2
Fundamentação Teórica

Este capítulo apresenta os conceitos que fundamentam essa pesquisa. A Seção 2.1 conceitua linhas de produto de software e aponta os benefícios de sua adoção. Na Seção 2.2, é apresentado o conceito de aprendizagem colaborativa com suporte computacional, além de conceituar scripts de colaboração e descrever o framework para a criação de scripts usado nessa tese. Em seguida, na Seção 2.3 são apresentados os elementos que constituem a BPMN, que foi usada para representar os scripts de colaboração. A Seção 2.4 apresenta o Modelo 3C de Colaboração e seus recursos para o desenvolvimento de groupware, o RUP 3C-Groupware e o Groupware Workbench. A Seção 2.5 define o conceito de desenvolvimento dirigido por modelos usando (Model Driven Development – MDD) na implementação do protótipo desenvolvido nesta tese. A Seção 2.6 define End-user programming, conceito usado em conjunto com o MDD que permite com que o usuário final do protótipo possa derivar seu próprio groupware a partir da LPS descrita nessa tese. Por fim, na Seção 2.7 são apresentados trabalhos relacionados a essa pesquisa.

2.1. Linhas de Produto de Software (LPS)

Uma linha de produto de software (LPS) é um conjunto de sistemas computacionais compartilhando um conjunto de características comuns, gerenciáveis que satisfazem as necessidades específicas de um segmento de mercado particular (família de software) e que são desenvolvidos a partir de um conjunto comum de artefatos de forma sistemática [2].

As LPS objetivam prover, a custos reduzidos, software customizados. A adoção das LPS provê benefícios [27], tais como:

- **Customização em massa.** Consiste na produção em larga escala de software adaptado às necessidades individuais dos usuários.
- **Tempo de entrega reduzido.** Apesar do tempo de entrega de produtos derivados de LPS ser inicialmente alto, devido ao esforço inicial no desenvolvimento dos artefatos comuns que farão parte do núcleo da LPS,

PUC-Rio - Certificação Digital Nº 0812625/CA

esse tempo é reduzido com o reuso desses artefatos para entrega de novos produtos.

- **Baixo custo de desenvolvimento.** Da mesma maneira que o tempo de entrega, o custo inicial do desenvolvimento de LPS é alto (investimento inicial), porém o autor argumenta que o investimento inicial é justificado a partir da entrega do terceiro produto da LPS aproximadamente.
- **Melhoria na qualidade dos produtos.** Os artefatos da LPS são revistos e testados em diversos produtos, e precisam funcionar bem em todos eles. Esses testes intensivos implicam em uma chance maior de detectar e corrigir eventuais problemas, aumentando assim a qualidade de todos os produtos da linha.

Os benefícios da adoção de LPS identificados são conseqüência do reuso intensivo e sistemático de software. Os produtos são desenvolvidos a partir de um conjunto de artefatos de software de forma sistemática, em contraste com o desenvolvimento individual de software, a partir do zero, ou através do reuso de software de forma arbitrária [2]. Destaca-se, porém, que a abordagem de LPS pode ser confundida com outras abordagens no desenvolvimento de software [2, 28], tais como:

- **Reuso fortuito de pequena granularidade.** Essa abordagem trata do reuso de pequenos pedaços de software geralmente escritos em bibliotecas de funções, módulos, objetos ou pequenos componentes. Na abordagem de LPS, o reuso é planejado, possibilitado e forçado, ao contrário do reuso oportunístico. A base de artefatos inclui aqueles que possuem o maior custo de desenvolvimento no caso de sistemas desenvolvidos isoladamente. Esses artefatos são projetados para serem reusados em mais diversos sistemas.
- **Desenvolvimento de software individual com reuso.** Essa abordagem trata do reuso oportunístico no desenvolvimento de software. No desenvolvimento individual de software, é comum que o desenvolvedor já tenha desenvolvido soluções semelhantes que podem ser aplicadas ao novo software em desenvolvimento. Nesse caso, o desenvolvedor geralmente “copia e cola”, se apropriando do conhecimento adquirido em experiências anteriores de desenvolvimento. Como resultado tem-se sistemas totalmente diferentes e não sistemas construídos a partir da mesma base. Em LPS, os artefatos são projetados explicitamente para o reuso, e a linha de produtos é tratada como um todo, não como vários produtos que são vistos e mantidos separadamente. Cada produto é simplesmente resultado da composição e adequação dos artefatos do núcleo da linha de produtos e de,

PUC-Rio - Certificação Digital Nº 0812625/CA

eventualmente, uma pequena coleção de artefatos adicionais exclusivos para esse produto.

