ativa através da notificação, com destaque para o participante P1 que teve o maior período entre todos os participantes. O participante P4 não considerou um problema grave não ter reparado imediatamente na notificação uma vez que era sua primeira utilização da plataforma e em usos futuros ele deveria ficar atento a essa funcionalidade. O participante P6 sugeriu trocar o ícone da notificação para algo que remeta a um alerta, como um sino.

Os participantes P1, P3, P5 e P6 destacaram a utilidade da visualização apresentada após a notificação para o cenário, sendo mais apropriada do que a visualização utilizada antes da mudança de contexto.

O participante P2 menciona a viabilidade de usar tanto a visualização presente na Figura 6.6 como a visualização da Figura 6.7, a fim de complementar a tarefa de alocação de pessoas considerando aspectos técnicos, no caso dos componentes em cada pacote, como aspectos sociais, para as colaborações entre dois membros da equipe.

O participante P4 gostaria de poder interagir com a visualização removendo nós e arestas, de forma a permitir uma visualização dinâmica durante a alocação de desenvolvedores, removendo os nós referentes a desenvolvedores já alocados. Outra sugestão dada por este participante diz respeito à representação de um nó, cujo tamanho poderia ser proporcional ao número total de colaborações do desenvolvedor relacionado a tal nó. Desta forma, seria mais fácil identificar os desenvolvedores que mais colaboraram na organização.

O participante P5 relata que na interação com os nós desejou que fosse possível selecionar e deslocar apenas um dos nós sem os demais se moverem juntamente (comportamento atual), o que poderia facilitar a identificação de relacionamentos. Como alternativa, ainda foi sugerido destacar um nó e seus relacionamentos a partir de sua seleção. Isso poderia se confirmar necessário com um número grande de desenvolvedores e relacionamentos, em que poderia ser difícil identificar certos relacionamentos entre dois desenvolvedores.

Dado que as soluções poderiam ser diversas, com interpretações bem diferentes, os participantes foram também instigados a mencionar caso pensassem em mais de uma solução para o problema do cenário. Apenas um participante (P4) formulou mais de uma solução. As soluções propostas por cada participante podem ser verificadas a seguir:

- **P1:** Alocar desenvolvedores com maior número de colaborações em componentes relacionados. Nó verde indicaria líder, alguém que tivesse muito conhecimento sobre o domínio do projeto;
- **P2:** Alocar desenvolvedores que já têm número grande de colaborações em componentes relacionados;
- **P3:** Promover colaboração, buscando alocar desenvolvedores que têm baixa colaboração com outros que possuem alta colaboração com demais membros. Buscar alocar em componentes relacionados os desenvolvedores dos nós marrom, roxo e azul com os desenvolvedores dos nós laranja e vermelho, a fim de não sobrecarregar o desenvolvedor do nó verde que colabora com diversos membros;
- **P4:** Programação por pares, (i) alocando para um mesmo componente desenvolvedores com maiores números de colaborações, ou (ii) alocando para um mesmo componente um desenvolvedor que tenha alto número de colaborações, considerado aqui como experiente, com outro desenvolvedor com baixo número de colaborações, considerado como pouco experiente;
- **P5:** Programação por pares, alocando em um mesmo componente desenvolvedores com números baixos de colaborações e aumentar o número de colaborações, alocando os demais desenvolvedores que tenham colaborado entre si nos mesmos componentes;
- **P6:** Buscar que os membros com poucas colaborações trabalhem com os desenvolvedores com números maiores de colaboração ou entre si. O foco de promoção da colaboração é especialmente sobre os membros que colaboraram pouco com outros.

Com uma visualização intencionalmente mais apropriada para o cenário descrito se comparada com a visualização anterior à notificação, esperava-se que os participantes não tivessem dificuldades em utilizar tal representação gráfica como suporte para propor soluções para a questão de baixa colaboração. Nesse sentido, os resultados indicam essa adequação, reforçada pelos relatos dos participantes de que a segunda visualização era mais adequada e eficiente para tomar decisões relativas ao cenário apresentado.

Com relação à notificação, assim que notaram seu alerta, os participantes conseguiram utilizá-la sem dúvida. A questão se coloca mesmo no tempo que alguns levaram para percebê-la. Nesse sentido, é importante considerar alterações na interface de modo a tornar essa percepção de mudança no contexto mais eficaz. Talvez fosse

interessante ampliar as dimensões do ícone de notificação, bem como alterar totalmente a figura do ícone. Dois participantes sugeriram a adoção de alertas mais intrusivos, que apareceriam no meio da tela, porém acredita-se que tal solução não seja adequada, uma vez que poderia atrapalhar a análise de um usuário que estivesse diante de uma visualização, interrompendo seu raciocínio.

### **6.7.5 Cenário D**

Conforme apresentado no Apêndice C, o Cenário D descreve uma situação de acompanhamento de tarefas durante o desenvolvimento de um projeto. Neste cenário, sua descrição não aponta diretamente que tipo de problema estaria ocorrendo, deixando tal análise para o participante verificar por meio de visualizações.

Uma notificação é apresentada para o participante no início da execução da tarefa relativa a este cenário, indicando uma definição de contexto ativa que permite a seleção entre dois focos de representação. Neste cenário, busca-se observar como os participantes usam os diferentes focos de representação a partir da indicação automática de mudança de contexto.

Para o foco “Distribuição”, a Figura 6.8 apresenta a visualização disponibilizada para o participante, exibindo todas as tarefas com status “aberta” através de bolhas, em conjunto de detalhes como a prioridade de uma tarefa por meio de cores e formas de comparar a idade de todas as tarefas, por meio do tamanho de cada bolha. Já para o foco de representação “Comparação de Valores Individuais”, a visualização presente na Figura 6.9 lista todos os desenvolvedores, exibindo dados referentes ao número de tarefas alocadas para cada membro na forma de um *dashboard* com um gráfico de barras e um gráfico de pizza sendo apresentados simultaneamente.

Enquanto o gráfico de bolhas para o foco “Distribuição” não disponibiliza nenhum tipo de interação adicional, o *dashboard* permite ao participante interagir selecionando uma barra referente a um desenvolvedor ou uma fatia relativa a um status de tarefa ao mover o cursor do *mouse* sobre um desses elementos na representação gráfica. Tal forma de seleção atualiza a exibição do outro gráfico no *dashboard*, atualizando o gráfico de pizza apenas com as tarefas de um desenvolvedor ou atualizando o gráfico de barras com as tarefas em um determinado status para todos os desenvolvedores. O elemento de legenda no *dashboard* também atualizada com os valores adequados de acordo com a interação do participante.

