

**iPlan - Uma Linha de Produtos de Software
Baseada em Padrões Pedagógicos para gerar
Planos de Aula em Ambientes Aprendizagem via Web**

Reginaldo do Prado

TESE APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
DOUTOR EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof. Dr. Leônidas de Oliveira Brandão

São Paulo, Setembro de 2015

iPlan - Uma Linha de Produtos de Software
Baseada em Padrões Pedagógicos para gerar
Planos de Aula em Ambientes Aprendizagem via Web

Esta é a versão original da tese elaborada pelo
candidato Reginaldo do Prado, tal como
submetida à Comissão Julgadora.

Agradecimentos

Agradeço, em primeiro lugar, a Deus por me dar a felicidade de realizar este trabalho e, ao longo do processo, conhecer pessoas maravilhosas que marcaram a minha vida de modo positivo e duradouro. A primeira destas pessoas é, sem dúvida, o meu orientador, o Prof. Leônidas Brandão. Sempre amigo e paciente, se mostrou um exemplo de trabalho e dedicação que poucas vezes vi; serve de grande inspiração para minha vida futura. À querida Anarosa que sempre, com muito carinho, co-orientou este trabalho informalmente; agradeço muito a ambos por terem tocado a minha vida. A todos os professores cujas aulas frequentei, desde os anos iniciais até chegar ao IME, agradeço pelo exemplo, pela capacidade e pelo fato de terem me feito, cada vez mais, acreditar no poder transformador da Educação. Aos tantos colegas que passaram momentos de tensão e de alegria nas disciplinas que cursamos juntos agradeço pela força que sempre me deram.

Certamente o convívio acadêmico foi incrível, mas nada disso seria possível sem o apoio da família. Agradeço, então, às principais apoiadoras de tudo que faço, minha esposa Sonia e minha filha Isadora. A elas, que conviveram bastante tempo comigo por perto e, ao mesmo tempo, inacessível, só posso dizer que amo muito; não há palavras suficientes para lhes agradecer. Por último, porém não menos importante, agradeço à minha mãe, Maria das Graças (*in memoriam*), por ter sido a grande incentivadora do trabalho, da vida correta, da honestidade e do amor pelo aprendizado. Onde estiver saiba que este título não seria possível sem a sua participação.

Resumo

Prado, R. **iPlan - Uma Linha de Produtos de Software Baseada em Padrões Pedagógicos para gerar Planos de Aula em Ambientes Aprendizagem via Web.** 2015. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2015.

O avanço do ensino *online* fez crescer a demanda por professores capazes de elaborar aulas adequadas aos Ambientes de Aprendizagem via Web (AAW). Entretanto, o processo de autoria de conteúdos digitais não é trivial, seja ele para *Web* ou não. Diante desta dificuldade, os professores frequentemente optam por repetir no ambiente digital as práticas de suas aulas presenciais deixando de aproveitar todo o potencial didático dos recursos digitais. Uma possível solução é oferecer aos professores componentes testados, ou seja, utilizar padrões, como ocorre com os Padrões de Projeto no desenvolvimento de software orientado a objetos. Padrões, quando aplicados ao caso didático, recebem o nome de Padrões Pedagógicos e, assim como seus similares em orientação a objetos, encapsulam o conhecimento de especialistas da área de modo que este conhecimento se torne facilmente acessível e aplicável por outros professores. Faltam, no entanto, ferramentas que possibilitem ao professor fazer uso desta abordagem em suas aulas. Neste trabalho apresentamos o *iPlan*, uma Linha de Produto de Software que gera planos de aula para AAW construídos com base em Padrões Pedagógicos, além de alguns resultados positivos obtidos com sua aplicação. Experimentos com professores mostraram que a ferramenta pode efetivamente facilitar o processo de autoria em ambientes *Web*. De um lado, os professores experientes conseguiram com o *iPlan* produzir aulas em menor tempo e, de outro lado, os professores iniciantes no ensino via *Web* apresentaram uma redução no tempo de aprendizagem para a autoria de um curso *Web* quando usando o *iPlan*. O *iPlan* foi implementado como um *plugin* do sistema Moodle devendo ser brevemente disponibilizado para a comunidade como *software livre*.

Palavras-chave: Padrões Pedagógicos, Linha de Produto de Software, Ensino via Web, Moodle, Formato de Curso

Abstract

Prado, R. **iPlan - A Software Product Line Based on Pedagogical Patterns to generate Lesson Plans in Learning Environments on the Web.** 2015.

Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2015.

Advances in online education has increasing the demands for teachers capable to manage Web courses. However, the digital content authoring is not a simple task. Considering this difficult, it is common that some teachers reproduce in the digital environment the face-to-face model, which reduce potential improvements that could be performed by the digital tools. A possible solution to this is the use of tested components, like the design patterns, used in the object oriented system development. When inserted in the educational context, they are called Pedagogical Patterns (PP). As their software counterpart, PP encapsulates the specialists knowledge, in order to make it available to other teachers. Nonetheless, it is missing tools to empower teachers to use this approach to produce their classes. In this work it is presented iPlan, a Software Product Line to produce lesson plans to Web learning environments (WLE), using PP, besides some positive results with the iPlan use by teachers. Experiments with teachers show that the iPlan can effectively simplify the Web authoring process. On one hand, teachers experienced with tecnologies presented a reduction in their production time when using iPlan, and inexperienced teachers presented a short time to learn the how to prepare a Web course when using iPlan. The iPlan has been implemented to Moodle system, should soon be available to the community as a free software.

Keywords: Pedagogical Pattern, Software Product Line, Teaching by Web, Moodle, Course Format

Sumário

| | |
|--|-------------|
| Lista de Abreviaturas | xi |
| Lista de Símbolos | xiii |
| Lista de Figuras | xv |
| Lista de Tabelas | xvii |
| 1 Introdução | 1 |
| 1.1 Motivação | 1 |
| 1.1.1 A especificação IMS-LD | 1 |
| 1.1.2 Uma solução aproximada | 3 |
| 1.2 Objetivo | 3 |
| 1.3 Questões de Pesquisa | 3 |
| 1.4 Metodologia de Pesquisa | 3 |
| 1.5 Organização do Texto | 4 |
| 2 Sistemas de Apoio à Preparação de Aulas para o Ensino Online | 5 |
| 2.1 Introdução | 5 |
| 2.1.1 A importância do <i>e-learning</i> | 5 |
| 2.1.2 Os desafios ao professor | 5 |
| 2.1.3 A necessidade de uma nova postura docente | 6 |
| 2.1.4 Sobre a receptividade dos estudantes | 6 |
| 2.2 Planejamento de Aulas para Ensino <i>Online</i> | 6 |
| 2.2.1 Foco em objetivos | 7 |
| 2.2.2 Atividades sob um novo ponto de vista | 7 |
| 2.2.3 A importância da preparação do instrutor | 7 |
| 2.2.4 Avaliação e <i>feedback</i> | 7 |
| 2.2.5 Requisitos para o bom planejamento | 8 |
| 2.2.6 Boas práticas no planejamento de <i>e-learning</i> | 8 |
| 2.3 Revisão Sistemática sobre Sistemas de Apoio à Preparação de Aulas | 9 |
| 2.3.1 Um Resumo do Protocolo de Revisão | 9 |
| 2.3.2 Descobertas Decorrentes da Revisão Sistemática | 10 |
| 2.4 Ferramentas e Metodologias para <i>e-learning</i> Encontradas na Revisão | 10 |
| 2.4.1 Navegador da Internet | 10 |
| 2.4.2 A abordagem de Templates | 11 |

| | | |
|----------|--|-----------|
| 2.4.3 | Educational Modelling Language (EML) | 11 |
| 2.4.4 | IMS Learning Design | 13 |
| 2.4.5 | Conclusão | 19 |
| 3 | Padrões Pedagógicos | 21 |
| 3.1 | Introdução | 21 |
| 3.2 | O conceito de Padrão | 21 |
| 3.2.1 | A finalidade do Uso de Padrões | 22 |
| 3.2.2 | A Gênese dos Padrões | 22 |
| 3.3 | Padrões Aplicados ao Ensino | 23 |
| 3.3.1 | Padrões como Ferramenta Docente | 23 |
| 3.3.2 | Catálogos de Padrões Pedagógicos | 24 |
| 3.4 | Relatos sobre o uso de Padrões Pedagógicos | 24 |
| 3.4.1 | To be continued... | 24 |
| 4 | Linhas de Produto de Software | 25 |
| 4.1 | Introdução | 25 |
| 4.2 | Produtos de Software : Conceitos Básicos | 25 |
| 4.3 | Benefícios das LPSs | 27 |
| 4.4 | Abordagem de <i>features</i> | 28 |
| 4.5 | Projetando uma LPS | 28 |
| 4.6 | Processo de Desenvolvimento da LPS | 28 |
| 4.6.1 | Etapas do Desenvolvimento de LPS | 28 |
| 4.6.2 | Tarefas do Desenvolvimento de LPS | 29 |
| 4.6.3 | Escalando uma LPS | 30 |
| 4.7 | Implementando uma LPS | 31 |
| 5 | Instanciando o iPlan | 33 |
| 6 | Experimentos com o uso do iPlan | 35 |
| 7 | Conclusões | 37 |
| 7.1 | Considerações Finais | 37 |
| 7.2 | Sugestões para Pesquisas Futuras | 37 |
| A | Protocolo da Revisão Sistemática | 39 |
| B | Questionários Utilizados nos Experimentos | 41 |
| C | Avaliação de software para Learning Design | 43 |
| C.1 | Introdução | 43 |
| C.2 | Modelo usado nas avaliações | 43 |
| C.2.1 | Questões sobre a finalidade do software | 43 |
| C.2.2 | Questões sobre projeto e funcionalidades do software | 43 |
| C.2.3 | Questões sobre aspectos técnicos do software | 43 |

Referências Bibliográficas

Lista de Abreviaturas

| | |
|------|---|
| ACM | Association for Computing Machinery |
| CBIE | Congresso Brasileiro de Informática na Educação |
| FIE | Frontiers In Education |
| IEEE | Institute of Electrical and Electronics Engineers |
| IMS | Instructional Management System |
| LInE | Laboratório de Informática na Educação |
| PPP | Pedagogical Patterns Project |
| SBIE | Simpósio Brasileiro de Informática na Educação |
| SGC | Sistema Gerenciador de Cursos |
| SPA | Sistema de apoio à Preparação de Aulas |
| UA | Unidade de Aprendizagem |
| UML | Unified Modeling Language |
| WIE | Workshop de Informática na Escola |
| XML | eXtensible Markup Language |

Lista de Símbolos

- ω Frequência angular
- ψ Função de análise *wavelet*
- Ψ Transformada de Fourier de ψ

Lista de Figuras

| | | |
|-----|--|----|
| 1.1 | Diagrama de classes do modelo conceitual de learning design proposto pela IMS-LD | 2 |
| 2.1 | Diagrama do Modelo Teatral da IMS-LD. | 15 |
| 2.2 | IMS-LIP (Learner Information Package) | 17 |
| 4.1 | Sistema de Configuração de Autmóveis da BMW | 26 |
| 4.2 | Sistema de Configuração de Lanches de um Restaurante Hipotético | 27 |
| 4.3 | Processo de Desenvolvimento de Linhas de Produto de Software | 29 |
| 4.4 | Exemplo: seleção de <i>features</i> do Kernel do Linux | 30 |
| 4.5 | Geração de Produtos em uma Linha de Produtos de Software | 31 |

Lista de Tabelas

| | | |
|-----|--|----|
| 2.1 | Atividades-chave em Learning Design (Britain, 2004) | 17 |
| C.1 | Questões sobre a finalidade do software | 44 |
| C.2 | Questões sobre projeto e funcionalidades do software | 45 |
| C.3 | Questões sobre aspectos técnicos do software | 45 |

Capítulo 1

Introdução

Neste capítulo são apresentadas a motivação da pesquisa e a definição do problema a ser considerado. Levantamos ainda algumas hipóteses, indicamos possíveis caminhos de solução e apresentamos a abordagem metodológica utilizada ao longo deste trabalho.