- **Apenas desenvolvimento baseado em componentes ou serviços.** Embora os produtos nas LPS sejam desenvolvidos através da composição de componentes, alguns dos quais podem ser também Web Services, todos esses componentes devem ser especificados pela arquitetura de linha de produtos. Além disso, os componentes são montados de forma prescrita, o que inclui embutir mecanismos de variação nos componentes para usá-los em produtos específicos. Em uma linha de produtos, a forma genérica do componente é desenvolvida e mantida na base de artefatos essenciais (núcleo da linha de produtos). No caso do desenvolvimento baseado em componentes, se houver necessidade de alguma variação no produto, isso implica em escrever mais código, e essas variações geralmente são mantidas separadamente. No desenvolvimento baseado em componentes falta, também, os aspectos de gestão técnica e organizacional que são tão importantes para o sucesso de um produto de software. Nesta pesquisa, foi usado o desenvolvimento baseado em componentes para a implementação dos artefatos da linha de produtos, conforme descrito no Capítulo 3.
- **Apenas uma arquitetura reconfigurável.** Arquiteturas de referência e frameworks são projetados para serem reusados em múltiplos sistemas e, se necessário, serem reconfigurados. Reusar estruturas de arquitetura de software é apropriado visto que é uma parte essencial de qualquer sistema, além de ter um alto custo de desenvolvimento. Uma arquitetura de LPS é projetada para dar suporte à variação necessária para os produtos da linha de produtos, assim torná-la reconfigurável é desejável. No entanto, a arquitetura da LPS é apenas um artefato, embora importante, no núcleo da linha de produtos.
- **Releases e versões de produtos individuais.** Uma LPS gera vários produtos ao mesmo tempo, todos esses produtos passam por seus próprios ciclos de lançamento e controle de versões simultaneamente. Assim, a evolução de um único produto deve ser considerada dentro de um contexto mais amplo, ou seja, a evolução da LPS como um todo. No contexto de produtos individuais, uma vez que é lançada uma nova versão, as versões anteriores perdem seu valor e são descartadas. Mas em uma LPS uma versão anterior de um produto, que ainda é considerado com potencial de mercado, é mantida como um membro viável da família de produtos, dado

PUC-Rio - Certificação Digital Nº 0812625/CA

que consiste em uma instância de artefatos do núcleo da linha de produtos, assim como outras versões de outros produtos.

- **Apenas um conjunto de normas técnicas.** Muitas organizações adotam um conjunto de normas técnicas para limitar as escolhas de seus desenvolvedores de software no que diz respeito aos tipos e fontes dos componentes a serem usados nos sistemas que desenvolvem. Essas normas consistem em restrições para promover a interoperabilidade e reduzir os custos associados à manutenção e suporte de componentes comerciais. No entanto, essas normas são simplesmente restrições impostas às LPS.

Na engenharia de linhas de produtos de software são identificadas duas atividades principais:

- **Engenharia de domínio**, que consiste na definição dos aspectos comuns e variáveis das LPS, na realização desses aspectos em artefatos de software e a definição dos links de rastreabilidade entre esses artefatos. Os links de rastreabilidade facilitam o reuso sistemático e consistente dos artefatos de software.
- **Engenharia de aplicação** que tem por objetivo realizar a derivação dos produtos de software através do reuso dos artefatos produzidos na atividade de engenharia de domínio, explorando a variabilidade definida para a LPS.

Essa pesquisa engloba essas duas atividades. A atividade de engenharia de domínio é realizada na definição da abordagem para o desenvolvimento das LPS de serviços descrita na Seção 3.1 desta tese. A atividade de engenharia de aplicação é realizada através da abordagem de derivação de produtos a partir dos scripts de colaboração que é apresentada na Seção 3.4 desta tese.

A principal atividade da engenharia de domínio consiste na identificação dos aspectos comuns e variáveis entre os membros de uma família de software. Essa variabilidade é descrita em termos de features que são definidas como “uma propriedade do sistema que é relevante para algumas das partes interessadas e é usada para capturar ou discriminar semelhanças entre os produtos em SPL” [29].

Após identificadas, as features são então organizadas em um modelo de features (*feature modelling*). Essa atividade, além de identificar as features comuns e variáveis da LPS, captura também suas interdependências. Foi originalmente proposta pelo método Feature Oriented Domain Analysis (FODA) [30] e tem sido usada em várias abordagens de SPL.

O suporte à variabilidade é alcançado, tendo em conta a modularização das features e adotando técnicas para promovê-la. O principal benefício de se

PUC-Rio - Certificação Digital Nº 0812625/CA

ter features modularizadas é possibilitar que uma feature seja incluída ou removida da LPS viabilizando a derivação sistemática de produtos e, consequentemente, reduzindo o tempo e custos do processo de derivação na atividade de engenharia de domínio. Diferentes técnicas de desenvolvimento de software podem ser usadas para modularizar features [31] como, por exemplo, polimorfismo, design patterns, frameworks, componentização, compilação condicional e programação orientada a aspectos.

De acordo com Galster [32], a variabilidade acontece em diferentes fases no ciclo de vida do software como, por exemplo, na fase de projeto do software, chamada variabilidade de projeto (design time variability), e em tempo de execução, chamada variabilidade em tempo de execução (runtime variability). As fontes para a variabilidade tanto de projeto quanto de execução são tanto mudanças nos requisitos ou propagação de modificações em diferentes níveis (nível de negócio ou de processo, por exemplo) quanto à necessidade de responder a situações de erros.

As LPS tradicionais tratam da variabilidade em tempo de projeto, onde as possíveis variações são previstas, modeladas e implementadas durante o projeto da LPS. Para tratar da variabilidade em tempo de execução, a literatura aponta o conceito de linhas de produtos de software dinâmicas (DSPL – Dynamic Software Product Lines). DSPL é definida por Burégio et. al. [33] como uma "estratégia promissora para lidar com projeto e implementação de mudanças que devem ser realizadas em tempo de execução em aplicações de domínios emergentes", uma vez que produz software capaz de se adaptar a mudanças nas necessidades dos usuários em tempo de execução.

A adaptação de software em tempo de execução é uma característica desejável no contexto de groupware, dado que o trabalho colaborativo é dinâmico, o que exige maior flexibilidade dos serviços para se adaptar a mudanças nos requisitos de colaboração. Os requisitos de colaboração nessa pesquisa são formalizados em scripts de colaboração. O uso desses scripts tem sido explorado em pesquisas na área da aprendizagem colaborativa com suporte computacional (CSCL – Computer-supported collaborative learning).