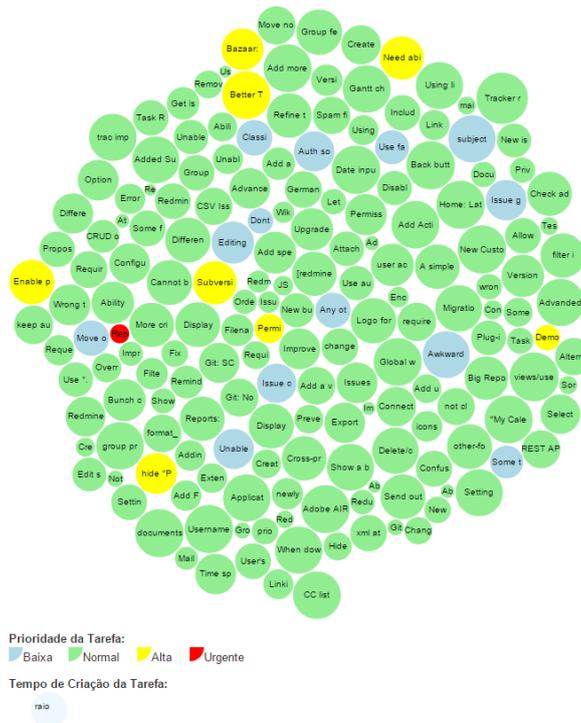


Figura 6.8 - Visualização com as tarefas abertas

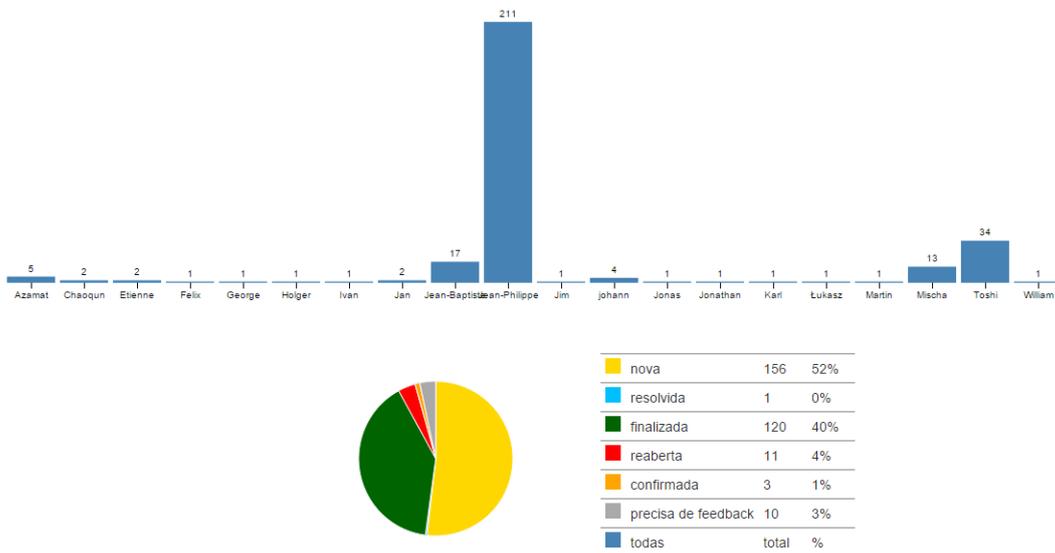


Figura 6.9 - Visualização com dados sobre as tarefas alocadas a desenvolvedores

Embora o gráfico de bolhas apresente algumas tarefas de prioridade alta abertas há um período razoável, o *dashboard* mostra uma quantidade muito grande de novas tarefas alocadas para um mesmo desenvolvedor, de forma desbalanceada se comparada com os demais membros da equipe. Essa percepção entre as informações disponíveis nas duas visualizações foi avaliada e também foi analisado como os participantes compararam a utilidade de cada visualização para a tomada de decisão no cenário em

destaque. A Tabela 6.5 lista os resultados obtidos para todos os participantes ao longo deste cenário.

**Tabela 6.5** - Resultado da execução da tarefa do Cenário D

|                                     | Participantes    |                  |       |                  |                  |       |
|-------------------------------------|------------------|------------------|-------|------------------|------------------|-------|
|                                     | P1               | P2               | P3    | P4               | P5               | P6    |
| <b>Tempo (minutos)</b>              | 16:44            | 12:14            | 19:54 | 24:26            | 22:46            | 23:01 |
| <b>Tempo no <i>dashboard</i></b>    | 05:08            | 07:54            | 02:55 | 12:50            | 04:47            | 03:37 |
| <b>Tempo no gráfico de bolhas</b>   | 09:42            | 04:16            | 14:11 | 09:54            | 17:09            | 17:59 |
| <b>Qual visualização mais útil?</b> | <i>Dashboard</i> | <i>Dashboard</i> | -     | <i>Dashboard</i> | <i>Dashboard</i> | -     |
| <b>Quantas soluções diferentes?</b> | 1                | 2                | 2     | 1                | 1                | 2     |

O tempo médio de execução da tarefa deste cenário foi de 19:51. O tempo médio em que os participantes permaneceram na visualização do *dashboard* foi de 06:12. Já o tempo médio em que os participantes permaneceram na visualização de gráfico de bolhas foi de 12:12. Como, além do tempo em que os participantes permaneceram analisando as duas visualizações, ainda havia interações na interface da plataforma, com seleção da notificação e escolha de focos de representação, a soma dos tempos individuais em cada visualização acaba sendo inferior ao tempo total de realização da tarefa no cenário.

O participante P2 considerou as duas visualizações disponíveis úteis, permitindo tomar decisões diferentes a partir de cada uma, porém a visualização do *dashboard* continha um problema mais destacado e urgente para o cenário descrito. Este participante também mencionou uma confusão pelo nome do foco, em que, para o foco de nome “Distribuição”, esperava como resultado uma visualização de matriz de dispersão. As descrições de cada foco estavam disponíveis por meio de *tooltips* na tela de seleção de foco, mas, talvez uma forma mais direta, que explicitasse o propósito de cada foco, fosse mais adequado.

Os participantes P3 e P6 consideraram as duas visualizações como complementares, igualmente úteis, uma vez que relatavam diferentes questões que mereciam atenção. Por isso, não consideraram uma visualização mais importante que a outra.

O participante P4 gostaria de alterações na forma de interação com a visualização na forma de *dashboard*, a fim de ser possível selecionar por meio de um clique um elemento do gráfico de barras ou do gráfico de pizza. Para a visualização do gráfico de bolhas, o participante P4 gostaria de outras informações disponíveis além da prioridade e da idade de uma tarefa, como complexidade. Já no *dashboard*, se fosse possível ter informações sobre a tarefa alocada a cada desenvolvedor, seria possível avaliar o esforço necessário para cada desenvolvedor. Este participante também sugeriu a existência de

uma visualização mais objetiva, na forma de uma tabela com dados sobre cada tarefa. Esta nova visualização poderia ser disponibilizada através de outro foco de representação.

O participante P6 gostaria de obter informações como o desenvolvedor alocado a cada tarefa e o número de dias desde a sua criação na visualização com o gráfico de bolhas.

Como as soluções poderiam ser diversas, com interpretações diversificadas, os participantes foram instigados a mencionar caso pensassem em mais de uma solução para a questão do cenário. As soluções propostas por cada participante podem ser verificadas a seguir:

- **P1:** Redistribuir as tarefas entre os desenvolvedores, de forma a ficar mais balanceada;
- **P2:** (i) Alocar tarefas para desenvolvedores que parecem ociosos, atualmente com poucas tarefas, e (ii) Verificar a demora na resolução de tarefas com prioridade alta;
- **P3:** (i) Redistribuir as tarefas entre os desenvolvedores, de forma a ficar mais balanceada, e (ii) Realocar tarefas de mais alta prioridade para desenvolvedores mais experientes, que conseguiram finalizar mais tarefas;
- **P4:** Redistribuir as tarefas entre os desenvolvedores, de forma a ficar mais balanceada;
- **P5:** Realocar tarefas de mais alta prioridade e idade para desenvolvedores mais experientes, que conseguiram finalizar mais tarefas;
- **P6:** (i) Redistribuir as tarefas entre os desenvolvedores, de forma a ficar mais balanceada, e (ii) Verificar a demora na resolução de tarefas com prioridade alta.