1.1 Motivação

Os recursos da *World Wide Web* (*Web*) têm sido cada vez mais empregados para propiciar educação, formal ou informal. A universalização do acesso aos computadores e o aumento na velocidade de transmissão de dados têm incentivado as instituições de ensino de todos os níveis a investir na criação de cursos à distância. Com este aparato tecnológico à disposição, aumentam também os cursos que utilizam a *Web* para a divulgação de material e de atividades de apoio à aprendizagem, os chamados cursos semipresenciais (ABED, 2012). Como consequência, cresce a demanda por professores que sejam capacitados para preparar aulas e/ou cursos a serem ministrados pela Internet.

A maioria desses professores, segundo Durand e Downes (2009), enfrenta duas dificuldades: a primeira está ligada à autoria de conteúdos em ferramentas que pouco dominam por não serem, em geral, especialistas em Informática e a segunda está ligada às peculiaridades do planejamento da aula a ser ministrada em ambiente digital. Para Stark *et al.* (2002), as aulas a serem ministradas em ambientes digitais criadas por estes professores acabam sendo uma reprodução de suas aulas presenciais e, por esse motivo, os resultados não são bons, pois sabemos que o ensino não é aprimorado pela tecnologia quando é meramente copiado do meio físico para o digital. O desafio de usar estratégia e material pré-existentes é que estes nem sempre se traduzem bem do ensino presencial para o ambiente virtual. Seria interessante, portanto, que existissem ferramentas de uso simples e que, além disso, permitissem sistematizar o processo de criação de aulas para o ambiente virtual.

Para que tal sistematização seja possível, se faz necessária a codificação dos elementos pertinentes ao processo de ensino *online* de algum modo interpretável por máquinas. Existem várias propostas de codificação do processo de ensino/aprendizagem e, dentre elas, se destacam os Padrões Pedagógicos (PPP, 2000), a Linguagem de Modelagem Educacional (EML) (Rawlings *et al.*, 2002), o IMS Learning Design, (IMS-LD, 2003) e o Projeto Instrucional (IDOrg, 2013). Cada uma destas metodologias tem vantagens e desvantagens, porém a que tem sido considerada mais adequada à interpretação por máquinas é a especificação IMS-LD.

1.1.1 A especificação IMS-LD

Publicada em 2003, pela IMS¹ Global Consortium (IMS-LD, 2003), a especificação IMS-LD foi projetada para promover a interoperabilidade e o intercâmbio de material instrucional de modo a facilitar a reutilização de estratégias de ensino e metas educacionais. Seu objetivo é fornecer uma

¹IMS vem de Instructional Management System, nome que provoca mais dúvidas que esclarecimentos. Por este motivo a própria instituição solicita ser referenciada apenas como IMS

estrutura que possa descrever qualquer Unidade de Aprendizagem (UA), seja ela uma aula, um seminário, um roteiro de atividades, um curso ou qualquer outra atividade de ensino-aprendizagem. Para atingir este objetivo, IMS-LD representa os *learning designs*², ou seja, a descrição dos processos de ensino e aprendizagem que é parte das UA's (Valentín *et al.*, 2007), de modo semântico, formal e interpretável por computadores. O problema é que o usuário deve ter o domínio desta descrição para se beneficiar com o uso da especificação IMS-LD.

A IMS, em seu Guia de Boas Práticas em *Learning Design* (IMS-BP, 2003), descreve os passos para a criação de uma UA começando com a fase de análise onde o cenário didático é capturado na forma de uma narrativa. Depois, um diagrama de atividade em *Unified Modeling Language* (UML, 1995) deve ser criado com base nesta narrativa e este diagrama, por sua vez, deve ser convertido em um arquivo escrito em *eXtensible Markup Language* (XML, 1990). Finalmente este arquivo é colocado em um pacote de conteúdos junto com os recursos necessários para a execução da aula. Esta descrição mostra claramente um processo de criação complexo (Figura 1.1), pouco acessível a professores não especialistas em Informática e que reforça a ideia da necessidade de software que dê apoio ao professor no desenvolvimento de *learning designs*. Em decorrência desta necessidade surgiram dois tipos de ferramentas, os editores e os interpretadores (*players*) de roteiros. Podemos citar, por exemplo, o *Coopercore* (2003), o *Lams* (2003), o *Reload* (2003) e o *Cadmos* (2011). O objetivo de todos é permitir ao professor, principalmente àquele que tem menos afinidade com a Informática, criar e executar seus planos de aula. Entretanto o uso destas ferramentas tem se mostrado inviável pois, mesmo com apoio de interfaces gráficas, é difícil usar todas as características da especificação IMS-LD (Durand e Downes, 2009).

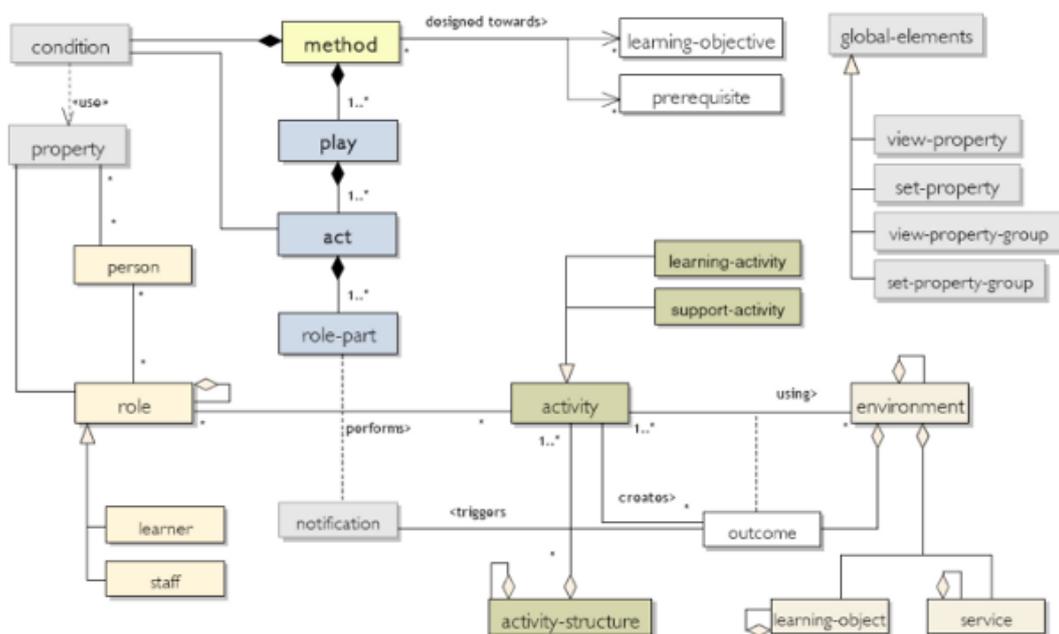


Figura 1.1: Diagrama de classes do modelo conceitual de learning design proposto pela IMS-LD

Os professores que tentam criar material instrucional usando a especificação da IMS encontram ainda um outro inconveniente: as ferramentas de autoria/interpretação estão desconectadas do ambiente onde os cursos são, de fato, ministrados. Logo, para viabilizar o emprego destes editores de UA's é necessário que o sistema gerenciador de curso (SGC) seja compatível, ou seja, que o editor possa exportar os dados da UA para o formato esperado pelo SGC ou que o SGC consiga interpretar o formato gerado pelo editor. Em se verificando tal compatibilidade, os professores podem intercambiar seus conteúdos entre editores de IMS-LD e SGC. As ferramentas citadas acima, com exceção do Cadmos, enquadram-se neste contexto, ou seja, elas geram arquivos de acordo com

²Learning Design com iniciais maiúsculas se refere à especificação da IMS e com iniciais minúsculas se refere à descrição das atividades previstas em uma Unidade de Aprendizagem

as especificações da IMS, mas deixam para o usuário o trabalho de exportar o pacote de conteúdos em formato IMS-LD para o SGC de sua instituição. Devido à sua importância a IMS-LD será descrita, detalhadamente, na Seção 2.4.4.

1.1.2 Uma solução aproximada

Conforme mencionado anteriormente, existe uma notável exceção a este modo de funcionamento que exige a exportação manual do pacote de conteúdos, que é o sistema CADMOS (*Courseware Development Methodology for Open instructional Systems*). Esta ferramenta foi desenvolvida em 2011 e, assim como os mais conhecidos, Coopercore, LAMS e Reload, não está diretamente associada a nenhum SGC, ou seja, seu funcionamento não é integrado a qualquer gerenciador de curso específico. Apresenta, no entanto, um passo adicional no sentido de facilitar seu uso por parte dos professores: ele pode exportar o roteiro para um SGC em particular, o Moodle³. Este roteiro é, basicamente, uma lista textual indicando a ordem de execução das atividades previstas em uma UA. Os dois problemas apontados acima, todavia, se mantêm: *i*) continua sendo necessário que o professor conheça as especificações do IMS Learning Design para produzir a UA e *ii*) não está disponível para o professor a opção de exportar, automaticamente, para o Moodle a unidade de aprendizagem produzida no Cadmos.

1.2 Objetivo

Neste cenário, esta pesquisa busca desenvolver um arcabouço teórico para preparação de aulas e/ou cursos a serem ministrados em ambiente *online* de modo sistemático e acessível para o professor não especialista em Informática bem como produzir uma ferramenta que implemente este modelo e que tenha o seu funcionamento integrado a um SGC. É nossa expectativa que decorra de tal desenvolvimento, por um lado, a liberação do professor da necessidade de conhecer extensas especificações técnicas e, por outro lado, a possibilidade de criação dos componentes da aula ou curso dentro do próprio SGC. A ferramenta resultante desta implementação, denominada iPlan, produzirá UA's que poderão ser armazenadas em repositórios e reutilizadas dentro do próprio sistema ou compartilhadas e executadas em outros ambientes compatíveis.

1.3 Questões de Pesquisa

- QP1A - Quais são as opções de modelagem de planos de aula existentes para cursos *online*?
- QP1B - Quais destas opções são interpretáveis por computador?
- QP1C - Quais destas opções podem ser usadas pelos professores sem que eles tenham a necessidade de conhecer especificações complexas?
- QP2A - Quais são os detalhes de funcionamento dos programas de uso mais simples?
- QP2B - Como pode ser feita a integração deste tipo de programa a SGCs?

1.4 Metodologia de Pesquisa

A metodologia adotada nesta pesquisa (Hernandez-Leo, 2007) se divide em quatro fases: informacional, proposicional, analítica e de avaliação.

Na **fase informacional**, o objetivo é coletar informações com a finalidade de, por um lado, identificar e formular as questões de pesquisa e, por outro lado, obter uma visão geral sobre o estado

³ Moodle: *Modular Object-Oriented Dynamic Learning Environment*; Sistema Gerenciador de Cursos

da arte na área de Modelagem Educacional. São previstas as seguintes atividades:

- Busca, revisão e análise da literatura especializada nos tópicos do domínio do problema: uso de padrões pedagógicos, linguagem de modelagem educacional e software para edição e execução de roteiros educacionais.