2.2.Aprendizagem Colaborativa com Suporte Computacional

A aprendizagem colaborativa refere-se a atividades de aprendizagem explicitamente projetadas e executadas por pares ou pequenos grupos de estudantes para atingir objetivos de aprendizagem comuns [10].

PUC-Rio - Certificação Digital Nº 0812625/CA

A motivação para a investigação da aprendizagem colaborativa se dá pelo fato das pessoas aprenderem muito umas com as outras. Essa aprendizagem acontece de diferentes maneiras: resolvendo problemas em conjunto, obtendo informações sobre problemas já resolvidos, explicando soluções para problemas, debatendo sobre vantagens e desvantagens de uma determinada escolha, fazendo ou recebendo críticas, dentre outras atividades em grupo [34].

A estruturação da aprendizagem colaborativa ocorre a partir dos seguintes princípios [34]: (1) os estudantes trabalham juntos com o objetivo comum do aprendizado e; (2) os estudantes são responsáveis tanto pela sua própria aprendizagem quanto pela aprendizagem dos demais. Isso implica no estabelecimento de metas coletivas que, quanto melhor atendidas, maiores são as possibilidades de aprendizagem de cada estudante sobre o assunto estudado.

A aprendizagem colaborativa é praticada por educadores nos diversos níveis escolares, do ensino fundamental à pós-graduação [34] e em diversas situações de educação, desde a formal, nas escolas à informal, como no caso de museus [11]. O suporte computacional para essas atividades é resultado de avanços na área de Aprendizagem Colaborativa com Suporte Computacional (CSCL).

Essa pesquisa contribui para a área de CSCL dado que objetiva prover um ambiente de suporte computacional adequado a técnicas de colaboração descritas em scripts de colaboração. Os scripts são usados na transposição das técnicas de aprendizagem colaborativas para contextos mediados por computador [35]. A próxima seção detalha o conceito de scripts de colaboração.

PUC-Rio - Certificação Digital Nº 0812625/CA

2.2.1.Scripts de Colaboração

Um script de colaboração consiste em um cenário pedagógico a ser seguido pelos estudantes envolvidos em um ambiente de aprendizagem colaborativa [1]. Scripts de colaboração estruturam o processo interativo entre dois ou mais estudantes propiciando a colaboração através da especificação de uma sequência de atividades de aprendizagem em conjunto com funções (papéis) apropriadas para os estudantes, a fim de provocar o envolvimento em atividades sociais e cognitivas que de outra forma não ocorreria [35, 36].

Pesquisas em scripts de colaboração [37, 1, 35] diferenciam dois tipos de scripts:

- **Micro-scripts** são modelos de diálogos, geralmente modelos de argumentação, que são embutidos no ambiente computacional e que os estudantes devem adotar e progressivamente internalizar, por exemplo: inícios de sentenças, perguntas ou descrições detalhadas de atividades.
- **Macro-scripts** são modelos pedagógicos focados na organização e estruturação da atividade colaborativa. Eles estruturam a interação enfatizando a sequência de atividades que devem ser realizadas pelos estudantes.

Essa pesquisa foca nos macro-scripts para guiar o processo de derivação de groupware pois informam em alto nível como as pessoas deverão interagir no groupware.

Não há, até o momento dessa pesquisa, uma padronização na descrição de cenários pedagógicos para a o uso combinado com tecnologias computacionais. A Open University of the Netherlands (OUNL) desenvolveu uma linguagem, a IMS Learning Desin (IMS-LD), que foi projetada para possibilitar a descrição de diferentes práticas pedagógicas [38]. No entanto, a linguagem carece de software de suporte, o que dificulta seu uso por pessoas com conhecimentos limitados em computação.

A falta de padronização dos scripts de colaboração levou ao desenvolvimento do framework genérico [35] usado nessa pesquisa para determinar a linguagem de especificação desses scripts. Esse framework usa um pequeno número de componentes e mecanismos que são descritos a seguir:

- **Componentes.** Compreendem os indivíduos que participam em um script de colaboração, as atividades que eles realizam, os papéis que assumem, o recurso que usam e os grupos que forma.
    - **Participantes**. Em geral indica o número de participantes que um script deve ter, por exemplo: três a oito participantes. Pode-se também levar em consideração a opinião ou conhecimento em um domínio, por exemplo participantes com opiniões divergentes.
    - **Atividades**. Indica as atividades que os participantes devem executar no script.
    - **Papéis**. Refere-se a participantes específicos para os quais atividades devem ser atribuídas e recursos devem ser alocados. Também se pode associar participantes com privilégios, obrigações e expectativas.
    - **Recursos**. Compreende objetos virtuais ou físicos que podem oferecer uma informação ou funcionalidade importante. Em geral, é

PUC-Rio - Certificação Digital Nº 0812625/CA

associado a participantes, mas também podem ser associados a atividades.

- **Grupos**. Participantes podem ser agrupados para executar as atividades definidas no script. Os critérios para a formação de grupos são diversos como, por exemplo, gênero, faixa etária ou apenas grupos aleatórios para alcançar a quantidade de participantes necessária para as atividades.

- **Mecanismos**. Descrevem a natureza distribuída dos scripts através da definição de atributos como:
  - **Distribuição de tarefas**. Descreve como as atividades, os papéis e recursos são distribuídos entre os participantes.
  - **Formação de grupos**. Descreve como participantes são distribuídos entre os grupos.
  - **Sequenciamento**. Descreve como componentes e grupos estão distribuídos através do tempo.