Os participantes, de forma geral, identificaram como problema principal um desbalanceamento entre o número de tarefas atribuídas aos membros da equipe. Sobre o desenvolvedor com número maior de tarefas alocadas, a maioria dos participantes considerou tal desenvolvedor como o mais experiente, uma vez que este também já finalizou um número grande de tarefas. Além disso, os participantes também puderam verificar tarefas de alta prioridade, que foram criadas há um tempo razoavelmente grande, necessitando investigar a causa da demora para sua resolução. Nesse sentido, as duas visualizações foram utilizadas para tomada de decisão sob aspectos diferentes para o cenário em questão.

A partir de tais resultados, nota-se a importância de se oferecer ao usuário da plataforma CAVE a opção com relação à perspectiva de análise para um dado problema. Com diferentes focos de representação para uma mesma definição de contexto, diferentes questões podem ser avaliadas, o que poderia provocar uma sobrecarga de informações em uma única visualização.

### **6.7.6 Vantagens**

Após a execução das tarefas referentes aos quatro cenários, os participantes fizeram uma avaliação geral de utilização da plataforma CAVE. Como vantagens todos os seis participantes relataram ter observado vantagens na adoção da plataforma.

Para análise de informações referentes a gerenciamento de projetos, os participantes mencionaram a utilidade de acompanhar por meio de visualizações os dados de situações de interesse. Os participantes P2 e P6, com maior experiência em termos de gerenciamento, destacaram os benefícios da plataforma para acompanhar informações relevantes para gerentes de projeto. O participante P2 destacou que os sistemas que podem servir de fonte de dados para a CAVE não costumam apresentar situações de forma tão rica, com visualizações para análise de um problema atual. O participante P6 ressaltou em especial as situações descritas nos cenários C e D para utilização por um gerente.

O participante P5 mencionou um caso em que a aplicação da plataforma poderia render mais benefícios. Projetos envolvendo equipes grandes, onde acompanhar o progresso de cada membro seja complicado, poderia deixar ainda mais claras as vantagens de adotar a CAVE. Mecanismos de suporte ao gerenciamento tendem a ser mais utilizados em situações assim, onde o contato com os desenvolvedores seja menos direto, como equipes grandes ou separadas em locais remotos, em um mesmo projeto.

### **6.7.7 Dificuldades**

Apesar das vantagens, os participantes também mencionaram dificuldades que notaram ao longo da execução das tarefas. Tais dificuldades servem de insumo para modificações na plataforma de forma a tornar sua adoção mais intuitiva e eficiente.

Os participantes P1, P3, P5 e P6 relataram dificuldades especiais ao analisar a visualização disponível para o Cenário B, visando obter informações úteis. Como não foi comentado com nenhum participante sobre o propósito de cada cenário, como no caso do Cenário B em que foi proposital a apresentação de uma visualização inadequada, tal

dificuldade já era esperada e confirma a necessidade de representações gráficas apropriadas para o contexto em vigência.

Além de dificuldades específicas relatadas nos resultados apresentados para cada cenário, para uma avaliação geral da plataforma, os participantes mencionaram dificuldades para interpretar algumas visualizações ou obter mais informações naquelas existentes. Nesse sentido, melhorar a descrição das legendas poderia significar um ganho para agilizar a visualização dos dados nos diferentes gráficos.

O participante P2 destacou uma possível dificuldade que pode surgir na utilização da funcionalidade de seleção manual de definições de contexto para um número grande de definições disponíveis. Nesse caso, um mecanismo de busca por definições seria de grande utilidade para usar a plataforma.

O participante P4 mencionou que o ícone referente a notificações é um pouco discreto. Nesse sentido, modificações para a forma de notificação podem facilitar a sua percepção.

Com relação ao desempenho, o participante P6 destacou certa lentidão para obter respostas do servidor. Como, rodando localmente, a plataforma apresentou tempos de resposta menores, considera-se que houve uma queda de desempenho quando o servidor em a que plataforma ficou hospedada precisou rodar as consultas na base de dados, a fim de popular as visualizações. Testes em diferentes configurações são importantes para avaliar essa dificuldade específica.

## **6.8 Validade**

A escolha dos participantes foi realizada de forma não aleatória, por conveniência, com pessoas conhecidas do experimentador. Os participantes têm perfil variado com relação à experiência nas áreas de gerenciamento de projeto e visualização de informação. Aliado a isso, o número reduzido de participantes também se coloca como outra restrição do estudo de observação conduzido.

Possíveis diferenças nas interpretações dos cenários também são possíveis, mas as respostas podem levar a contribuições sob diferentes óticas, o que é interessante para ampliar a aplicabilidade da abordagem.

Com suas limitações, esta avaliação não pode ser generalizada para outros grupos, porém foi útil para apontar deficiências da plataforma CAVE e oportunidades de melhoria para a abordagem. Como o estudo foi aplicado no contexto de software, não se pode estender o mesmo resultado para outros domínios. No entanto, como se espera uma

oportunidade de aplicação da abordagem de visualização sensível ao contexto para outras áreas, considera-se importante uma avaliação semelhante para cada domínio antes de uma adoção efetiva. Nesse sentido, pontos abordados neste estudo podem continuar sendo considerados, como experiência em visualização de informação, porém outros podem ser modificados ou adicionados, de forma a incluir especificidades de cada área.

## 6.9 Considerações Finais

O estudo conduzido foi imprescindível para uma avaliação da abordagem em conjunto com a implementação consolidada na plataforma CAVE. Dificuldades em entender a mecânica de sensibilidade ao contexto, bem como interpretar uma visualização apresentada como adequada, são fatores que podem comprometer a tomada de decisão por um usuário, no caso desta avaliação, no papel de um gerente de projeto.

Apesar de se considerar a falta, modificação ou emprego errôneo de uma técnica de visualização como uma pendência simples em um cenário amplo para adoção da abordagem, dependendo do impacto de tal técnica, a visualização pode se tornar um verdadeiro obstáculo em vez de ferramenta de suporte. Por isso, ao longo da análise de cada cenário, foi importante destacar as menções de cada participante com relação a necessidades que estes tiveram ao interagir com as representações gráficas.

De forma a consolidar todos as observações compiladas durante a execução do estudo, uma análise geral deve responder às seis questões colocadas como direcionadoras da avaliação como um todo. Com isso, é importante retomar as perguntas e agregar a elas os resultados, conforme apresentado a seguir:

- *Os participantes conseguiram identificar questões/problemas apontados em cada visualização?*
  - Nos diferentes cenários, os participantes conseguiram apontar questões relevantes para o contexto da organização. Reconhecer tais questões era primordial para cogitar ações a serem realizadas como medida necessária.
- *Os participantes conseguiram identificar questões/problemas após a mudança de contexto?*
  - Nos cenários em que foi promovida a mudança de contexto (cenários C e D), os participantes identificaram problemas rapidamente, o que ficou evidente através do acompanhamento de raciocínio por meio do protocolo *think aloud*. No caso do Cenário C, idêntico ao Cenário B, foi possível observar como a visualização recomendada permitiu um esclarecimento das questões

existentes e sua relação com as descrições do cenário. Ao contrário, antes da mudança de contexto e consequente mudança de visualização, apenas dúvidas eram relatadas, bem como suposições sobre elementos na visualização de modo a se adequarem às necessidades de informação dadas pelo cenário.