- Discussões e trocas de experiências com o time de pesquisadores do LInE (Laboratório de Informática na Educação). São particularmente importantes a experiência de implementação do módulo iTarefa (Rodrigues e Brandão, 2010) e o projeto de implementação do repositório de conteúdos instrucionais iRepositório (Nascimento *et al.*, 2013). Os módulos iRepositório e iTarefa são elementos fundamentais, pois formam a infraestrutura necessária para o funcionamento do sistema proposto neste trabalho.

- Participação em eventos acadêmicos cujo foco se relacionem, ainda que indiretamente, com os principais tópicos desta pesquisa, como, por exemplo: Simpósio Brasileiro de Informática na Educação (SBIE)⁴, Workshop de Informática na Escola (WIE) e *Frontiers in Education* (FIE)⁵.

Na **fase proposicional** serão apresentadas as possíveis soluções às questões de pesquisa identificadas usando as informações obtidas na fase anterior. Espera-se que nesta fase o modelo teórico já tenha um nível aceitável de maturidade. A partir deste modelo espera-se obter uma representação computacional que venha a permitir a implementação do sistema gerador de planos de aula.

Na **fase analítica** buscar-se-á explorar as soluções propostas com vistas à efetiva implementação e uma prova de conceito será apresentada. Espera-se que, nesta fase, várias iterações sejam realizadas até que se obtenha um modelo estável de mapeamento dos planos de aula para o ambiente computacional. Outro resultado que se almeja obter nesta fase é uma noção clara sobre as limitações deste modelo de mapeamento e, eventualmente, algumas soluções que permitam contorná-las.

Finalmente, na **fase da avaliação**, espera-se obter a validação dos modelos propostos por meio de alguns experimentos que nos permitirão avaliar as contribuições deste trabalho. Serão necessários estudos para verificar *i*) se o modelo de mapeamento produziu o resultado esperado, ou seja, simplicidade de uso para o professor, *ii*) se o público-alvo do sistema, os professores, reconhece sua importância como ferramenta de apoio na melhoria da qualidade do seu trabalho e *iii*) se o uso desta ferramenta reduz, de forma significativa, o tempo de preparação de aulas. Para que isso seja possível, apresentaremos a ferramenta para professores e/ou alunos de licenciatura, observaremos o seu uso e analisaremos as características destas interações.

1.5 Organização do Texto

Após a leitura desta introdução, o leitor encontrará no Capítulo 2 uma revisão sistemática da literatura sobre Sistemas de Apoio à Preparação de Aulas (SPA); no Capítulo 3 apresentamos a definição e uma discussão sobre Padrões Pedagógicos no contexto deste trabalho; no Capítulo 4 o conceito de Fábrica de Software é analisado e utilizado como embasamento para a produção do modelo teórico da tese; no Capítulo 5 detalhamos uma instanciação deste modelo teórico, o iPlan; no Capítulo 6 descrevemos os experimentos realizados com esta ferramenta e, por fim, no Capítulo 7 mostramos nossas conclusões, uma síntese das contribuições e um mapeamento de trabalhos futuros.

O protocolo da revisão sistemática apresentada no trabalho, os questionários utilizados nos experimentos e os dados de avaliações de softwares similares estão disponíveis, respectivamente, nos Apêndices, A, B e C.

⁴CBIE (SBIE + WIE) <http://www.cbie.org.br>

⁵<http://fie-conference.org>

Capítulo 2

Sistemas de Apoio à Preparação de Aulas para o Ensino Online

Neste capítulo apresentamos algumas noções sobre *e-learning*, sua importância na academia e na indústria, o papel do professor e os riscos e oportunidades que se apresentam para os estudantes. Tratamos do planejamento de aulas para o ambiente de ensino *online*, apresentamos a estratégia de busca e os resultados de uma revisão sistemática realizada pelo autor desta tese cuja finalidade principal foi determinar o estado da arte sobre Sistemas de Apoio à Preparação de Aula (SPA) e, no final do capítulo, apresentamos nossas conclusões.

2.1 Introdução

Em tempos recentes, a quantidade de alunos inscritos em cursos online nos Estados Unidos tem crescido a uma taxa de 10% ao ano, um incremento consideravelmente maior que a taxa de crescimento do número de matrículas no ensino superior daquele país, 1% ao ano, segundo de Vord e Pogue (2011, p. 1). No Brasil, esta tendência de crescimento também se verifica. Segundo dados da ABED (2013), em 2013¹ havia 4 milhões de brasileiros matriculados, distribuídos em 15.733 cursos a distância, o que representa um crescimento de 18,7% com relação ao ano de 2012.

2.1.1 A importância do *e-learning*

Parece clara, atualmente, a noção de que obter conhecimentos e habilidades pelo acesso à educação, treinamento e/ou desenvolvimento profissional usando as Novas Tecnologias, o que chamamos de *E-Learning*, (Stark *et al.*, 2002, p. 1), frequentemente produz ganhos em eficiência, melhora no reuso de materiais pedagógicos e, não raramente, redução nos custos do ensino e da aprendizagem (Baehr, 2012, p. 2). Várias universidades têm obtido sucesso no desenvolvimento e distribuição de cursos *online* com redução considerável de custos (Herman e Banister, 2007, p. 1); o setor produtivo também tem reconhecido a importância do ensino *online* como plataforma de treinamento para sua força de trabalho construindo redes de colaboração e aprendizagem (Baehr, 2012, p. 1).

2.1.2 Os desafios ao professor

Ocorre, no entanto, que desenvolver cursos *online* de qualidade está se tornando uma tarefa crescentemente complexa (Derntl e Calvo, 2011, p. 2) e requer uma profunda compreensão sobre tecnologia, *mídia* e modos de interação do usuário. A criação de ações de *e-Learning* apresenta vários desafios ao instrutor devido à crescente quantidade de produtos tecnológicos e estes desafios são ainda maiores para professores que têm pouca experiência didática e pouco ou nenhum apoio para o desenvolvimento de seus cursos. Os estudantes, atualmente, esperam muito mais que simples acesso pela Internet ao material instrucional, fomentando a demanda por mudanças no modo como

¹O censo de 2013 é o último publicado pela ABED em 08/2015

são implementados os cursos a distância (Stark *et al.*, 2002, p. 1). O ensino com o uso de Novas Tecnologias geralmente inclui uma combinação de várias tecnologias, *mídias* e modos de comunicação (Baehr, 2012, p. 1). Logo, a habilidade de avaliar e usar uma grande quantidade de ferramentas tecnológicas se tornou parte fundamental do trabalho do professor na criação de um curso *online* bem sucedido (Baehr, 2012).

2.1.3 A necessidade de uma nova postura docente

O modelo tradicional de instrução tem o professor como o centro do processo de ensino e aprendizagem sendo ele, geralmente, o responsável por criar o currículo e apresentá-lo aos seus alunos em aulas que o mesmo organiza; este modelo não necessariamente funciona para a instrução *online* (Stark *et al.*, 2002, p. 1). Worley e Tesdell (2009, p. 2) destacam que o corpo docente deve rever seus conceitos sobre o que constitui uma sala de aula e o que caracteriza o papel do professor enquanto busca empregar as tecnologias emergentes no ensino. Existe uma certa relutância do professor ao engajamento em projetos de *e-learning* em detrimento da crescente demanda (Stark *et al.*, 2002, p. 6). Estudos apontados por de Vord e Pogue (2011, p. 2) "... consistentemente sugerem que o corpo docente percebe o ensino *online* como muito mais trabalhoso que o ensino presencial" e que o mesmo não está interessado em dedicar o tempo necessário para criar *e-learning* (Stark *et al.*, 2002, p. 6). Para Baehr (2012, p. 9) esta visão se confirma; o autor afirma que o tempo de preparação de cursos *online* pode ser até 20% maior que o tempo de preparação para cursos presenciais, confirmando a impressão dos professores avessos à ideia de ministrar cursos via *Web*. Aparentemente, no entanto, esta impressão pode estar errada. No trabalho de de Vord e Pogue (2011) são apontados pelo menos dois estudos que sugerem que o tempo necessário para ministrar um curso *online* é menor que o tempo empenhado em um curso presencial. Neste mesmo trabalho há, contudo, outras indicações do contrário, são apontados vários autores que chegaram a resultados conflitantes sobre o tema. As autoras concluíram que o fator principal para determinar o esforço de preparação de um curso *online* é a própria preparação do professor para lidar com o aparato tecnológico, o que fica claro pela afirmação: "a quantidade de esforço necessário para ministrar um curso a distância pode ser inversamente proporcional ao esforço investido no aprendizado do desenvolvimento de *design* instrucional".

2.1.4 Sobre a receptividade dos estudantes

O corpo discente, por sua vez, tem se mostrado muito mais receptivo às novas tecnologias no ensino. Surpreendentemente, o grau de participação dos alunos em cursos *online* é maior do que aquele verificado em aulas presenciais. Altamente independentes, os alunos de cursos *online*, geralmente, se tornam produtores e consumidores do próprio conhecimento (Worley e Tesdell, 2009, p. 3) e se comportam bem com a autonomia sobre o próprio aprendizado. Alguns, no entanto, necessitam de ajuda para superar a transição do presencial para o ambiente virtual, sentem-se facilmente perdidos e/ou desmotivados e requerem atenção do professor com mais frequência (Gibson *et al.*, 1998). O papel do professor, neste caso, deve ser o de ajudar o estudante a se mover da dependência docente para um aprendizado autodirigido e independente, refinando a visão do aluno sobre sua responsabilidade pelo aprendizado, ajudando-o a criar uma agenda de estudos adaptada às suas necessidades, orientando-o na realização de suas atividades, avaliando sua performance, encorajando e motivando (Zhi-ying e Hong, 2010, p. 2).

2.2 Planejamento de Aulas para Ensino *Online*

Planejar as aulas para o ambiente de ensino *online* também requer do professor um novo posicionamento. Ao preparar um curso online, o professor deve levar em conta modos de interação diferentes daqueles usados no ensino presencial, como, por exemplo, troca de mensagens ou envio de conteúdo por áudio, texto e/ou vídeo além de considerar fatores globais como tempo, fuso horário e distância geográfica (Baehr, 2012, p. 1). Para que a comunicação seja efetiva é necessário, portanto,

um cuidadoso planejamento, pois podem ocorrer diversos problemas de codificação de ambos os lados (Gibson *et al.*, 1998, p. 2), desde problemas de ordem técnica como, por exemplo, ausência de software adequado para visualização de arquivos, até problemas de simples interpretação de intenções, tanto da parte do aluno quanto da parte do professor.

2.2.1 Foco em objetivos

O professor precisa ainda entender que o desenvolvimento do conteúdo para cursos *online* deve se basear em objetivos e não em tópicos como se verifica com frequência no ensino presencial. Stark *et al.* (2002, p. 2) mostraram que professores que têm grande experiência em sala de aula apresentam dificuldades para a mudança do paradigma de ensino baseado em tópicos para o ensino baseado em objetivos. De fato, os autores concluem que, para a criação de ações de *e-learning* bem sucedidas, os objetivos instrucionais devem guiar todo o processo de desenvolvimento do conteúdo a ser ministrado.