Dado que os scripts de colaboração estruturam o fluxo das atividades colaborativas, os grupos e os papéis envolvidos nessas atividades, nessa pesquisa propôs-se o uso do Business Process Modeling Notation (BPMN) para descrevê-los. A próxima seção apresenta o BPMN e seus principais elementos.

2.3.BPMN – Business Process Model Notation

Business Process Modeling Notation (BPMN) [39] é uma notação para modelar processos de negócios especificada pela Object Management Group (OMG [40]). Foi desenvolvida para prover uma notação que pudesse ser entendida por diferentes tipos de usuários, desde analistas de negócios, que criam as versões inciais do processo, a desenvolvedores responsáveis pela tecnologia de suporte a tais processos, além dos usuários responsáveis pelo gerenciamento do processo. BPMN cria uma ligação padrão entre o projeto de processos de negócio e a implementação desses processos.

BPMN define um diagrama de processo de negócios (Business Process Diagram – BPD), que é baseado na técnica de fluxograma adaptada para a criação de modelos gráficos de operações de processos de negócios. Um modelo de processos de negócio, então, consiste numa rede de objetos gráficos que são atividades e fluxos de controle que define a ordem de execução das atividades [41]. Os elementos do BPD são divididos em 4 categorias:

PUC-Rio - Certificação Digital Nº 0812625/CA

- **Objetos de fluxo.** Um BPD conta com três elementos para representar os objetos de fluxo que são:
    - **Evento (event).** Um evento é representado por um círculo e representa algo que “acontece” durante o andamento de um processo de negócio. Esses eventos afetam o fluxo dos processos e geralmente têm uma causa ou um impacto. Existem três tipos de eventos, baseados em quando eles afetam o fluxo: início, intermediário e fim, representados respectivamente na Figura 1(a).
    - **Atividade (activity).** Uma atividade é representada por um retângulo com cantos arredondados e é um termo genérico para o trabalho que a companhia executa. Uma atividade pode ser atômica ou composta. Os tipos de atividade são: tarefas e subprocessos. O subprocesso é identificado por um sinal de soma “+” centralizado na parte inferior da forma. A atividade é representada na Figura 1 (b).
    - **Gateway.** Um gateway é representado por um losango e é usado para controlar a divergência e convergência da sequência de fluxo.

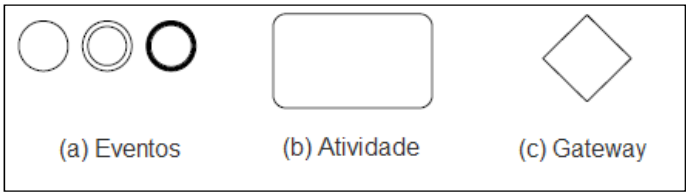

Figura 1. Objetos de fluxo BPMN [41].

- **Objetos de conexão.** Os objetos de fluxo são interligados para criar a estrutura do processo de negócio, para tanto, existem três objetos de conexão representados pela Figura 2 e descritos abaixo:
    - **Fluxo de sequência (sequence flow).** Um fluxo de sequência é representado por uma seta sólida direcionada e é usada para ordenar as atividades a serem executadas no processo.
    - **Fluxo de mensagem (sequence message).** Um fluxo de mensagem é representado pela seta entrecortada direcional e é usada para mostrar o fluxo das mensagens entre dois processos participantes.
    - **Associação (association).** Uma associação é representada por uma seta pontilhada direcionada e é usada para associar dados, textos e outros artefatos com objetos de fluxo. Associações são usadas para ilustrar as entradas e saídas das atividades.

PUC-Rio - Certificação Digital Nº 0812625/CA

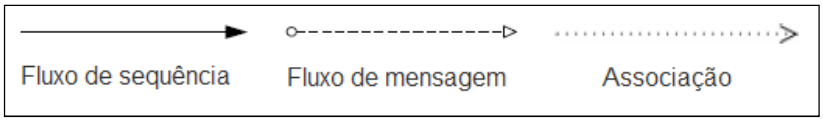

Figura 2. Objetos de conexão BPMN [41].

- **Swimlanes.** Diversos processos usam o conceito de swimlanes como um mecanismo para organizar atividades em categorias visualmente separadas para ilustrar diferentes possibilidades funcionais ou responsabilidades. As swimlanes estão ilustradas na Figura 3. São dois tipos de estruturas:
    - **Pools.** Uma pool representa um participante em um processo. Também atua como um container gráfico para dividir o conjunto de atividades de outras pools.
    - **Lanes.** Uma lane é uma partição dentro de uma pool e ocupa a largura inteira da pool. Lanes são usadas para organizar e categorizar atividades.

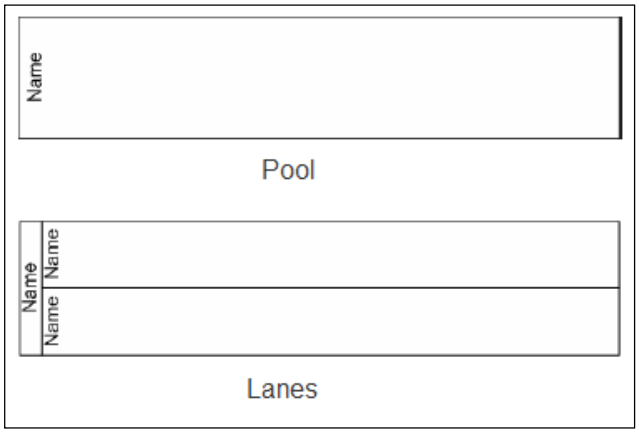

Figura 3. Swimlanes BPMN [41].