- *Quanto tempo levou para que os participantes conseguissem interpretar as visualizações antes da mudança de contexto?*
  - Para o Cenário B, considerado como estado anterior à mudança de contexto que ocorre na execução do Cenário C, o tempo médio decorrido para análise foi de 15:29. Este tempo pode ser considerado alto para os detalhes disponíveis na visualização. Entende-se que a maioria dos participantes utilizou um tempo considerável nesta etapa a fim de responder às pendências do cenário, imaginando que a visualização apresentada pudesse, de fato, colaborar diretamente para apoiar a tomada de decisão.
- *Quanto tempo levou para que os participantes conseguissem interpretar as visualizações após a mudança de contexto?*
  - Para o Cenário C, em que efetivamente pode-se comparar com uma análise antes da mudança de contexto, o tempo médio decorrido foi de 13:47. O resultado indica um tempo menor se comparado ao tempo antes da mudança de contexto, porém a diferença não é tão alta. No entanto, uma análise qualitativa neste ponto possibilitou verificar que tão logo a nova visualização foi apresentada, os participantes já conseguiram identificar elementos apropriados e começaram a desenvolver um raciocínio que remetesse ao cenário em foco.
- *Quais dificuldades os participantes tiveram e/ou observaram no uso da plataforma durante as tarefas realizadas?*
  - Por falta de familiaridade com algumas visualizações, como a de dependências entre classes, os participantes demoraram a compreender a relação entre as metáforas apresentadas e sua relação com os elementos mencionados na descrição do cenário. De forma geral, esclarecimentos com relação às legendas e formas de interação surgiram.
- *Mudanças de visualizações ajudam o usuário a entender melhor o problema atual?*

- Quando a mudança ocorre para uma visualização que apresente um problema atual, a identificação de tal problema ocorre de forma rápida. A apresentação de uma visualização no Cenário B que não destacava diretamente algum ponto que merecesse atenção só confundia a análise do participante. Da mesma forma, com as opções de focos de representação, como no Cenário D, foi possível verificar como a mudança de uma visualização para outra que destaca o mesmo problema, sob perspectiva diferente, pode ser útil para prover uma compreensão mais ampla do problema ou problemas que estejam ocorrendo em uma organização.

Um sucesso relativo pode ser considerado após o levantamento realizado. O destaque para situações que estejam relacionadas a problemas ou questões que mereçam atenção colabora de forma direta na tomada de decisão por parte de um *stakeholder* que acompanhe o processo de desenvolvimento de software. Por analisar o contexto e suas mudanças, a abordagem permite ao seu usuário acompanhar rapidamente pontos que necessitem de suporte. Com a oferta de focos para cada representação, uma maior cobertura para análise de tal questão é fornecida.

Com as oportunidades de melhoria ressaltadas ao longo do estudo, pode-se considerar essas sugestões e necessidades na hora de se customizar as regras de contexto, provendo novos elementos de visualização para o resultado gráfico. Além disso, modificações na interface da plataforma podem ser repensadas, como a falta de completude nas legendas em conjunto com a exibição das notificações como alertas de mudanças de contexto.

# CAPÍTULO 7 - CONCLUSÃO

## 7.1 Epílogo

Esta dissertação abordou a utilização de visualizações para ampliar a percepção em cenários multivariados, adotando conceitos de sensibilidade ao contexto para fornecer representações gráficas orientadas ao contexto mapeado desses cenários. Um exemplo para ambientes de desenvolvimento de software foi apresentado de forma a avaliar sua adoção durante projetos de software.

O objetivo da abordagem é promover a percepção em cenários em que diferentes atividades estejam sendo desenvolvidas. Entende-se que integrar o gerenciamento de contexto com técnicas de visualização de informação podem colaborar para destacar visualizações mais adequadas para cada situação, especialmente nas quais existem problemas em aberto ou questões de interesse. Nesse sentido, visualizações sensíveis ao contexto podem apoiar *stakeholders* em situações críticas, permitindo análises e indicando pontos passíveis de melhoria.

Visando o desenvolvimento da abordagem, a revisão da literatura identificou alguns conceitos importantes de sensibilidade ao contexto (Capítulo 2) e visualização de informação (Capítulo 3). Com o conhecimento reunido, foi possível verificar técnicas interessantes e necessárias para a construção da abordagem (apresentada no Capítulo 4) e sua implementação (Capítulo 5).

Um estudo de observação foi conduzido (Capítulo 6) com a finalidade de avaliar a adequação de diferentes visualizações para cenários de interesse de um *stakeholder* com perfil de gerente de projeto. A partir dos resultados obtidos, verificou-se que há indícios de que a adoção da abordagem proposta possa trazer benefícios a seus usuários, atuando como mecanismo de suporte e promovendo percepção durante o desenvolvimento de tarefas. Algumas modificações foram indicadas para tornar seu uso mais intuitivo, adicionando novas funcionalidades e aprimorando as existentes.

## 7.2 Contribuições

Esta dissertação apresentou uma abordagem de sensibilidade ao contexto aplicada a mecanismos de visualização de informação, provendo resultados para um determinado cenário, de forma a verificar a aplicabilidade da abordagem. O exemplo apresentado, avaliado por meio do estudo de observação, é voltado para cenários de desenvolvimento

de software, e serve como prova de conceito. Cabe destacar a possibilidade de generalização da abordagem, que é aplicável a diferentes domínios – o que requereria novas avaliações a fim de verificar sua adequação.

Entre as principais contribuições deste trabalho, é possível destacar:

- Definição de um modelo de contexto para a apresentação e atualização de visualizações, propondo um conjunto de elementos que podem influenciar na escolha de uma representação gráfica;
- Estudo de elementos de visualização e definição de um modelo de características para a área de visualização de informação (realizada em conjunto com um aluno de doutorado), possibilitando planejar composições de diferentes técnicas de visualização de forma a obter representações gráficas mais adequadas e estruturadas;
- Abordagem CAVE, que visa promover percepção ao gerenciar o contexto de um cenário e apresentar visualizações para apoiar a análise de questões de interesse de *stakeholders* em determinada situação;
- Um protótipo da plataforma CAVE, implementando e disponibilizando funcionalidades em acordo com a abordagem definida;

### 7.3 Limitações

Analisando as deficiências da abordagem, é importante destacá-las a fim de reconhecer limites de aplicação, bem como planejar modificações futuras. As limitações identificadas são:

- A abordagem atualmente não provê suporte a contexto dinâmico, uma vez que não incorpora automaticamente novas informações ou elementos de contexto. Nesse sentido, destaca-se um contexto estático na forma de seu modelo sendo mapeado a informações e definições de contexto suportadas na plataforma. A fim de incorporar novos elementos de contexto para serem gerenciadas na plataforma, é necessário incluir tais elementos a um modelo e usá-lo como entrada;
- Os modelos de contexto e de característica para visualização não foram validados. Apesar da proposta de contexto baseada em aspectos que possam impactar visualizações e da construção do modelo de características a partir de elementos extraídos da literatura, é necessário avaliar a validade e deficiência dos modelos, a fim de torná-los mais abrangentes e menos propensos a incoerências.

- A avaliação realizada apresenta restrições, tanto em representatividade pelo número de participantes como nas características dos cenários escolhidos. Avaliações envolvendo cenários mais complexos, diferentes *stakeholders* (além de gerentes de projeto) e um número maior de participantes, com perfis variados, poderiam prover uma evidência mais significativa para a aplicabilidade da plataforma. Além disso, outros estudos em outros domínios podem auxiliar na avaliação da adequabilidade da abordagem proposta para outros domínios.