2.2.2 Atividades sob um novo ponto de vista

Além disso, é muito difícil transpor as ações comumente adotadas no ensino presencial diretamente para o ambiente de *e-learning*. As tentativas de fazer isso falharam em obter o engajamento dos estudantes porque a instrução era estática e não interativa (Stark *et al.*, 2002, p. 1). Uma nova abordagem é, portanto, necessária para que esta transposição seja eficiente. As atividades apresentadas aos alunos de cursos *online* devem ser, inequivocamente, detalhadas e não se pode contar com os elementos que frequentemente ficam subentendidos em uma apresentação normal de sala de aula. Se for sugerido a um aluno que busque informações na Internet, por exemplo, deve ser informado, de maneira clara, como fazê-lo e porquê (Cheon *et al.*, 2002, p. 2) (Stark *et al.*, 2002, p. 5). É necessário que o professor entenda, na prática, como o ensino *online* difere do presencial e que a formulação de conteúdo leve em consideração as diferentes possibilidades em termos de apresentação, formato e, até mesmo, estilos de aprendizagem para que sejam exploradas, ao máximo, as possibilidades de uso da interatividade.

2.2.3 A importância da preparação do instrutor

Fica claro, portanto, que estar preparado para o ensino *online* não requer apenas conhecimento das novas tecnologias, algo que, naturalmente, o instrutor deve ter, mas também envolve um profundo conhecimento sobre como os estudantes respondem à interação com o material do curso e como aprendem no ambiente de *e-learning*. A solução, apontada por Herman e Banister (2007, p. 9), passa pelo treinamento do corpo docente sobre replanejamento de curso, criação de material instrucional e critérios de avaliação. Corroborando esta visão, Stark *et al.* (2002, p. 2) relatam uma experiência em que professores especialistas em suas áreas foram convidados para produzir um curso *online* e que os resultados ficaram muito aquém do esperado em termos de qualidade, faltava a este grupo mudar o foco de tópicos para objetivos na criação do curso. Todos os professores, então, passaram por uma sessão de treinamento sobre desenvolvimento de material instrucional para *e-learning* e, depois disso, eles conseguiram criar o material instrucional adequado.

2.2.4 Avaliação e *feedback*

Além do formato de proposição de atividades, os professores ainda têm que se preocupar em estabelecer novos critérios e métodos de avaliação. É importante reconhecer que o ensino *online* não pode ser avaliado de acordo com os critérios usados no ensino presencial (Worley e Tesdell, 2009, p. 3) e que o professor precisa saber, com muita clareza, o que esperar que o seu aluno seja capaz de fazer, como o estudante deve demonstrar que ocorreu a aprendizagem e qual é o nível aceitável de performance dos participantes de seu curso (Stark *et al.*, 2002, p. 2).

Tendo ocorrido a avaliação segue-se, necessariamente, o *feedback*, que deve ser emitido o mais rápido possível, sobre as realizações dos estudantes. Este *feedback*, quando estabelecido de maneira

periódica, ajuda os estudantes a criar hábitos e práticas que os habilitam ao sucesso estudantil (Herman e Banister, 2007, p. 7). O ciclo crítico do *feedback* ocorre quando o estudante indica por ação, gesto ou palavra que recebeu uma mensagem e qual foi o significado atribuído a esta mensagem (Gibson *et al.*, 1998, p. 2) e, no ambiente *online*, isso pode acontecer muito rapidamente. A troca de informações entre professor e aluno é eletrônica, o que implica na necessidade de um maior dinamismo de ambas as partes, bem maior do que o que se verifica nos cursos presenciais, fazendo com que alunos e professores estejam mais conscientes do andamento das aulas. Para os docentes, a avaliação do trabalho dos estudantes de cursos *online* é mais trabalhosa (de Vord e Pogue, 2011, p. 10) porque requer empenho e celeridade na produção do *feedback*, porém, para os alunos, os benefícios desta abordagem são consideráveis (Stark *et al.*, 2002, p. 5). Herman e Banister (2007, p. 1), após ministrar o mesmo curso, simultaneamente, para uma turma em ambiente presencial e outra na modalidade *e-learning*, analisaram e concluíram não haver diferença significativa na qualidade dos projetos finais dos estudantes, correlacionando os bons resultados dos alunos de cursos *online* aos princípios adotados na preparação do curso. Estudos apontados por Worley e Tesdell (2009, p. 3) reafirmam este ponto de vista mostrando que os estudantes aprendem igualmente bem quando os cursos *online* são projetados com base em princípios pedagógicos confiáveis.

Existem modelos pedagógicos que se aplicam à educação tradicional mas que devem ser adaptados para seu uso em *e-learning*. No planejamento de um curso *online* os instrutores devem considerar esta e outras adaptações como, por exemplo, o seu próprio papel perante o curso, deixando de ser a fonte única de informação para se tornar um facilitador ou guia do aprendizado de seus alunos (Worley e Tesdell (2009, p. 2).

2.2.5 Requisitos para o bom planejamento

Projetar um curso *online* requer, entre outras coisas, a compreensão do papel da motivação no aprendizado, avaliação e uso dos conhecimentos prévios dos alunos, noções sobre os estilos de aprendizagem dos participantes, avaliação dos resultados dos alunos e do curso como um todo (Stark *et al.*, 2002, p. 2). Requer ainda a habilidade de representação dos conceitos usando a tecnologia, técnicas pedagógicas que usem a tecnologia de modo construtivo para a distribuição de conteúdo e conhecimentos sobre o que torna um conceito mais fácil ou mais difícil de ser entendido e como a tecnologia pode ajudar os estudantes na apropriação deste conceito. É necessário, ainda, planejar para o reuso, tanto de materiais quanto de estratégias, evitando o retrabalho a cada novo curso (Derntl *et al.*, 2010, p. 2).

Percebemos, portanto, que o planejamento avançado é muito importante para o sucesso dos cursos *online*, talvez mais importante do que de fato é para o ensino presencial, porque, neste ambiente, os erros são bem mais difíceis de serem percebidos e, eventualmente, corrigidos. Experiências de ensino *online* bem sucedidas, frequentemente, são relatadas por instrutores que gastaram um bom tempo com o planejamento do curso, tentando prever e evitar problemas, buscando assegurar que o curso funcione conforme o esperado (Gibson *et al.*, 1998, p. 3).

É difícil, no entanto, que um professor domine todos os conhecimentos acima mencionados simultaneamente. É crítico que o desenvolvimento de *e-learning* se torne mais eficiente, que tenha custos de tempo e de esforço reduzidos (Stark *et al.*, 2002, p. 5). O ideal seria que todo curso *online* tivesse uma assessoria técnica especializada em *design* instrucional que pudesse auxiliar o professor na escolha da abordagem pedagógica mais apropriada para ministrar um conteúdo em particular, identificando e sugerindo a tecnologia mais adequada, facilitando o trabalho do especialista de cada área (Alsofyani e bin Aris, 2011, p. 1), mas esta, infelizmente, não é a realidade da maioria das escolas.

2.2.6 Boas práticas no planejamento de *e-learning*

Como mencionamos acima, seria bom se os professores especialistas pudessem colaborar com os *designers* para efetivamente adaptarem seus cursos para o ensino *online*. Na impossibilidade desta colaboração, no entanto, alguns autores sugerem um conjunto de boas práticas que podem auxiliar

o instrutor no planejamento de cursos *online*. Comece escrevendo as metas e objetivos do curso, mantenha o foco nestes objetivos. Considere, cuidadosamente, o tipo de *mídia* que deve ser usada para atingir estes objetivos. Determine a estrutura e a sequência de apresentação das principais seções do curso. Analise e equilibre os papéis de facilitador do conhecimento, treinador e gerente da informação. Inclua em seu plano oportunidades para a prática do aluno. Utilize a avaliação e o *feedback* como ferramentas para incentivar o estudante ao aprimoramento. E, sempre que puder, invista tempo adicional na revisão da preparação do curso e em sua própria preparação (IMS-BP, 2003).

2.3 Revisão Sistemática sobre Sistemas de Apoio à Preparação de Aulas

Boas práticas no ensino são sempre bem vindas, mas no caso da preparação de aulas para o ensino *online* elas podem não ser suficientes. Vimos que a criação de cursos a serem ministrados *online* é uma tarefa complexa e requer a combinação de habilidades e experiências de diversos campos profissionais (Jones *et al.*, 2000). Professores, principalmente aqueles que não são especialistas da área de Informática, normalmente encontram dificuldades para expressar suas necessidades ou reaproveitar materiais e aulas e, além disso, faltam meios para orientá-los (Clayer *et al.*, 2013). Por este motivo, ferramentas de apoio e orientação que promovam a reflexão sobre a prática docente são necessárias (Verpoorten *et al.*, 2006).

2.3.1 Um Resumo do Protocolo de Revisão

Com o objetivo de entender o contexto sobre o uso de ferramentas de apoio ao professor na preparação de aulas para cursos *online*, realizamos uma revisão sistemática da literatura. Apresentamos abaixo um resumo do protocolo de pesquisa (Apêndice A). Foram considerados neste estudo, inicialmente, artigos publicados a partir de 2003, consultados nas seguintes bases:

- IEEE xplora
- ACM Digital Library
- Compendex
- ISI Web Of Science
- Kluwer Online
- Science Direct - Elsevier
- Springer Link
- Wiley Inter Science Journal Finder

Sobre todas as bases foi pesquisada a seguinte string de consulta: ("*lesson plan*"OR "*lecture plan*"OR "*learning design*"OR "*instructional design*") AND ("*web-based course*"OR "*online course*") AND ("*software*" OR "*computer program*"). Foram recuperados, somados os resultados de todas as bases, 968 artigos.

A estratégia adotada para a leitura dos trabalhos coletados divide a revisão em três passos. Primeiro foram considerados apenas os títulos dos artigos. Se o título de um artigo não estivesse claramente ligado a um dos tópicos de interesse destacados na string de consulta ele seria classificado como inconclusivo e separado para posterior leitura de seu resumo. Esta primeira fase classificou os artigos recuperados, levando em consideração o nível de relação percebida entre o título do artigo com o tema do estudo, nas seguintes categorias: *i.* nenhuma relação, *ii.* título inconclusivo, *iii.* pequena relação e *iv.* relação direta. Artigos da categoria (i) foram imediatamente descartados.

Um exemplo de título que foi incluído na categoria *nenhuma relação com o objeto desta pesquisa* foi "*Simulation of mixed-signal circuits for crosstalk evaluation*". O artigo intitulado "*My Computational Education*" foi considerado de título inconclusivo. Como exemplos de artigos classificados nas categorias pequena relação e relação direta podemos citar "*Online course quality factor analysis based on fuzzy mathematics*" e "*Design and implementation of an undergraduate bioinformatics curriculum in an online environment*" respectivamente.

No segundo passo, foram lidos os resumos de cada artigo classificado como de título inconclusivo. Como resultado desta fase, alguns artigos foram promovidos para as categorias (iii) e (iv). Finalmente, os artigos destas categorias foram lidos na íntegra para determinar o conhecimento atual sobre o objeto da pesquisa.

2.3.2 Descobertas Decorrentes da Revisão Sistemática

A revisão sistemática da literatura nos permitiu concluir que: *i*) não existem ferramentas que apoiem o trabalho do professor na preparação de aulas para cursos *online* e que o permita gerar, em passos simples, um plano de aula; *ii*) o uso de editores e tocadores baseados nas especificações IMS-LD (ver 1.1.1) é feito, geralmente, por especialistas em Informática e *iii*) nenhum estudo empírico foi realizado para verificar o quanto um professor não especialista da área de Informática pode, confortavelmente, trabalhar com estas ferramentas.

Notamos ainda a necessidade de mais e melhores estudos empíricos sobre o tema. Nenhuma revisão sistemática foi recuperada e apenas estudos em inglês foram incluídos.