- **Artefatos.** Os artefatos podem ser adicionados ao diagrama de acordo com a necessidade do processo que está sendo modelado (Figura 4). São três os elementos para representar artefatos:
    - **Objetos de dados (data objects).** Objetos de dados são mecanismos para ilustrar como o dado é requerido ou produzido pelas atividades. Eles são conectados a atividades por associações.
    - **Grupo (group).** Um grupo é representado por um retângulo com cantos arredondados desenhado com linha pontilhada. O agrupamento pode ser usado com o propósito de documentação ou análise e não afeta o fluxo de sequência.
    - **Anotação (annotation).** Anotações são mecanismos para prover informação textual adicional ao leitor de um diagrama BPMN.

PUC-Rio - Certificação Digital Nº 0812625/CA

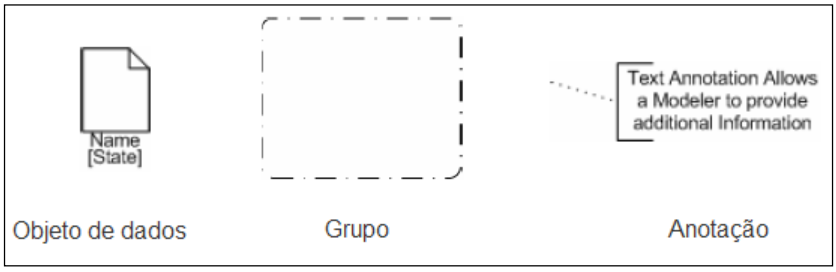

Figura 4. Artefatos BPMN [41].

Conforme dito anteriormente, nessa pesquisa, o BPMN foi usado como base para definir os scripts de colaboração. Nem todos os elementos descritos acima foram utilizados. Outros tiveram seus significados alterados para uma maior aderência ao framework de elaboração de scripts de colaboração usado. A Seção 3.2 detalha como os scripts de colaboração devem ser escritos com os elementos do BPMN. Esses scripts servem de fonte para a derivação de groupware na LPS desenvolvida que diferencia-se das demais LPS pela análise da colaboração através do Modelo 3C de Colaboração.

2.4.Groupware e Modelo 3C de Colaboração

A colaboração nesta pesquisa é analisada de acordo com o Modelo 3C ilustrado pela Figura 5, que considera que esta é alcançada através da interação entre a comunicação, coordenação e cooperação.

Durante a comunicação ocorre uma troca de mensagens objetivando futura ação comum. Coordenação trata das pessoas e suas interdependências necessárias para o cumprimento do plano de ação acordado. Cooperação compreende as ações tomadas pelos membros do grupo no espaço compartilhado.

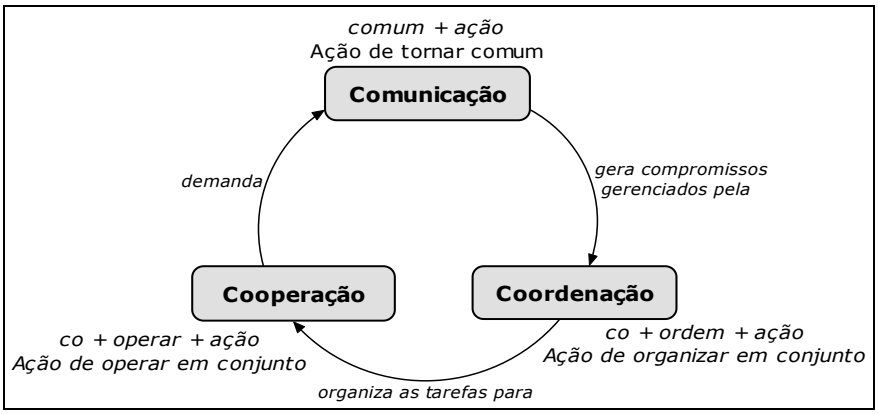

Figura 5. Modelo 3C de Colaboração [42].

PUC-Rio - Certificação Digital Nº 0812625/CA

Para dar suporte à comunicação, Fuks et. al [42] afirma que o projetista das ferramentas de comunicação define os elementos de comunicação que, por sua vez, definem o canal de comunicação entre os interlocutores como propósito, dinâmica, tempo e espaço. De acordo com as necessidades do grupo, aspectos como privacidade e sobrecarga de informação são levados em consideração.

Se o objetivo for coordenar pessoas, o foco da coordenação deve ser as ferramentas que provêm agenda e contexto. Coordenar recursos está relacionado ao espaço compartilhado, onde as ações acontecem. Coordenar tarefas consiste em gerenciar interdependências entre tarefas que devem ser executadas para atingir o objetivo do grupo. Então, o projetista deve considerar esses diferentes aspectos da coordenação ao projetar ferramentas.

Para dar suporte à cooperação, o projetista deve configurar o espaço compartilhado onde as ações irão acontecer. Um conjunto de ferramentas para armazenamento e manutenção de artefatos (documentos, planilhas, apresentações e outros) deve ser oferecido.

Apesar de parecerem independentes, os “C”s possuem um intra-relacionamento [43]. Por exemplo, em uma ferramenta cujo propósito é a comunicação, como no caso de um fórum de discussão, é possível verificar os outros “C”s do modelo. Usuários em um fórum postam mensagens que estão disponíveis a outros (cooperação) e existe uma lista de participantes (coordenação).

2.4.1.RUP 3C-Groupware

RUP-3C-Groupware [25] consiste numa extensão e especificação do Rational Unified Process que tem por objetivo documentar como o Modelo 3C de Colaboração é sistematicamente usado nas diferentes etapas de um processo de desenvolvimento de groupware. O RUP 3C-Groupware se aplica neste trabalho na etapa de análise de domínio da linha de produto, para a classificação dos produtos de groupware resultantes e seus elementos. Os procedimentos para realizar a análise de domínio são documentados pelo fluxo “Analisar Domínio” do RUP 3C-Groupware, conforme esquematizado na Figura 6.