## 7.4 Perspectivas Futuras

Identifica-se na abordagem CAVE um ponto de partida para ampliar a percepção em diferentes cenários. Nesse sentido, é importante destacar que a abordagem pode ser ampliada, de forma com que sua eficiência seja destacada. Trabalhos futuros podem ser desenvolvidos em continuações deste trabalho. Dentre tais perspectivas, é possível listar:

- Avaliação do modelo de contexto e do modelo de características para visualização por especialistas, verificando a completude e a adequação dos elementos compilados;
- Incorporação da modelagem de contexto às funcionalidades da plataforma CAVE, permitindo ao administrador adicionar e editar todos os elementos de contexto diretamente na plataforma. Com isso, a abordagem não dependeria do ambiente Odyssey para um formato específico de modelo de contexto;
- Disponibilizar um conjunto mais amplo de elementos de visualização implementados na plataforma, possibilitando uma maior flexibilidade de customização das regras de contexto e, conseqüentemente, das visualizações;
- Adicionar permissão de acesso na plataforma CAVE, de modo que seja possível indicar a quais definições de contexto um usuário teria acesso, tanto para visualizações como para notificações. É possível também pensar em um mapeamento não por usuário, mas por papel que este desempenhe, onde um conjunto de definições de contexto só poderão ser analisadas com visualizações por certos *stakeholders*;
- Prover insumo para a abordagem APPRAiSER (OLIVEIRA, 2014) (SCHOTS, 2014), voltando o foco para informações e definições de contexto de interesse para a área de reutilização de software.

## REFERÊNCIAS BIBLIOGRÁFICAS

- AMAR, R., EAGAN, J., STASKO, J. "Low-level components of analytic activity in information visualization". *IEEE Symposium on Information Visualization (INFOVIS)*, pp. 111-117, Minneapolis, Minnesota, USA, 2005.
- APACHE SOFTWARE FOUNDATION. Maven SCM API. *Maven SCM API - About*, 2014. Disponível em: <http://maven.apache.org/scm/maven-scm-api/>. Acesso em: 30 Nov 2014.
- BAECKER, R. M., MARCUS, A., *Human Factors and Typography for More Readable Programs*. Reading, Massachusetts, Addison-Wesley, 1990.
- BALDAUF, M., DUSTDAR, S., ROSENBERG, F. "A Survey on Context-Aware Systems", *International Journal of Ad Hoc and Ubiquitous Computing*, v. 2, n. 4, pp. 263-277, 2007.
- BALL, T., EICK, S. G. "Software Visualization in the Large", *Computer*, v. 29, n. 4, pp. 33-43, 1996.
- BASILI, V., CALDIERA, G., ROMBACH, H. "Goal Question Metric Paradigm". In: MARCINIAK, J. J. *Encyclopedia of Software Engineering*, New Jersey, John Wiley & Sons, pp. 528-532, 1994.
- BECK, F., BURCH, M., DIEHL, S. "Matching application requirements with dynamic graph visualization profiles", *17th International Conference on Information Visualisation*. pp. 11-18, London, UK, 2013.
- BERTINI, E. *Dealing with Clutter in Information Visualization*. Ph.D. Report, Roma, Italy, 2004.
- BLOIS, A. P. T. B., DE OLIVEIRA, R. F., MAIA, N., WERNER, C., BECKER, K. "Variability modeling in a component-based domain engineering process", *Reuse of Off-the-Shelf Components*, pp. 395-398, 2006.
- BOSTOCK, M. D3.js – Data Driven Documents. *D3 Data Driven Documents*, 2013. Disponível em: <http://d3js.org/>. Acesso em: 31 Agosto 2014.
- BRÉZILLON, P., POMEROL, J. C. "Contextual Knowledge Sharing and Cooperation in Intelligent Assistant Systems", *Le Travail Humain*, v. 62, n. 3, pp. 223-246, 1999.
- BROWN, M. H. "Algorithm animation", *ACM Distinguished Dissertations*, Cambridge, MIT Press, 1988.

- CARD, S. K., MACKINLAY, J. D., SHNEIDERMAN, B. *Readings in Information Visualization: Using Vision to Think*. San Francisco, Morgan Kaufman Publishers Inc., 1999.
- CHEN, C. *Information Visualization: Beyond the Horizon*. 2 ed. London, Springer, 2006.
- CHEN, G., KOTZ, D. *A Survey of Context-Aware Mobile Computing Research*. Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.
- CHI, E. H. "A taxonomy of visualization techniques using the data state reference model", *IEEE Symposium on Information Visualization (INFOVIS)*, pp. 69-75, Salt Lake City, USA, 2000.
- CHI, E. H., RIEDL, J. T. "An operator interaction framework for visualization systems", *IEEE Symposium on Information Visualization*, pp. 63-70, Durham, USA, 1998.
- CHILIMBI, T., BALL, T., EICK, S. G., LARUS, J. R. "Stormwatch: A toll for visualizing memory system protocols". In: *Proceedings of the IEEE/ACM SC95 Conference (Supercomputing)*, pp. 38-38, San Diego, California, 1995.
- CLEMENTS, P., NORTHROP, L. *Software Product Lines: Practices and Patterns*. 3 ed. Boston, Addison-Wesley Professional, 2001.
- DEY, A. K., ABOWD, G. D. "Towards a Better Understanding of Context and Context-Awareness", *1st International Symposium on Handheld and Ubiquitous Computing*, pp. 304-307, Karlsruhe, Germany, 1999.
- DIEHL, S. *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*. 1 ed. London, Springer, 2007.
- DIX, A., FINLAY, J., ABOWD, G. D., BEALE, R. *Human-computer interaction*. 3 ed. Harlow, Pearson Prentice Hall, 2004.
- DOURISH, P., BELLOTTI, V. "Awareness and coordination in shared workspaces". In: *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '92)*, pp. 107-114, New York, 1992.
- EICK, S. G., KARR, A. F. "Visual Scalability", *Journal of Computational and Graphical Statistics*, v. 11, n. 1, pp. 22-43, 2002.
- FERNANDES, P., WERNER, C., MURTA, L. "Feature Modeling for Context-Aware Software Product Lines", *20th International Conference on Software Engineering and Knowledge Engineering (SEKE'08)*. pp. 758-763, Redwood City, San Francisco Bay, USA, 2008.

- FERNANDES, P., WERNER, C., TEIXEIRA, E. "An Approach for Feature Modeling of Context-Aware Software Product Line", *Journal of Universal Computer Science*, v. 17, n. 5, pp. 807-829, 2011.
- FEW, S. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. 1 ed. Oakland, Analytics Press, 2009.
- GIBSON, J. J. *The Ecological Approach to Visual Perception*. Boston, Houghton Mifflin, 1979.
- GRAMBOW, G., OBERHAUSER, R., REICHERT, M. "Knowledge Provisioning: A Context-Sensitive Process-Oriented Approach Applied to Software Engineering Environments", *7th International Conference on Software and Data Technologies*, pp. 506-515, Rome, Italy, 2012.
- GUTWIN, C., PENNER, R., SCHNEIDER, K. "Group awareness in distributed software development". In: *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '04)*, pp. 72-81, New York, 2004.
- HURLBURT, A. *Layout: o design da página impressa*. São Paulo, NBL Editora, 1986.
- LATOZA, T. D., VENOLIA, G., DELINE, R. "Maintaining mental models: A study of developer work habits". In: *Proceedings of the 28th International Conference on Software Engineering (ICSE '06)*, pp. 492-501, New York, 2006.
- LEE, K., KANG, K. C., LEE, J. "Concepts and guidelines of feature modeling for product line software engineering", *Software Reuse: Methods, Techniques, and Tools*, pp. 62-77, Austin, USA, Springer Berlin Heidelberg, 2002.
- LEWIS, C. *Using the "thinking-aloud" method in cognitive interface design*. Yorktown Heights, IBM T.J. Watson Research Center, 1982.
- LI, J., ZHANG, D., ZHANG, P. "Supporting Dynamic Situation Awareness in Online Group Discussion: A Visualization Approach", *46th Hawaii International Conference on System Sciences (HICSS)*, pp. 470-479, Manoa, Hawaii, USA, 2013.
- MACKINLAY, J. D. "Automating the Design of Graphical Presentations of Relational Information", *ACM Transactions on Graphics (TOG)*, v. 5, n. 2, pp. 110-141, 1986.
- MANN, T. M. "Visualization of WWW-search results". In: *Proceedings of the Tenth International Workshop on Database and Expert Systems Applications (DEXA 99)*, pp. 264-268, Florence, Italy, 1999.