2.4 Ferramentas e Metodologias para *e-learning* Encontradas na Revisão

Com o crescimento da demanda por ensino *online*, cresce também a necessidade dos professores de incorporar à sua prática estratégias, métodos e ferramentas que os ajudem a desenvolver melhores produtos para cursos e treinamentos na modalidade *e-learning* (Baehr, 2012, p. 2). Às instituições, faltam ferramentas para este fim e, quando elas existem, tendem a focar mais no gerenciamento de conteúdo do que propriamente na preparação e organização didático-pedagógica deste conteúdo (Turani, 2007, p. 15). Como resultado, o professor sem experiência em *design* instrucional acaba adotando o modelo de ensino instrutivista, muito comum em aulas presenciais, em que o estudante apenas observa e segue as instruções do professor, onde o conhecimento é transferido, da mente do instrutor para mente do aprendiz.

2.4.1 Navegador da Internet

O instrucionismo foi uma das primeiras abordagens aplicadas ao ensino online e, surpreendentemente, ainda é muito utilizada (Turani, 2007, p. 41). Muitos instrutores seguem o modelo de publicar material instrucional em páginas da *Web* para que seus alunos acessem, leiam, ouçam ou assistam, e depois, apresentem alguma atividade desenvolvida com base no material estudado. Para a prática deste modelo instrucional, basta que professores e alunos tenham acesso à Internet e a um navegador simples como, por exemplo, o Internet Explorer (2015), o Mozilla Firefox (2003) ou o Google Chrome (2015). Nesta linha de ensino *online* baseado em páginas *Web*, uma das primeiras iniciativas em busca de uma melhor sistematização da organização dos cursos foi o *Automated Lesson Planner* (Cheon *et al.*, 2002). Os autores desenvolveram um sistema *Web* para que os professores criassem seus planos de aula usando o próprio navegador de modo a facilitar a criação e distribuição de suas aulas. Este sistema passou a ser utilizado por professores da rede Coreana de ensino e seus estudos mostraram que estes professores tiveram uma resposta positiva sobre o uso da ferramenta. Iniciativa semelhante foi descrita no trabalho de (Zhi-ying e Hong, 2010), desta vez com foco na aprendizagem colaborativa. O projeto destes autores, *The Web-based Electronic English Lesson Plan Design*, se propõem a integrar as mais modernas tecnologias da informação e

comunicação em um ambiente centrado no estudante, baseado em tarefas por meio da aprendizagem colaborativa dentro e fora da sala de aula. Ambas as iniciativas acima mencionadas têm o mérito de se proporem a instrumentalizar o professor para a tarefa da preparação da aula. Não obstante, ambas falham em prover ao usuário uma estrutura pedagógica adequada e servem apenas como sistemas organizadores de conteúdo.

2.4.2 A abordagem de Templates

Apenas disponibilizar o material de estudo para os alunos em páginas da Internet não é o suficiente, pois existe a necessidade de uma estruturação pedagógica para este conteúdo. Uma possível solução se apresenta com o uso de *templates*. *Templates* de planos de aula são criados na esperança de facilitar e acelerar o trabalho do professor na preparação de suas aulas para cursos *online*. Uma ferramenta que se destaca por adotar esta abordagem é o *TPACK*. Este sistema inclui uma lista contendo vários tipos de atividades e modos de aprendizagem. Escolher a melhor combinação fica a cargo do professor (Alsofyani e bin Aris, 2011, p. 1).

Para verificar a efetividade dos *templates* cinco professores testaram o sistema em uma seção de treinamento. Os autores afirmam que o uso dos *templates* existentes na ferramenta contribuiu para que os professores compreendessem as relações entre pedagogia, tecnologia e conteúdo, e que houve uma redução no tempo de planejamento para o ensino online. Os professores, por sua vez, reportaram uma percepção positiva sobre a efetividade do uso de *templates* para induzir a percepção das relações entre objetivo da aula, formas de atividades e modos de atividades (Alsofyani e bin Aris, 2011, p. 3). Os professores ainda reportaram que, de fato, o *TPACK* pode diminuir o tempo de preparação de aulas pois o usuário só precisa selecionar os recursos que julgar mais adequado, sem ter que fazer quaisquer outras considerações (Alsofyani e bin Aris, 2011, p. 4). *TPACK*, diferentemente das ferramentas já mencionadas, não é um sistema *Web*, ele foi desenvolvido usando-se apenas tabelas do programa Microsoft *Word* (2015).

Além do projeto *TPACK*, há outras iniciativas baseadas em *templates*. Um dos argumentos a favor desta abordagem é a padronização de documentos produzidos por professores diferentes para utilização, eventualmente, no mesmo curso. Normalmente, cada professor adota um estilo pessoal na preparação de material instrucional. Consequentemente, os documentos resultantes são muito diversos, o que dificulta a integração e reuso de material criado por membros distintos de uma equipe (Stark *et al.*, 2002, p. 3). Segundo estes autores, a principal finalidade do uso de *templates* é o encorajamento à consistência do material instrucional de cursos coordenados por vários especialistas. de Vord e Pogue (2011, p. 5) relatam que todos os cursos produzidos pelo corpo docente da *Washington State University*, por serem criados com o auxílio de *designers* instrucionais, fazem uso de *templates* afim de se obter consistência na navegação entre os diversos cursos e facilitar a interação do professor especialista de cada área.

A crítica que fazemos a este modelo de preparação de aulas está ligada à rigidez que o mesmo impõem ao professor. Embora os estudos analisados mostrem um ganho na velocidade de preparação das aulas, a perda em termos de autonomia do professor é significativa.

2.4.3 Educacional Modelling Language (EML)

Até 2003, não havia sido criado nenhum sistema notacional eficiente que permitisse ao usuário codificar, inequivocamente, unidades de estudo como cursos, seções de cursos e programas de estudo; foi então que surgiu a Linguagem de Modelagem Educacional (EML). A EML descreve não apenas o conteúdo das unidades de estudo (texto, tarefas, testes, avaliações), mas também os papéis, as relações, interações e atividades de professores e estudantes, ou seja, sua flexibilidade permite modelar uma grande variedade de estilos e de abordagens pedagógicas. Por definição, EML é um modelo semântico de informação usado para descrever o conteúdo e o processo de aprendizagem previsto para uma unidade de estudo, a partir de uma perspectiva pedagógica, com o objetivo de apoiar reuso e interoperabilidade [Rawlings, 2002].

2.4.3.1 Requisitos para a modelagem educacional

Para apoiar reuso e interoperabilidade, algumas características são necessárias a qualquer linguagem que pretenda modelar unidades educacionais. O estudo realizado por [Koper, 2001], elencou onze requisitos que uma linguagem de modelagem educacional deve satisfazer:

1. **Formalização:** EML deve ser capaz de descrever modelos pedagógicos de maneira formal, de modo que a leitura e interpretação por máquina seja possível.
2. **Flexibilidade pedagógica:** EML deve ser capaz de descrever unidades de estudo que sejam baseadas em diferentes teorias e modelos de aprendizagem e instrução.
3. **Tipificação de objetos de aprendizagem:** EML deve ser capaz de expressar o significado semântico de cada objeto de aprendizagem no contexto da unidade de estudo.
4. **Completeza:** EML deve ser capaz de descrever a unidade de estudo completamente, incluindo todos os tipos de objetos de aprendizagem, o relacionamento entre os objetos e as atividades e o fluxo de trabalho de todos os usuários com os objetos de aprendizagem.
5. **Reprodutibilidade:** EML deve descrever as unidades de estudo de modo que seu uso repetido seja possível.
6. **Personalização:** EML deve ser capaz de descrever aspectos de personalização de modo que materiais e atividades de aprendizagem possam ser adaptadas com base em preferências, conhecimentos prévios e necessidades educacionais.
7. **Transparência ao meio:** a notação das unidades de estudo, onde possível, deve ser transparente ao meio, de modo que ela possa ser usada em diferentes formatos de publicação, como a *web*, artigos, *e-books*, dispositivos móveis, etc.
8. **Interoperabilidade e Sustentabilidade:** separação entre os padrões de descrição e as técnicas de interpretação. Por meio desta separação, investimentos em desenvolvimento educacional se tornarão resistentes às mudanças técnicas e a problemas de conversão.
9. **Compatibilidade:** EML deve se adaptar aos padrões e especificações existentes.
10. **Reusabilidade:** EML deve tornar possível identificar, isolar, descontextualizar e intercambiar objetos de aprendizagem úteis e reusá-los em outros contextos.
11. **Ciclo de vida:** EML deve possibilitar a produção, mutação, preservação, distribuição e arquivamento de unidades de estudo e todos os objetos de aprendizagem que elas contêm.

2.4.3.2 A EML e o ensino *online*

Três dos requisitos acima são fundamentais para o uso de EML em ambientes de aprendizagem *online* (Hermans *et al.*, 2003):

a) Formalismo

Este é o requisito mais importante para o uso da EML em ambientes de aprendizagem online, pois garante que o arquivo resultante das definições de uma unidade de estudo seja processável por computador. Este requisito implica que EML deve ser uma linguagem formal, com seu próprio alfabeto, palavras e sintaxe.

b) Flexibilidade Pedagógica

A importância deste requisito decorre da constante necessidade de adaptação de sistemas de aprendizagem mediante o surgimento de novos paradigmas de aprendizagem como, por exemplo, Aprendizado Baseado em Competências (Schlusmans, 1999), Aprendizado Colaborativo (Dillenbourg, 1995)

e Abordagens de Aumento de Desempenho (Robinson, 1995). Para apoiar a estes novos paradigmas educacionais e a outros que porventura venham a ser criados, ambientes de aprendizagem precisam ser ricos, flexíveis e acessíveis a qualquer tempo e de qualquer lugar. No entanto, a maioria destes ambientes não reconhece a diversidade de modelos pedagógicos; eles, em geral, fornecem (implicitamente) um modelo pedagógico estático ou sequer levam esse ponto em consideração. Com o uso de EML esta característica fica garantida.

c) Interoperabilidade

Instituições educacionais frequentemente recebem investimentos em infraestrutura e a rápida mudança nas tecnologias adotadas pode ser um problema, especialmente quando o desenvolvimento e a aplicação dos cursos estão diretamente ligados a esta tecnologia. A maioria dos ambientes digitais de aprendizagem armazena seu conteúdo em formatos proprietários. Como resultado, torna-se difícil ou mesmo impossível fazer a adaptação de materiais criados em um ambiente para o uso em outro ambiente [Koper, 2004] e, frequentemente, a única solução é fazer a conversão manual do conteúdo, o que é uma tarefa, geralmente, demorada. Às vezes este problema surge até mesmo na atualização de um software, por falta de retrocompatibilidade. Estes problemas induzem a demanda por soluções interoperáveis. Interoperabilidade pode ser definida como a habilidade de um sistema ou um produto funcionar com outros sistemas ou produtos sem esforço especial por parte do usuário. O ponto chave desta questão é criar e gerenciar informação de modo que as oportunidades de intercâmbio e reuso dos dados, interna ou externamente à instituição, sejam maximizadas. O raciocínio acima nos leva a entender a importância da interoperabilidade e notar que é um requisito fundamental da EML.

2.4.3.3 A utilização da EML nos dias atuais

É importante notar que EML não é a única linguagem de modelagem educacional, nem mesmo o foi nos dias de sua criação. No entanto, EML foi a linguagem mais bem sucedida para descrever unidades de estudo com apoio tecnológico.

EML foi uma linguagem tão bem sucedida que serviu como base para a especificação de outra linguagem de modelagem educacional, a *IMS Learning Design*. Hoje, seu uso individual como linguagem é, praticamente, inexistente, devido ao sucesso na adoção do IMS LD.