De acordo o fluxo, cabe ao Analista de Domínio analisar diferentes aplicações do domínio para o qual o novo groupware está sendo desenvolvido. O analista estabelece comparações entre as aplicações para identificar e abstrair os elementos de comunicação, coordenação e cooperação do domínio. Como

PUC-Rio - Certificação Digital Nº 0812625/CA

resultado desta atividade, objetiva-se construir um Quadro Conceitual 3C do domínio, ou aperfeiçoar algum já existente. Ao analisar as aplicações do domínio, devem-se documentar as principais funcionalidades classificando-as de acordo com o Quadro Conceitual 3C. O analista também deve caracterizar o que é uma aplicação típica daquele domínio, identificando o conjunto mais relevante de elementos. Isto consistirá no núcleo da linha de produto de aplicações do domínio, a partir do qual novas características poderão ser adicionadas aos diferentes produtos derivados da linha de produtos.

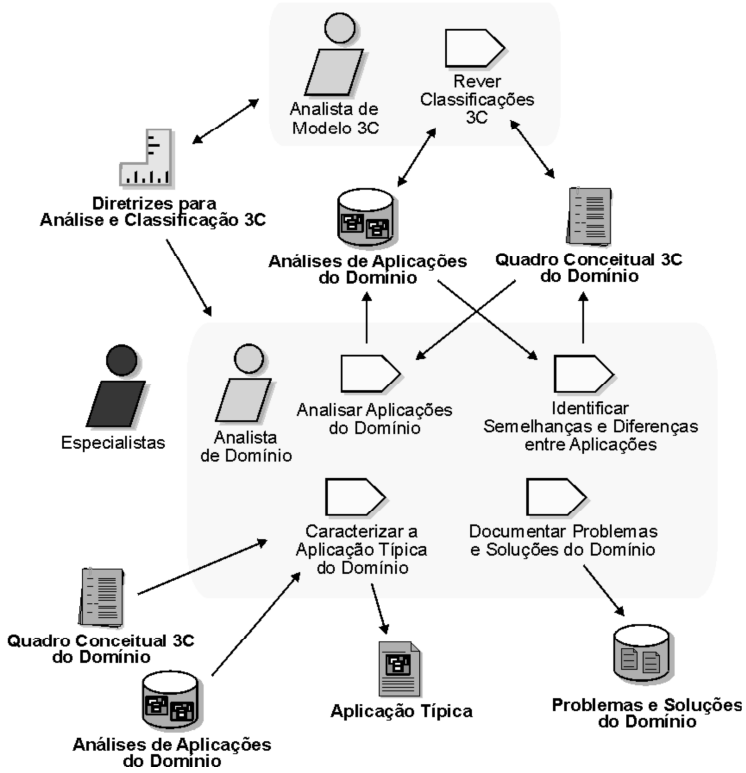

Figura 6. Fluxo Analisar Domínio do RUP 3C-Groupware [25].

Alguns problemas e soluções do domínio já podem ser conhecidos e devem estar documentados num repositório, tornando-se útil para auxiliar o analista na seleção ou especificação de uma variação de solução já conhecida em outras aplicações. Deve-se, ainda, contar com um Analista de Modelo 3C que será responsável pelo uso consistente do modelo ao longo do processo de desenvolvimento.

PUC-Rio - Certificação Digital Nº 0812625/CA

2.4.2.Groupware Workbench

Groupware Workbench (GW) é uma bancada baseada em componentes para o desenvolvimento de sistemas colaborativos. A bancada oferece aos engenheiros de software uma infraestrutura componentizada específica para o domínio de groupware baseados no Modelo 3C de Colaboração para instrumentar a construção e manutenção de sistemas colaborativos extensíveis e adaptáveis. Assim, os engenheiros de software lidam com o projeto da colaboração em um alto nível [44].

A maioria dos groupware provê um conjunto comum de serviços colaborativos a seus usuários como fóruns de discussão, agenda, repositório de arquivos, questionários, gerenciamento de links e relatório de atividades. Essas características são apropriadas para o uso das técnicas de desenvolvimento baseado em componentes e linhas de produto, dado que serviços colaborativos são componentes de groupware e podem ser adaptados para atender alguma necessidade específica de colaboração. Esses componentes de groupware no GW são denominados Collablets.

A mesma análise aplicada a sistemas e seus serviços é aplicada para serviços e suas funcionalidades, que são recorrentes. Por exemplo, diversos serviços dentro de um ambiente reusam a categorização de mensagens e avaliação, controle de permissões, dentre outros. Encapsular essas funcionalidades em componentes possibilita também a outros desenvolvedores o reuso dessas funcionalidades em seus projetos, tornando possível a evolução, adaptação e construção de serviços variando e reconfigurando os componentes de colaboração. Esses componentes de colaboração no GW são chamados CollabElements. No GW, tanto Collablets quanto CollabElements são organizados de acordo com o Modelo 3C de Colaboração.

Assim, o desenvolvimento de groupware usando o GW consiste na composição de Collablets que implementam serviços de groupware. Esses Collablets são resultados da união de CollabElements seguido por suas adaptações para prover funcionalidade específica. Isso favorece um reuso de componentes em dois níveis: na composição do sistema colaborativo (groupware) e na composição de cada serviço colaborativo (Collablet).