- MÜLLER, W., SCHUMANN, H. "Visualization methods for time-dependent data-an overview". In: *Proceedings of the 35th Winter Simulation Conference*, pp. 737-745, New Orleans, USA, 2003.
- NUNES, V. T., SANTORO, F. M., BORGES, M. R. S. "Um Modelo para Gestão de Conhecimento Baseado em Contexto", *XXVII Simpósio Brasileiro de Sistemas Colaborativos (SBSC)*, pp. 69-82, Rio de Janeiro, 2007.
- OLIVEIRA, M. S., 2011, *PREViA: Uma Abordagem para a Visualização da Evolução de Modelos de Software*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- OLIVEIRA, M. S., 2014, *On the Use of Visualization for Supporting Software Reuse*. Exame de Qualificação, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- PRIETO-DIAZ, R., ARANGO, G. "Domain Analysis Concepts and Research Directions". In: PRIETO-DIAZ, R., ARANGO, G. *Domain Analysis and Software Systems Modeling*, Los Alamitos, USA, IEEE Computer Society Press, 1991.
- QUEIROZ, A. R., SCHOTS, M., WERNER, C. *Apoio ao Gerenciamento de Projetos por meio da Análise Visual de Dados de Cenários*. Iniciação Científica, (Graduação em Engenharia de Computação e Informação) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.
- RAMESH, B., TIWANA, A. "Supporting collaborative process knowledge management in new product development teams", *Decision support systems*, v. 27, n. 1, pp. 213-235, 1999.
- REUSE. *Odyssey. Equipe de Reutilização de Software*, 2014. Disponível em: <http://reuse.cos.ufrj.br/odyssey>. Acesso em: 11 Nov 2014.
- ROBERTSON, P. K. "A methodology for choosing data representations", *IEEE Computer Graphics and Applications*, v. 11, n. 3, pp. 56-67, 1991.
- SCHILIT, B., ADAMS, N., WANT, R. "Context-aware computing applications". In: *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 85-90, Santa Cruz, California, IEEE Computer Society Press, 1994.
- SCHNEIDERMAN, B. "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations", *IEEE Conference on Visual Languages*, pp. 336-343, Columbus, USA, 1996.
- SCHOTS, M. "On the Use of Visualization for Supporting Software Reuse". *36th International Conference on Software Engineering (ICSE), Doctoral Symposium*, pp. 694-697, Hyderabad, India, 2014.

- SCHOTS, M., VASCONCELOS, R., QUEIROZ, A. R., WERNER, C. *Organization of Visualization Knowledge for Supporting the Choice of Software Visualizations*. Relatório Técnico, COPPE/UFRJ, Rio de Janeiro, 2015 (to appear).
- SCHOTS, M., VASCONCELOS, R., WERNER, C. *A Quasi-Systematic Review on Software Visualization Approaches for Software Reuse*. Relatório Técnico ES-748/14, COPPE/UFRJ, Rio de Janeiro, 2014.
- SCHOTS, M., WERNER, C. *Explorando o Intangível: Um Panorama da Visualização de Software e suas Aplicações*. Tutorial III Brazilian Congress on Software: Theory and Practice (CBSOFT 2012), Natal, Brasil, 2012.
- SCHOTS, M., WERNER, C., MENDONÇA, M. "Awareness and comprehension in software/systems engineering practice and education: trends and research directions", *26th Brazilian Symposium on Software Engineering (SBES)*, pp. 186-190, Natal, 2012.
- SCHULZ, H. J., NOCKE, T., HEITZLER, M., SCHUMANN, H. "A design space of visualization tasks", *IEEE Transactions on Visualization and Computer Graphics*, Piscataway, v. 19, n. 12, pp. 2366-2375, 2013.
- SHERIF, K., VINZE, A. "Barriers to adoption of software reuse: A qualitative study", *Information & Management*, v. 41, n. 2, pp. 159-175, 2003.
- SHORTER Oxford English Dictionary. 3 ed. v. 2, Oxford, UK: Oxford University Press, 1972.
- SILVA, M. A., 2012, *Linha de Produtos para Visualização de Software: Uma Infraestrutura para Apoiar Atividades de Compreensão por Meio da Construção de Mecanismos de Visualização*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.
- SPÍNOLA, R. O., MASSOLAR, J., TRAVASSOS, G. H. "Characterizing Ubicomp Software Projects through a Checklist". In: *Proceeding of I Workshop on Pervasive and Ubiquitous Computing (WPUC 2007)*, v. 1, pp. 1-6, Gramado, 2007.
- TELEA, A., VOINEA, L., SASSENBURG, H. "Visual Tools for Software Architecture Understanding: A Stakeholder Perspective", *IEEE Software*, v. 27, n. 6, pp. 46-53, 2010.
- TORY, M., MOLLER, T. "Rethinking visualization: A high-level taxonomy". In: *Proceedings of the IEEE Symposium on Information Visualization*, pp. 151-158, Austin, USA, 2004.

- TREUDE, C., STOREY, M. "Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds", *ACM/IEEE 32nd International Conference on Software Engineering*, pp. 365-374, Cape Town, South Africa, 2010.
- VASCONCELOS, R., SCHOTS, M., WERNER, C. "An information visualization feature model for supporting the selection of software visualizations", *22nd International Conference on Program Comprehension, Early Research Achievements track*, pp. 122-125, Hyderabad, India, 2014.
- VIEIRA, V., 2008, *CEManTIKA: A Domain-Independent Framework for Designing Context-Sensitive Systems*. Tese de D.Sc., Centro de Informática, UFPE, Pernambuco, Brasil.
- VOIGT, M., PIETSCHMANN, S., GRAMMEL, L., MEIBNER, K. "Context-aware Recommendation of Visualization Components", *The Fourth International Conference on Information, Process, and Knowledge Management (eKNOW 2012)*, pp. 101-109, Valencia, Spain, 2012.
- WARE, C. *Information Visualization: Perception for Design*. 2 ed. San Francisco, Elsevier, 2004.
- WEISER, M. "The Computer for the 21st Century", *Scientific American*, v. 265, n. 3, pp. 94-104, 1991.
- YI, J. S., AH KANG, Y., STASKO, J. T., JACKO, J. "Toward a deeper understanding of the role of interaction in information visualization", *IEEE Transactions on Visualization and Computer Graphics*, v. 13, n. 6, pp. 1224-1231, Piscataway, New Jersey, USA, 2007.