2.4.4 IMS Learning Design

Em 2003 o IMS Global Learning Consortium lançou a primeira versão da especificação IMS Learning Design *IMS-LD (2003)*, que se tornou um marco importante no desenvolvimento recente na área de ensino *online*. O IMS-LD alterou o foco do ensino *online* enfatizando as atividades de aprendizagem em detrimento do foco anterior, que recaía sobre os objetos de aprendizagem (Durand e Downes, 2009).

Learning Design é, antes de qualquer coisa, uma linguagem (Durand e Downes, 2009). Sua especificação, IMS-LD, foi projetada para promover a interoperabilidade e o intercâmbio de material instrucional de modo a facilitar o reuso de estratégias de ensino e metas educacionais. Seu objetivo era fornecer um framework de elementos que pudessem descrever qualquer unidade de ensino-aprendizagem de maneira formal [Rodríguez, 2004]. A ideia fundamental era associar conteúdo educacional com informação que descrevesse também a estratégia instrucional empregada de modo consistente e interpretável por máquinas. IMS-LD revolucionou o ensino online por focar no processo educacional ao invés de se preocupar com uma simples coleção de conteúdos ordenados a serem estudados (Durand *et al.*, 2010).

A especificação IMS-LD visa representar o learning design de cada unidade de aprendizagem de modo semântico, formal e interpretável por computadores. Uma unidade de aprendizagem pode ter qualquer granularidade e pode ser, desde um curso até uma simples lição. Um *learning design* é definido como a descrição do processo de ensino-aprendizagem que é parte da unidade de aprendizagem [ref]. IMS-LD fornece uma meta-linguagem para modelar processos de aprendizagem com

base nos conceitos de Learning Design. O produto da modelagem baseada em IMS-LD é um arquivo XML que pode ser executado por algum *player* que reconheça sua sintaxe e especificações.

2.4.4.1 *Learning Design* da IMS vs. *learning design* genérico

O conceito de *learning design* e a especificação IMS-LD são, às vezes, difíceis de entender. A expressão *Learning Design*, com iniciais maiúsculas, se refere à implementação do conceito de *learning design* realizada pela IMS, ao qual temos nos referido por IMS-LD, no entanto, o conceito de *learning design* é muito mais amplo. Uma forma elegante de entender o significado de *learning design* foi apresentada por (Britain, 2003), mostrando que se usa a expressão *learning design* alguém pode estar se referindo a:

- O conceito de *learning design*
- A implementação do conceito realizada pela IMS Global Consortium
- As realizações técnicas em torno da implementação do conceito

O conceito de *learning design* surgiu na literatura nos anos 90 trazendo a ideia de que *designers* e instrutores precisam escolher por si mesmos o melhor conjunto de experiências de aprendizagem, construtivistas e/ou comportamentalistas, para seus cursos *online* [Conole, 2005]. Mas o conceito de *learning design* é, provavelmente, tão antigo quanto o conceito de ensino e pode ser definido como a ?descrição do processo de ensino-aprendizagem definido para uma unidade de aprendizagem? [Chellman, 2000]. Um professor preparando um curso é um *learning designer* e praticar *learning design* deveria ser tão fácil quanto preparar um curso ou uma aula [Koper, 2006].

2.4.4.2 O modelo teatral de *learning design*

A especificação IMS Learning Design define atividades de aprendizagem que incluem simultaneamente vários papéis que podem ser assumidos por vários atores em um dado ambiente. Logo, IMS-LD pode ser visto como um *script* no qual cada um dos vários participantes assume seu papel, como se estivesse sob o comando de um diretor teatral. Essa forma de definição do fluxo de trabalho (Figura 2.1), também chamada de metáfora teatral do IMS-LD, tem um profundo impacto no seu modelo de informação.

Os conceitos de papéis e ambiente de aprendizagem são essenciais ao IMS-LD e o distinguem de seus predecessores tais como o IMS Simple Sequencing IMS-SS (2001). Diferente do IMS-LD, IMS-SS não é capaz de expressar atividades que incluem mais de um papel. Logo, IMS-LD pode expressar atividades colaborativas de modo que especificações anteriores não podiam.

2.4.4.3 A Anatomia do Learning Design

Os componentes do Learning Design identificados pelo grupo de trabalho que desenvolveu o IMS-LD, derivados da análise feita por Koper e Manderveld (2004) em seu trabalho para a definição da EML, se baseiam no conceito de Unidade de Aprendizagem ou Unidade de Estudo. Esta é a menor unidade que satisfaz um ou mais objetivos de aprendizagem e, na prática, pode ser um curso, um módulo, uma lição ou uma única atividade, tal como uma discussão em tempo real ou um fórum.

2.4.4.4 Elementos do Learning Design

Para qualquer unidade de aprendizagem alguns ou todos os seguintes elementos precisam ser descritos em um learning design da IMS:

Objetivos de Aprendizagem: Um ou mais objetivos a serem atingidos com a unidade

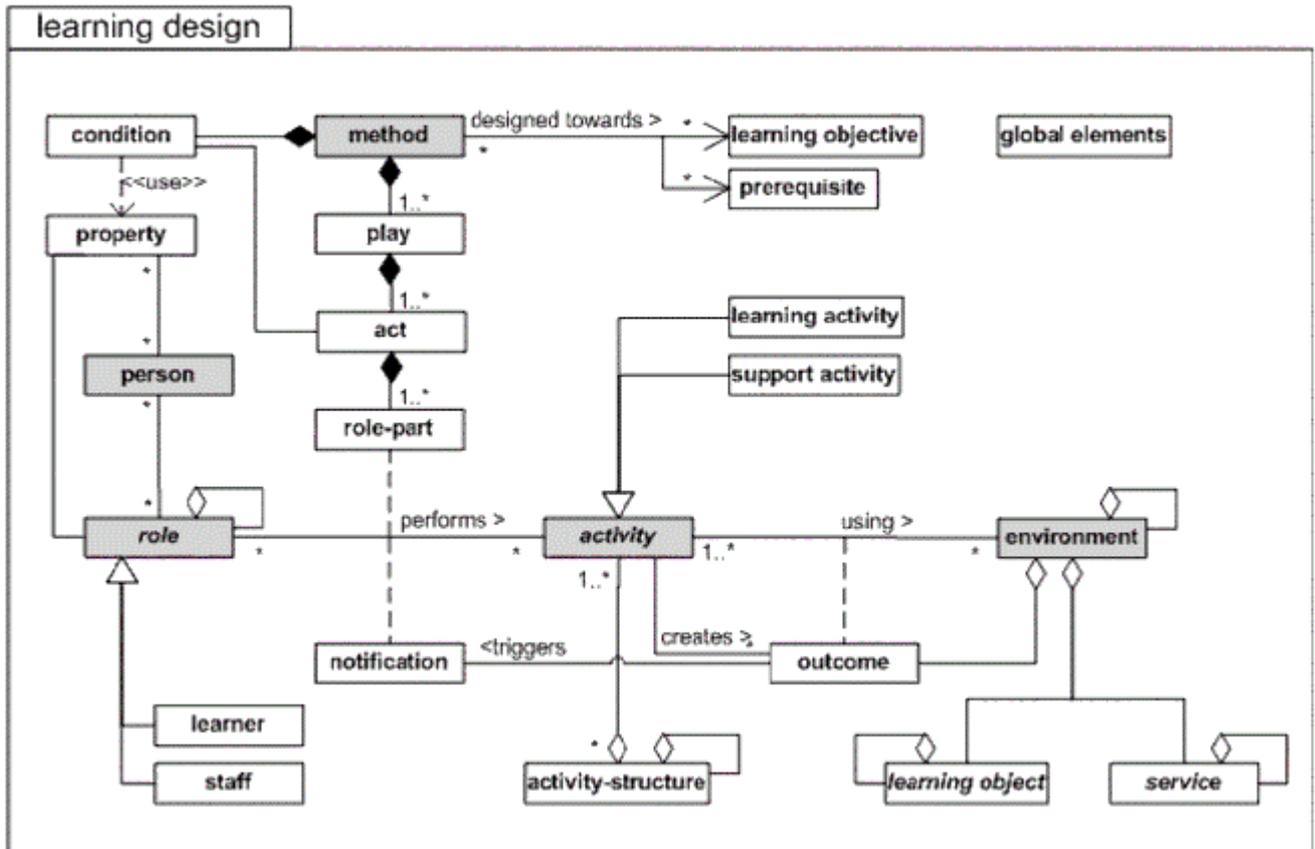


Figura 2.1: Diagrama do Modelo Teatral da IMS-LD.

Papéis: Há dois tipos de Papéis usados para representar as pessoas, estudantes ou especialistas. Indivíduos específicos não são componentes generalizáveis, mas Papéis são. Logo os papéis são especificados no design ao invés das pessoas.

Atividades: Podem ser de dois tipos, atividades de aprendizagem e atividades de suporte.

Estruturas de Atividades: Atividades podem ser agregadas usando estruturas de atividades. Estas estruturas podem referenciar outras estruturas e também unidade de aprendizagem externas.

Ambiente: O ambiente contém dois tipos de elementos básicos:

- Objetos de Aprendizagem que, tipicamente, são URLs para conteúdo externo, ferramentas ou testes como metadados opcionais.
- Serviços disponíveis no ambiente de execução mas que não podem ser especificados em tempo de design, por exemplo, fóruns de discussão, salas de chat, ferramentas de monitoramento e outras características oferecidas pelos ambientes virtuais de ensino.

Alguns dos componentes generalizáveis do design e os objetivos de aprendizagem descritos acima precisam ser associados a instâncias específicas em tempo de design, instanciação ou execução, dependendo do contexto. Para se obter esta associação são usados elementos chamados de Recursos.

Recursos: Recursos podem ser de cinco tipos: conteúdo web, conteúdo IMS-LD, pessoa, serviço ou dossiê. Finalmente, o learning design precisa especificar como o aprendizado e as atividades de apoio realizadas por diferentes atores usando serviços e objetos de aprendizagem são organizadas

em um fluxo de trabalho (workflow). Esta funcionalidade é oferecida pelo elemento Método.

Método: Consiste de uma ou mais Peças que contêm uma sequência de Atos. Cada ato contém uma ou mais cenas. Cada cena associa um Papel com uma atividade ou Estrutura de Atividade. Como vemos acima, a especificação Learning Design usa a metáfora de uma peça teatral para descrever o fluxo de trabalho (*workflow*) envolvido em um cenário de ensino e aprendizagem. O fluxo de trabalho é fundamentalmente sequencial como os atos são sequenciais, mas pode haver um comportamento mais complexo que uma única sequência, o que significa que ramificações e atividades simultâneas por subgrupos é possível.

2.4.4.5 Os Níveis de Learning Design

Para facilitar a produção da especificação e sua subsequente implementação, a IMS-LD foi dividida em três partes, conhecidas como Nível A, Nível B e Nível C.

Learning Design Nível A

Este nível inclui todos os elementos descritos na seção anterior. O principal valor agregado ao ensino online pelo Nível A decorre do fato de que ele define Atividades e Papéis como componentes reusáveis que podem ser inseridos em um fluxo de trabalho (workflow) usando o elemento Método, além de permitir que Serviços tais como email e conferências sejam especificados em tempo de design. Estas características são uma diferença qualitativa entre o IMS content-packaging e SCORM, que não incluem os conceitos de atividades ou papéis e apenas trabalha com conteúdos de modo que quando eles são inseridos em um AVA são completamente separados de discussões e outras tarefas colaborativas.