PUC-Rio - Certificação Digital Nº 0812625/CA

2.5.Desenvolvimento Dirigido por Modelos

Desenvolvimento dirigido por modelos (Model Driven Development - MDD) é uma abordagem descrita pela Engenharia Orientada a Modelos (Model Driven Engineering – MDE) que trata da redução da distância entre o domínio do problema e o domínio de implementação de software. Isso é atingido através do uso de tecnologias de suporte a transformações sistemáticas de modelos que representam abstrações no nível do problema para implementações de software [45].

A principal característica do MDD é que o foco do desenvolvimento de software está nos modelos e não na sua implementação. A maior vantagem dessa abordagem está no fato dos modelos usarem conceitos mais relacionados ao domínio do problema do que às tecnologias de implementação utilizadas. Em alguns casos, como o desta tese, pode ser até mais fácil para especialistas do domínio produzirem software do que para especialistas em tecnologia [46]. Isso torna os modelos independentes da tecnologia computacional escolhida e da evolução dessa tecnologia.

A seguir são apresentadas algumas motivações para a adoção do MDD [47]:

- **Aumento da produtividade**. O aumento da produtividade acontece em função da diminuição do tempo de desenvolvimento de software, sobretudo devido à automação na geração dos artefatos de software;
- **Melhoria da qualidade.** Melhorias na qualidade do código gerado, dos requisitos do sistema, no gerenciamento desses requisitos, e previsão de bugs no sistema;
- **Automação.** Geração de código e outros artefatos automaticamente durante o processo de desenvolvimento de software. Simulação e testes baseados em modelos;
- **Padronização e formalismo.** Provê um framework comum para o desenvolvimento de software que formaliza e organiza o conhecimento da engenharia de software em um nível mais alto de abstração e estabelece um formato comum de exportação de dados;
- **Manutenção e evolução.** Mantém a arquitetura intacta desde a análise até a implementação e a evolução de sistemas legados;
- **Melhoria na comunicação e compartilhamento de informação.** Entre os stakeholders e entre a equipe de desenvolvimento. Facilidade no aprendizado.

PUC-Rio - Certificação Digital Nº 0812625/CA

Características como a facilidade no aprendizado e a automação do processo de desenvolvimento de software foram determinantes para o uso dos conceitos de MDD nesta pesquisa. Com a representação do script de colaboração sob a forma de modelos desenvolvidos pelo usuário final, o groupware é derivado conforme detalhado na Seção 3.4.

2.6.Programação pelo usuário final

Os programadores usuários-finais são usurários que não foram formalmente treinados em programação, porém precisam programar para cumprir suas atividades diárias. Planilhas são frequentemente citadas como o maior sucesso da história da programação pelo usuário final (End-user programming - EUP). Milhões de usuários escrevem fórmulas em sistemas como o Microsoft Excel apesar de apenas poucos se considerarem programadores de fato. Muitos nem se dão conta que estão programando [48].

A EUP tem se tornado um conceito relevante em engenharia de software por, pelo menos, dois motivos: (i) o alto custo de construir e manter aplicações multifuncionais, e a demanda constante dos usuários por melhoramentos e extensões para os software, inclusive os mais bem sucedidos e; (ii) a crescente consciência de que os usuários não são mais consumidores passivos de software e podem exercer o papel de projetista e produtores de software [49].

A literatura cita diferentes abordagens no uso de EUP, tais como as descritas a seguir [50]:

- **Programação de preferências.** Preferências são alternativas pré-definidas pelo projetista da aplicação e usadas para suprir as necessidades de diferentes tipos de usuários finais. Consiste em uma maneira simplificada para o projetista adicionar alternativas à aplicação, porém, o excesso de alternativas pode se tornar um fator complicador para os usuários.
- **Programação por demonstração.** Gravadores de macros são usados para gravar as entradas dos usuários e depois repeti-las. Essas entradas gravadas são eventos de baixo nível como um clique no mouse em uma posição da tela, o que torna difícil reproduzir exatamente o mesmo efeito. A programação por demonstração, também é conhecida por programação por exemplos, consiste basicamente em um gravador de macros que ao invés de registrar eventos de baixo nível, produz um programa geral de eventos de nível médio ou ações de usuários.

PUC-Rio - Certificação Digital Nº 0812625/CA

- **Programação de planilhas.** Nas planilhas, ao criar fórmulas e construir modelos na forma de funções e referências de células, os usuários finais estão programando. As planilhas constantemente dão feedback aos usuários finais a medida em que eles progridem na programação mesmo quando o programa não está completo ou contém erros. Isso significa que os usuários finais podem continuar suas atividades sem serem interrompidos para compilar e testar os programas como na programação textual.
- **Programação de scripts.** Programação de scripts é quando a programação ocorre através da adoção de uma linguagem de scripts. Uma linguagem de script é uma linguagem pequena, geralmente com vocabulário especificamente projetado para a aplicação e seu domínio.

Na presente pesquisa, o conceito de EUP é usado sobre a abordagem da programação de scripts alinhada com MDD de modo a instrumentar os usuários finais da aplicação desenvolvida nessa tese para o desenvolvimento de seus próprios groupware. O usuário final é responsável pela criação dos scripts de colaboração, sob a forma de modelos, que servem de ponto de partida para a derivação de seu groupware de suporte.

A seguir são apresentados trabalhos na literatura que são relacionados à pesquisa apresentada nessa tese.

2.7. Trabalhos Relacionados

Essa tese envolve três conceitos principais: desenvolvimento de groupware, linhas de produtos de software e scripts de colaboração.