# APÊNDICE A - FORMULÁRIO DE CONSENTIMENTO

## **Estudo**

Este estudo visa caracterizar a viabilidade do uso da plataforma CAVE para prover visualizações sensíveis ao contexto e auxiliar no desempenho de atividades de gerenciamento de projetos de software.

## **Idade**

Eu declaro ter mais de 18 anos de idade e concordar em participar de um estudo conduzido por Renan Ribeiro de Vasconcelos na Universidade Federal do Rio de Janeiro.

## **Procedimento**

Este estudo acontecerá em uma única sessão, composta de quatro situações. Em todas as tarefas, os participantes deverão executar um conjunto de passos e responder a uma série de questionamentos relacionados às atividades realizadas. Eu entendo que, uma vez que o experimento tenha terminado, os trabalhos que desenvolvi serão estudados visando entender a eficiência dos procedimentos e a abordagem proposta.

## **Confidencialidade**

Toda informação coletada neste estudo é confidencial, e meu nome não será divulgado. Da mesma forma, me comprometo a não comunicar os meus resultados enquanto não terminar o estudo, bem como manter sigilo das situações e documentos apresentados e que fazem parte do experimento.

## **Benefícios e liberdade de desistência**

Eu entendo que os benefícios que receberei deste estudo são limitados ao aprendizado do material que é distribuído e apresentado. Eu entendo que sou livre para realizar perguntas a qualquer momento ou solicitar que qualquer informação relacionada a minha pessoa não seja incluída no estudo. Eu entendo que participo de livre e espontânea vontade com o único intuito de contribuir para o avanço e desenvolvimento de técnicas e ferramentas para as áreas de visualização de software e sensibilidade ao contexto.

### **Pesquisador responsável**

Renan Ribeiro de Vasconcelos

Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ

### **Pesquisador colaborador**

Marcelo Schots de Oliveira

Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ

### **Professor responsável**

Prof<sup>a</sup>. Cláudia Maria Lima Werner

Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ

**Nome (em letra de forma):** \_\_\_\_\_

**Assinatura:** \_\_\_\_\_ **Data:** \_\_\_\_\_

# APÊNDICE B - QUESTIONÁRIO DE CARACTERIZAÇÃO

## 1. Formação acadêmica

- Doutorado     Doutorando  
 Mestrado     Mestrando  
 Graduação     Graduando

Em que área(s)? \_\_\_\_\_

Ano de ingresso: \_\_\_\_\_ Ano de conclusão (ou previsão de conclusão): \_\_\_\_\_

Especialização. Em que área(s)? \_\_\_\_\_

## 2. Formação geral

2.1) Qual é a sua experiência com desenvolvimento de software? (Marque aquele item que melhor se aplica)

- já li material sobre desenvolvimento de software.  
 já participei de um curso sobre desenvolvimento de software.  
 nunca desenvolvi software.  
 tenho desenvolvido software para uso próprio.  
 tenho desenvolvido software como parte de uma equipe, relacionada a um curso.  
 tenho desenvolvido software como parte de uma equipe, na indústria.

2.2) Por favor, explique sua resposta. Inclua o número de semestres ou número de anos de experiência relevante em desenvolvimento de software. (Ex.: “Eu trabalhei por 2 anos como programador de software na indústria”)

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

2.3) Por favor, indique o grau de sua experiência nesta seção seguindo a escala de 5 pontos abaixo:

- 0 = nenhum  
1 = estudei em aula ou em livro  
2 = pratiquei em projetos em sala de aula  
3 = usei em projetos pessoais  
4 = usei em projetos na indústria

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 2.1.1) Gerenciamento de Projetos de Software                    | 0 | 1 | 2 | 3 | 4 |
| 2.1.2) Análise de Visualizações de Informação/ <i>Dashboard</i> | 0 | 1 | 2 | 3 | 4 |

# APÊNDICE C - TAREFAS

## Introdução

A empresa ABC, desenvolvendo software há alguns anos, busca melhorar o gerenciamento dos projetos desenvolvidos internamente. Nota-se que há uma dificuldade em reconhecer problemas e propor soluções prontamente, o que impacta no processo de desenvolvimento. Com a finalidade de tratar essa deficiência, a empresa optou recentemente por adotar a plataforma CAVE, baseada em visualizações orientadas ao contexto da organização.

As visualizações são mapeadas para situações previamente reconhecidas como relevantes para o contexto da organização. As situações destacam questões importantes que devem ser tratadas. As situações possíveis são listadas a seguir:

- Desenvolvimento concorrente
  - Projetos sendo implementados por diversos desenvolvedores ao mesmo tempo
- Projeto modificado
  - Várias modificações foram realizadas na última versão de um projeto
- Alta dependência entre classes
  - Classe com alto número de dependências
- Baixa colaboração
  - Apenas um desenvolvedor implementando vários componentes
- Excesso de tarefas em aberto
  - Muitas tarefas em aberto para serem realizadas no projeto

Como um gerente de projeto, pede-se que analise as visualizações apresentadas de acordo com dados atuais. A partir dessa análise, pede-se que sejam propostas alternativas para lidar com cada questão.

**Novas situações podem ocorrer a qualquer instante, então pede-se também que o gerente esteja atento a isso.**

Os cenários a seguir descrevem as características relevantes que devem ser consideradas ao se selecionar uma situação e analisar uma visualização.

## Cenário A

### Descrição:

Neste cenário, o projeto de software sendo implementado possui 16 classes. Existe uma preocupação por parte do gerente com relação ao acoplamento do sistema sendo desenvolvido, uma vez que isso servirá de indicador para alocar os desenvolvedores para a manutenção de determinadas classes. Desenvolvedores mais experientes serão alocados para manter classes com um número maior de dependências. Pretende-se verificar qual a classe mais importante em termos de responsabilidade, através de uma análise para identificar qual classe é alvo de dependências de outras com mais frequência.

## Cenário B

### Descrição:

Neste cenário, o projeto gerenciado apresenta 120 componentes já implementados e a equipe de desenvolvimento conta com 6 membros. A fase atual do projeto diz respeito à manutenção dos componentes existentes.

A colaboração entre os desenvolvedores não é uma prática comum na organização, sendo feita de forma esporádica pelos profissionais.

Como houve uma realocação significativa de desenvolvedores nos projetos da organização, os membros da equipe atual do projeto ainda não se conhecem. No entanto, alguns deles já trabalharam nos mesmos componentes reutilizados de projetos legados, compartilhando conhecimento em determinados componentes.

A partir da visualização apresentada, você precisa pensar em como ampliar a produtividade da equipe por meio de colaboração.

## Cenário C

### Descrição:

Este cenário é idêntico ao Cenário B, descrito anteriormente. O projeto gerenciado apresenta 120 componentes já implementados e a equipe de desenvolvimento conta com 6 membros. A fase atual do projeto diz respeito à manutenção dos componentes existentes.

A colaboração entre os desenvolvedores não é uma prática comum na organização, sendo feita de forma esporádica pelos profissionais.

Como houve uma realocação significativa de desenvolvedores nos projetos da organização, os membros da equipe atual do projeto ainda não se conhecem. No entanto, alguns deles já trabalharam nos mesmos componentes reutilizados de projetos legados, compartilhando conhecimento em determinados componentes.

A partir da visualização apresentada, você precisa pensar em como ampliar a produtividade da equipe por meio de colaboração.

## Cenário D

### Descrição:

Neste cenário, você conta com uma equipe de 20 desenvolvedores disponíveis.