Learning Design Nível B

Este nível permite a inclusão de propriedades e condições. Dois tipos de propriedades foram propostos: Internas e Externas. As propriedades internas são importantes porque interferem no fluxo de eventos; a definição de seus nomes e intervalos de valores deve ser feita em tempo de design. A inclusão de propriedades externas, por sua vez, é importante para a adaptação do design com base nas propriedades individuais do estudante como aquelas fornecidas pelas especificações de Acessibilidade e do IMS-LIP (Learner Information Package) (figura 2.2).

Learning Design Nível C

Este nível acrescenta a capacidade de notificação ou *messaging* entre componentes do sistema e entre papéis. Atividades podem, portanto, ser definidas como consequência de mudanças dinâmicas nos perfis dos estudantes e/ou de eventos gerados no desenvolvimento das atividades de aprendizagem.

De modo geral, o Nível C permite a automação do fluxo de atividades, permitindo sua liberação mediante o término de alguma tarefa específica, ao invés de exigir que seja seguido um fluxo de trabalho pré-planejado. Eventos colaborativos podem ser incluídos onde as atividades dos papéis sejam dependentes do estado das atividades de outros. É possível, portanto, que as atividades sejam projetadas como uma rede de eventos e não como uma cadeia fixa de eventos. Uma consequência desta disposição de eventos é que as atividades definidas para cada estudante não são mais previsíveis, pois elas dependem agora do andamento da colaboração. Nos níveis A e B a ordenação das atividades é previsível embora, naturalmente, no nível B com o uso de propriedades e condições o fluxo de aprendizagem possa se tornar condicional.

2.4.4.6 Construindo um learning design

O Guia de Boas Práticas (GBP) descreve uma sequência de passos que caracterizam o desenvolvimento de um learning design para uma unidade de aprendizagem.

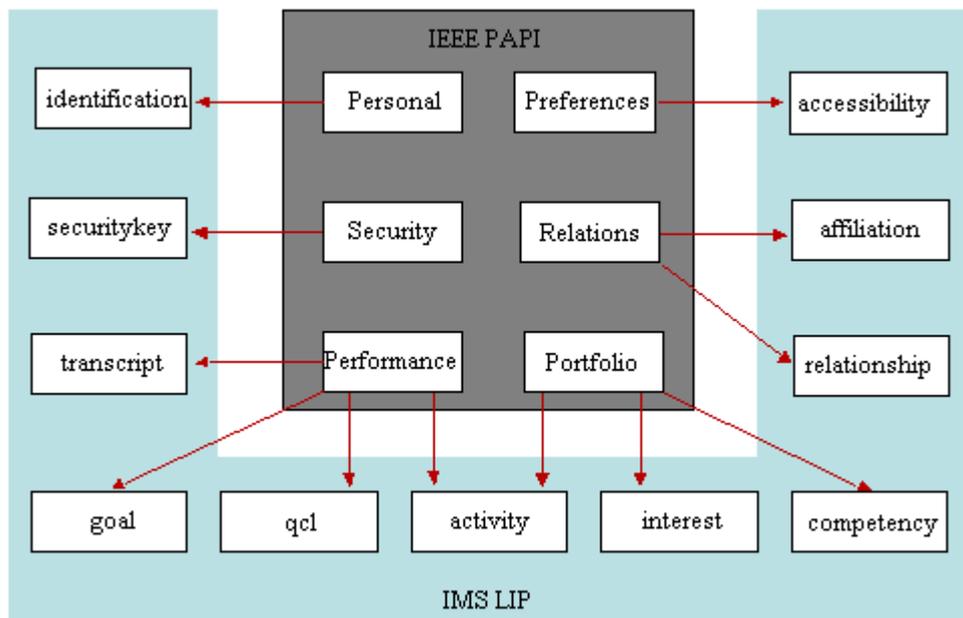


Figura 2.2: IMS-LIP (Learner Information Package)

| Learning Design Process | IMS-LD Process |
|---|---|
| simple test | |
| Define Learning Objectives | Specify Learning Objectives |
| Develop narrative description of learning and teaching scenario | Not defined within current scope |
| Create learning activity workflow from Narrative description | Create a Method using Play, Acts and Role-Parts |
| Assign resources, tools and people to activities | Specify Roles, Resources Environment and Services |
| Running (real-time) | Use a Learning Design aware player |
| Learner support and on-the-fly adaptation | Not Defined |
| Reflecting (including sharing outputs for peer reflection) | Not Defined |

Tabela 2.1: Atividades-chave em Learning Design (Britain, 2004)

1. A primeira tarefa é analisar um problema educacional específico como um caso de uso e então torná-lo um cenário descrevendo os objetivos de aprendizagem e as tarefas ou atividades estabelecendo a ordem básica de eventos que podem ser capturados na forma de narrativa.

2. É sugerido no GBP que a narrativa seja então convertida para um diagrama de atividade UML. Este diagrama UML então forma a base para a criação do documento XML que implementa a especificação IMS-LD.

3. Então, o conteúdo real (recursos) pode ser desenvolvido e, finalmente, um pacote de conteúdos pode ser criado e incorporado ao learning design.

Este resumo do processo de criação de um *learning design* reafirma a necessidade do desenvolvimento de software de apoio para que o professor comum seja capaz de utilizar as especificações da IMS.

2.4.4.7 Ferramentas de Apoio a IMS-LD

Mesmo que os professores fossem experientes no desenvolvimento de cenários na forma de narrativa, visto que a maioria não é (Derntl *et al.*, 2010), poucos contemplariam a possibilidade de transformá-los em diagramas UML e, depois, em XML em conformidade com a especificação IMS-LD. Logo, software que facilite a autoria e execução de learning designs é indispensável (Dutra *et al.*, 2005) (Valentín *et al.*, 2007).

Para que se possa tirar vantagem do paradigma oferecido por IMS-LD, dois conjuntos de ferramentas são necessários. A criação de unidades de ensino precisa de ferramentas de edição que sejam simples e intuitivas, permitindo que o usuário mediano consiga definir e combinar recursos facilmente. Depois de prontas, estas unidades de ensino precisam ser executadas em algum ambiente gerenciador de aprendizagem que seja capaz de interpretar as definições de IMS-LD e instanciar todos os recursos incluídos na unidade.

2.4.4.8 Sistemas Editores e Tocadores de Learning Design

O processo de design e a execução de uma unidade de aprendizagem requer a seleção de uma ferramenta de autoria e/ou de execução compatível com as especificações de IMS-LD. Neste capítulo são apresentadas e avaliadas algumas destas ferramentas.

a) Ambientes de criação

Ferramentas como o Reload Learning Design Editor (Reload, 2003) ou Ask Learning Design Toolkit (ASK-LDT, 2005), fornecem uma interface gráfica de alto nível que cobre inteiramente a especificação. No entanto, esta abordagem carece de meios práticos para gerenciar a complexidade conceitual e requer do professor um profundo conhecimento sobre a especificação, tornando o uso destas ferramentas muito difícil para usuários que não sejam especialistas em IMS-LD. Outras ferramentas, como Lams (2003), oferecem uma interface gráfica baseada em formalismos e orientada às habilidades dos usuários. Como a representação não é baseada em IMS-LD, a tradução para sua especificação pode implicar na perda de algumas informações sobre o fluxo de aprendizagem definido pelo usuário. Entre estas duas categorias, há editores como o Collage (1990) que são baseados em *templates* para produzir unidades de aprendizagem. Estas ferramentas oferecem uma interface gráfica simplificada para que usuários não especialistas em *learning design* possam criar unidades de aprendizagem, mas sua principal restrição é derivada do fato de que o conjunto de padrões previamente definidos pode não satisfazer às necessidades de todos os usuários.

b) Ambientes de Execução

Complementar ao processo de autoria, unidades de aprendizagem precisam ser distribuídas e instanciadas em algum ambiente de execução. O Coopercore (2003) é o sistema usado como implementação de referência e seu objetivo é facilitar a adoção da especificação. Mesmo cobrindo toda a especificação a ferramenta, no entanto, não é orientada para o usuário final. Na prática, esta ferramenta é usada para testar unidades de aprendizagem prontas e para verificar a compatibilidade dos ambientes de execução com a especificação. Há vários outros tocadores (*players*) e, a maioria deles, se baseia no CopperCore. Por exemplo, os programas Service-based Learning Design (SLeD, 2004) e o Player Reload (2003) fornecem uma interface gráfica para o sistema de execução, permitem a importação de unidades de aprendizagem e realizam todas as tarefas relacionadas à distribuição das unidades. Além destas duas ferramentas, merece destaque o GRAIL (GRADient runtime environment for IMS-LD at .LRN), que é uma implementação das especificações do ambiente de execução da IMS dentro do ambiente .LRN. Ele permite importar e distribuir unidades de aprendizagem. Oferece uma interface gráfica com o usuário para realização de tarefas administrativas e faz a inclusão dos recursos requeridos pelas unidades de aprendizagem usando serviços e ferramentas existentes no .LRN. Naturalmente, há várias outras ferramentas disponíveis no mercado. No Apêndice C apresentamos uma avaliação das ferramentas mais citadas pelos pesquisadores de *learning design* usando como base uma metodologia proposta por Britain (2003) afim de obter uma

avaliação pragmática destas ferramentas.

2.4.5 Conclusão

A especificação IMS-LD é adequada para a execução por computador, porém a sua complexidade é um fator limitante ao seu uso pela grande maioria dos professores. Seu estudo, no entanto, foi inspirador para o autor desta tese e permitiu perceber que esta opção é inadequada para os propósitos deste trabalho o que, de modo algum, tira o mérito da abordagem IMS-LD. Percebemos ainda que pela crítica geral encontrada nos artigos científicos analisados, esta especificação, mesmo já tendo se passado doze anos de seu lançamento, que ocorreu em Janeiro de 2003, não encontrou uma boa recepção na comunidade de professores e instrutores que ministram cursos online e que quanto mais o tempo passa menores se tornam as possibilidades deste quadro se reverter. A análise das ferramentas que implementam IMS-LD, editores e tocadores, deixou claro que, por melhores que tenham sido o rigor e o cuidado na implementação da especificação, se o sistema resultante não for facilmente acessível ao usuário, ocorrerá um desperdício de esforço. Com estes resultados em mente, prosseguimos em busca de opções que pudessem, realmente, tornar mais simples o trabalho de preparar aulas para o professor que ministra cursos *online*.

Capítulo 3

Padrões Pedagógicos

Neste capítulo discutimos o conceito de padrão pedagógico, suas origens e implicações para o ensino. Apresentamos também algumas iniciativas que buscaram incentivar o uso de padrões pedagógicos na prática docente cotidiana.

3.1 Introdução

Em várias áreas do conhecimento é comum enfrentarmos problemas recorrentes, que com frequência estão nos levando a reelaborar soluções semelhantes e de mesmo efeito prático (Medeiros *et al.*, 2007, p. 1). São exemplos de problemas recorrentes a desmotivação dos alunos, a seleção e sequenciamento das atividades, a avaliação da aprendizagem e etc. (Jaffry, 2003, p. 1). Estes problemas afetam tanto professores experientes quanto professores iniciantes, mas no caso destes últimos, eles se agravam.

Muitos professores, particularmente os que têm menos experiência, ministram suas aulas do mesmo modo que seus antecessores o fizeram: falando na frente de uma sala e escrevendo no quadro negro (Carle *et al.*, 2007, p. 1). Em muitos casos eles assim o fazem porque, simplesmente, não foram ensinados a ensinar e não é raro o caso de professores que acabaram em sala de aula por acidente (Eckstein *et al.*, 2001, p.1).