Com relação ao desenvolvimento de groupware, diversos trabalhos descrevem abordagens para minimizar o esforço durante desenvolvimento. A idéia dessas abordagens é encapsular as dificuldades técnicas da implementação de groupware como, por exemplo, gerenciamento de sessão e protocolos de comunicação. Dentre as motivações para o desenvolvimento baseado em componentes, essas abordagens concentram-se na composição (tailorability) de groupware [51, 52] e linhas de produtos [9, 53, 54]. A composição de groupware está relacionada com a questão da montagem do groupware pelo usuário final, porém, ainda exige conhecimentos específicos de computação e componentização. A questão das linhas de produtos apontada pelas abordagens citadas está relacionada ao reuso sistemático desses componentes, que são desenvolvidos com o objetivo de serem reusados em

PUC-Rio - Certificação Digital Nº 0812625/CA

diversos groupware, reduzindo o investimento total do desenvolvimento e os custos de manutenção de software.

Exemplos que ilustram essas abordagens são: (1) GroupKit [53], que provê um toolkit que encapsula as complexidades intrínsecas do desenvolvimento de groupware síncrono; (2) FreEvolve [51], que foi projetado para possibilitar a customização de groupware através da composição de componentes ou a adaptação de software existente para os usuários finais, e; (3) DreamTeam [54], que é uma plataforma baseada em componentes de suporte a construção de groupware síncrono.

Apesar de essas abordagens proverem técnicas que reduzem o esforço no desenvolvimento de groupware e citarem o uso de linhas de produto de software, elas não cobrem todo o processo de desenvolvimento das LPS. Faltam as definições de atividades como a análise de domínio, que é responsável pela elicitação dos requisitos comuns e variáveis (features), e da rastreabilidade dessas features, que provê meios de derivação sistemática de groupware em uma LPS.

Com relação às LPS, são poucos os trabalhos encontrados na literatura que aplicam esse conceito ao desenvolvimento de groupware. Um desses trabalhos é o apresentado por Gaspar et. al. [55] onde é descrita uma linha de produtos de software no domínio de aplicações síncronas na Web 2.0. Essa LPS deriva aplicações para dar suporte à colaboração síncrona, como por exemplo: sala de aula virtual (áudio, vídeo e quadro eletrônico), um bate papo (áudio, vídeo e texto), uma vídeo conferência (áudio e vídeo), ou outra que seja mais adequada a uma colaboração síncrona desejada. Outro trabalho que relaciona LPS ao desenvolvimento de groupware é o de Oliveira et. al. 2011 [56], que em sua pesquisa usou o método FODA junto com padrões de interação para identificar as features de compartilhamento de conteúdo em redes sociais. O foco da sua pesquisa está na etapa de análise de domínio, onde as features foram identificadas e implementadas com o Groupware Workbench.

Os trabalhos acima não especificam como os groupware podem ser derivados a partir dos requisitos do usuário. Eles apresentam uma abordagem para a identificação de features, implementam essas features e ainda apontam como combinar essas features para atender a necessidades específicas, porém, sem detalhar essas necessidades, como são levantadas e representadas. Tais necessidades são representadas na pesquisa dessa tese por meios de scripts de colaboração. Esses scripts guiam, nessa pesquisa, o processo de derivação de groupware.

PUC-Rio - Certificação Digital Nº 0812625/CA

Alguns trabalhos apontam o uso de groupware para o suporte de scripts de colaboração. Um exemplo de groupware que usa scripts de colaboração é o ManyScripts [57], que disponibiliza três scripts para configuração por parte dos mediadores da aprendizagem para usarem com os estudantes: o “ArgueGraph”, que foca nas estratégias de formação de grupos guiado pela comunicação entre os estudantes; o “ConceptGrid” que é uma implementação do JigSaw que guia o processo da distribuição do conhecimento entre o grupo de estudantes; e o “ICE” que consiste em um sistema de revisão por pares para exercícios. Outro groupware que faz uso de scripts de colaboração é o WikiPlus [58], que usa um sistema wiki para a edição e execução dos scripts. O diferencial do WikiPlus é a possibilidade de programação de novos scripts de colaboração, porém limita a ação dos participantes aos serviços providos pelo wiki. Outro trabalho que faz uso de scripts de colaboração usa a linguagem IMS-LD para descrever os scripts em um ambiente de aprendizagem usado em um contexto misto, onde atividades virtuais são combinadas com atividades face-a-face [59]. Em seu trabalho, Sanagustin et. al. [59] destaca a dificuldade em dar suporte computacional a mudanças da colaboração em tempo de execução dos scripts principalmente quando se considera o contexto misto de aprendizagem.

A linguagem para a representação dos scripts de colaboração usada na pesquisa apresentada nessa tese é baseada em BPMN. Alguns trabalhos já relacionam o uso conjunto de linhas de produto de software e modelos de processo de negócio. Porém, o enfoque desses trabalhos está na identificação e na orquestração das chamadas dos serviços de uma arquitetura orientada a serviços (Software-oriented Architecture - SOA). Kang et. al. [60] propõe uma abordagem para a identificação de serviços da SOA que envolve modelos de features e modelos de processo de negócios alcançando assim o reuso de serviços. Outro trabalho que envolve LPS e BPMN consiste no desenvolvimento de linhas de produtos de processos de negócios [61] com o objetivo de reusar e integrar sistematicamente vários processos de negócio orientados a serviços.

O capítulo a seguir apresenta a abordagem de linha dinâmica de produto para groupware. Nessa abordagem, os groupware são derivados através da formalização da colaboração em scripts de colaboração descritos em BPMN, sintetizando os conceitos aqui apresentados.

PUC-Rio - Certificação Digital Nº 0812625/CA