Um sistema de gerenciamento de tarefas (*issue tracker*) foi adotado recentemente para controlar a distribuição e o progresso de tarefas relativas a um projeto de software.

Você, como gerente, é responsável por atribuir tarefas a cada desenvolvedor por meio do sistema de gerenciamento de tarefas.

No entanto, você viajou a trabalho por algumas semanas e irá acompanhar o progresso do projeto exclusivamente por meio da plataforma CAVE, sem contato direto com os membros da equipe.

# APÊNDICE D - QUESTIONÁRIO DE OBSERVAÇÃO

Cenário \_\_\_\_\_

Horário de Início: \_\_\_\_\_

Horário de Término: \_\_\_\_\_

1) Qual problema você identificou neste cenário?

---

---

---

---

---

2) Com base no uso da plataforma CAVE, qual solução proporia para tratar o problema identificado?

---

---

---

---

---

---

---

---

---

## **APÊNDICE E - QUESTIONÁRIO DE OBSERVAÇÃO GERAL**

**1)** Você observou vantagens em utilizar a CAVE neste estudo? Especifique.

---

---

---

---

---

**2)** Você sentiu dificuldades em utilizar a CAVE neste estudo? Especifique.

---

---

---

---

---

**Observações gerais (sugestões e críticas sobre o uso da ferramenta):**

---

---

---

---

---

---

---

---

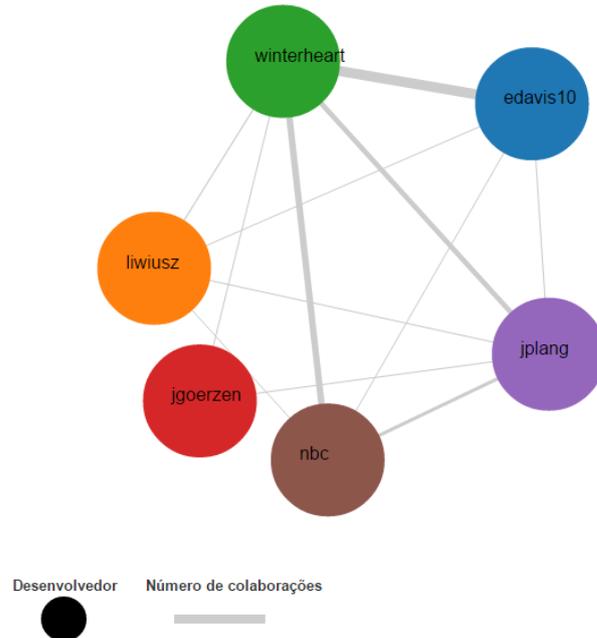


### Cenário C:

Página com a visualização referente à definição de contexto “Desenvolvimento Concorrente”;

Notificação para definição de contexto “Baixa Colaboração”;

Visualização de grafos, com os relacionamentos entre os desenvolvedores (participaram da implementação do mesmo componente anteriormente).



### Cenário D:

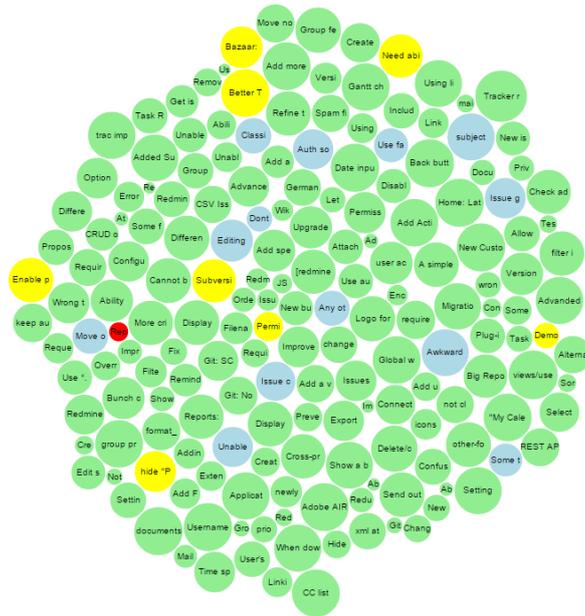
Página inicial, sem nenhuma notificação;

Notificação para definição de contexto “Excesso de tarefas em aberto”;

Página para seleção de foco de representação, com opções “Distribuição” e “Comparação de Valores Individuais”;

Opções de visualizações:

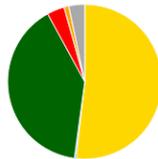
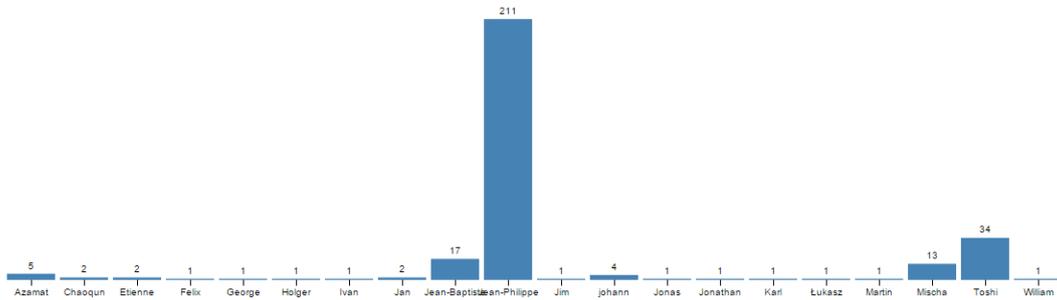
- Gráfico de bolhas, com todas as tarefas em aberto;
- *Dashboard*, com os detalhes das tarefas por desenvolvedor.



Prioridade da Tarefa:  
■ Baixa ■ Normal ■ Alta ■ Urgente

Tempo de Criação da Tarefa:

1s 0



|                     |       |     |
|---------------------|-------|-----|
| nova                | 156   | 52% |
| resolvida           | 1     | 0%  |
| finalizada          | 120   | 40% |
| reaberta            | 11    | 4%  |
| confirmada          | 3     | 1%  |
| precisa de feedback | 10    | 3%  |
| todas               | total | %   |

## APÊNDICE G - GABARITO DAS TAREFAS

### **Cenário A:**

O participante identifica a classe que é referenciada mais vezes pelas demais classes, ou seja, a classe Mídia.

### **Cenário B:**

O participante analisa visualização, mas esta não é adequada para tratar o problema de baixa colaboração descrito no cenário. Acredita-se que, se o participante der alguma resposta, esta provavelmente não será a mais adequada, devido a aquele não ter uma visualização de apoio adequada para a tomada de decisão.

### **Cenário C:**

O participante verifica a participação dos desenvolvedores nos mesmos componentes e sugere como poderia ser promovida a colaboração entre os membros (quem poderia colaborar com quem devido a trabalharem nos mesmos componentes e, por este motivo, potencialmente compartilharem conhecimento). Realocações de membros da equipe e/ou técnicas para facilitar a colaboração, como programação por pares, poderiam ser decisões tomadas pelo participante. Espera-se que, neste item, a resposta seja mais precisa ou mais próxima da realidade que no item anterior.

### **Cenário D:**

O gráfico de bolhas mostrará uma quantidade grande de tarefas em aberto, mas somente por meio do *dashboard* será possível verificar que apenas um membro da equipe não vem atualizando o status das tarefas ou não vem obtendo progresso, com diversas tarefas novas que foram alocadas pelo gerente.

Espera-se, portanto, que o participante tome a decisão baseado na visualização do *dashboard*.