Existem opções de treinamento para professores nas modernas técnicas pedagógicas, porém poucos investem tempo no aperfeiçoamento de sua didática (Carle *et al.*, 2007, p. 4). Nas universidades o foco recai sobre a pesquisa, e o ensino, em geral, é percebido por alguns professores como uma atividade de menor importância, desestimulando-os de investir esforço e tempo consideráveis em treinamento pedagógico (Jones *et al.*, 2000, p. 3). Há, portanto, uma certa resistência dos docentes, particularmente daqueles focados em pesquisa, em frequentar cursos voltados especificamente para a preparação para a docência (Verpoorten *et al.*, 2006, p. 3).

Em uma pesquisa nacional realizada nos Estados Unidos em 1997, Jones *et al.* (2000) mostraram que apenas um terço dos participantes acreditavam que as necessidades de treinamento e desenvolvimento docente estavam sendo satisfeitas. No mesmo trabalho, estes autores apontam algumas razões para a relutância dos professores em participar de cursos preparatórios para a docência, particularmente, de cursos online: 1) percepção de que a pesquisa é mais importante que o ensino; 2) a carga de trabalho do professor não permite sobrar tempo para este tipo de atividade; 3) dificuldade de organizar tempo, data e local que seja conveniente para todos os professores de uma instituição; 4) aversão à tecnologia. O quadro descrito acima indica que existe uma brecha na formação pedagógica dos professores, particularmente em relação ao ensino online, bem como a necessidade de iniciativas de apoio à preparação docente.

3.2 O conceito de Padrão

As soluções para os problemas acima mencionados que são vivenciados pelos professores em sua prática profissional podem diferir ligeiramente de acordo com o contexto em que forem aplicadas,

mas a essência destas soluções pode ser descrita como um padrão. De modo geral, um padrão descreve um problema e o núcleo da solução desse problema, objetivando que a solução proposta para o mesmo possa ser utilizada várias vezes (Alexander *et al.*, 1977).

Um padrão não é a solução completa para um problema, mas sim a orientação suficiente para que aquele que o utiliza personalize sua intervenção a cada reuso (Turani, 2007, p. 46), cada padrão captura a melhor prática para resolver um problema em um determinado contexto (Clayer *et al.*, 2013). Um padrão não é uma receita ou uma documentação dos passos de uma solução, mas sim uma solução representada de modo abstrato (Eckstein *et al.*, 2001, p. 2), uma forma genérica de resolver um tipo de problema de modo adaptável aos casos específicos, uma solução potencial pré-definida para problemas recorrentes em diferentes contextos (Jaffry, 2003, p. 1). Quando usado de maneira correta pode economizar tempo e aumentar a qualidade da solução (Jones *et al.*, 2000, p. 1).

Padrões descrevem um problema e o núcleo da solução deste problema com o objetivo de que esta solução possa ser utilizada várias vezes (Medeiros *et al.*, 2007, p. 1), podem ser vistos também como uma ponte que liga as evidências empíricas, a experiência e a teoria ao problema prático (Verpoorten *et al.*, 2006, p. 1). Padrões são simples e focados em um problema específico (? , p. 2), não são criados ou inventados (Verpoorten *et al.*, 2006, p. 1), derivam da prática e tratam do que funciona, tratam das melhores práticas. Padrões são bons quando resistem à prova do tempo, sendo continuamente reusados (Filho e Derycke, 2009, p. 2) e normalmente evoluem pelo acúmulo de conhecimentos derivados da experiência prática (Siviter e Wheeler, 2004, p.6)

3.2.1 A finalidade do Uso de Padrões

A intenção de Alexander *et al.* (1977) quando criou o conceito de padrão era capturar o conhecimento dos especialistas de modo que este pudesse ser compreendido e apreciado por indivíduos com consideravelmente menos conhecimento em sua área de estudo (Carle *et al.*, 2007, p. 2). De modo simplificado, podemos dizer que um padrão é um registro das lições aprendidas e das melhores práticas dos especialistas de uma área (Jones *et al.*, 2000, p. 5). O poder do uso de padrões vem de sua sólida conexão com a experiência prática. Uma técnica não pode ser considerada um padrão se não foi executada com sucesso e se não é praticada continuamente (Eckstein *et al.*, 2001, p.7).

Appleton (1999) in (Filho e Derycke, 2009, p. 2) explica que padrões são úteis porque a) resolvem problemas do mundo real, b) documentam decisões de projeto, c) reusam a experiência dos especialistas de cada domínio, d) levam ideias de especialistas a novatos e e) criam um vocabulário adequado para discussões sobre cada tipo de problema. Para Medeiros *et al.* (2007, p. 1), além de estabelecer um canal de comunicação padronizado, através de um catálogo, a principal vantagem do uso de padrões é a possibilidade de reutilização de soluções baseadas em experiências bem sucedidas.

3.2.2 A Gênese dos Padrões

Para se tornar um padrão, a solução de um problema deve passar pelo processo de abstração e validação de modo que sua essência seja fatorada em uma forma que seja genericamente aplicável (Eckstein *et al.*, 2001, p. 2). Sua documentação deve ser feita usando-se um formato particular que pode descrever motivação, contexto, recomendações, contraindicações, padrões relacionados, resultados esperados de sua aplicação e etc. O uso consistente de formatos específicos para a descrição de padrões torna mais fácil aprender, comparar e usar os padrões (Jones *et al.*, 2000, p. 5). Uma típica descrição inclui, pelo menos, nome do padrão, um resumo do problema genérico ao qual ele se aplica e uma descrição da solução geral proposta (Siviter e Wheeler, 2004, p. 6). Além disso, a aplicabilidade de um novo padrão deve ser verificada pela documentação de três usos conhecidos. Isso permite que o padrão seja aplicado de modo confiável por pessoas diferentes em diferentes situações (Eckstein *et al.*, 2001, p. 2). Novos padrões, quando aceitos, são mantidos por uma comunidade e não por instituições isoladas; a proposição de padrões é, acima de tudo, uma atividade colaborativa. O objetivo maior da abordagem de padrões é, portanto, oferecer um formato para registrar

o conhecimento e um processo para validá-lo possibilitando o compartilhamento de práticas bem sucedidas de modo que cada prática possa ser usada por diversos indivíduos de diversas maneiras distintas (Eckstein *et al.*, 2001, p. 7).

3.3 Padrões Aplicados ao Ensino

A abordagem de padrões não demorou para se tornar desejável em outros domínios. Seu uso extrapolou a área da Arquitetura, onde teve início, e se tornou prática comum em domínios tão distintos quanto a Engenharia de Software, com os Design Patterns (Gamma *et al.*, 2000) e o desenvolvimento de negócios online, com os Patterns for eBusiness (IBM, 2004), apenas para citar dois bons exemplos.

A educação também viria a se beneficiar fazendo aplicação de padrões ao ensino, criando os chamados Padrões Pedagógicos. Em vez de catalogar soluções eficazes para problemas inerentes à elaboração do projeto de um software ou à criação de empresas de base tecnológica os padrões pedagógicos descrevem soluções para problemas relacionados a atividades de ensino e de aprendizagem (Medeiros *et al.*, 2007), visando criar o contexto onde a aprendizagem possa acontecer (Eckstein *et al.*, 2001, p. 7).

Os padrões pedagógicos nada mais são do que padrões aplicados ao contexto de situações de ensino. Descrevem soluções para problemas relacionados a atividades de ensino-aprendizagem, baseadas na experiência prática fornecida por especialistas. (Medeiros *et al.*, 2007, p. 1). Eles capturam as melhores práticas no domínio educacional ajudando professores a delinear estratégias para superar problemas e dificuldades comuns ao seu trabalho, por exemplo, como motivar os alunos, como introduzir novos conceitos e como fazer o sequenciamento de atividades (Mylonakis *et al.*, 2013, p. 2).

A principal finalidade dos padrões pedagógicos é identificar as práticas de excelência em ensino e registrá-las no formato de padrão, compacto e que possa ser facilmente comunicado àquele que necessita deste conhecimento, de modo a fomentar a criação de um vocabulário comum para a prática e para a pesquisa pedagógica, ser acessível aos instrutores iniciantes e encorajar o reuso de técnicas que estejam solidamente fundamentadas na moderna pedagogia (Carle *et al.*, 2007, p. 2). Apresentar esta informação de modo coerente e acessível pode significar a diferença entre cada novo instrutor necessitar reaprender tudo o que os membros mais experientes do corpo docente já sabem e a fácil transferência e disseminação do conhecimento inerente à comunidade de educadores (Filho e Derycke, 2009, p. 3).

3.3.1 Padrões como Ferramenta Docente

Assim como os padrões propostos por Alexander *et al.* (1977), padrões pedagógicos possuem uma forma sistematizada de descrição tendo, geralmente, as seguintes seções: Problema, Forças (ou Contexto), Solução e Discussão. A seção Problema apresenta uma dificuldade na forma de uma pergunta. As forças são as considerações que levam à solução. A solução responde a pergunta formulada na seção Problema. Na seção Discussão alguns exemplos são apresentados ou os inconvenientes da solução são Comentados. Eventualmente, alguns padrões também apresentam a seção Padrões Relacionados (Eckstein *et al.*, 2001, p. 7) indicando práticas que complementam o padrão atual, contraindicações, recursos necessários e consequências. Diferentes autores usam formas levemente diferentes, inserindo ou omitindo alguns elementos. Todavia, os itens centrais estão sempre presentes embora possam usar uma nomenclatura particularizada de acordo com as preferências dos autores (Filho e Derycke, 2009).

Havendo consistência na documentação, mesmo um professor iniciante quando de posse de dados relacionados às seções dos padrões pedagógicos, normalmente, será capaz de determinar se um padrão se aplica às suas necessidades. Idealmente, a informação apresentada no padrão deve ser genérica o suficiente para que o instrutor possa adaptá-la às suas necessidades sem alterar negativamente a abordagem teórico-pedagógica do padrão (Carle *et al.*, 2007, p. 2). Desta maneira,

padrões pedagógicos podem ser usados como ferramenta para o desenvolvimento do corpo docente se considerados como base para a discussão e o trabalho colaborativo permitindo a estes profissionais identificar e melhor compreender componentes de sua própria prática além de estimular a abertura a novas práticas educacionais e à criatividade (Verpoorten *et al.*, 2006, p. 3).

3.3.2 Catálogos de Padrões Pedagógicos

Vários autores têm catalogado padrões pedagógicos, mas, aparentemente, a coleção mais bem sucedida foi catalogada pelo Pedagogical Patterns Project (PPP, 2000). A primeira versão do catálogo, escrita por Joseph Bergin, tinha dez padrões que foram posteriormente, em 2002, complementados com mais quatro pelo mesmo autor. A intenção original de Bergin era criar uma linguagem de padrões para o desenvolvimento de um curso de Orientação a Objetos, para alunos de graduação em Ciência da Computação na Universidade de Pace, Nova York.

3.4 Relatos sobre o uso de Padrões Pedagógicos

Desde as primeiras iniciativas que buscaram usar a abordagem de padrões no ensino, várias foram as aplicações, visto que as possibilidades de aplicações no ensino são realmente grandes. Nesta seção, visando mostrar a diversidade de aplicações, apresentamos alguns projetos que fizeram uso de PP.

3.4.1 To be continued...

Capítulo 4

Linhas de Produto de Software

Uma das abordagens mais promissoras para a indústria de *software* contemporânea é a de linha de produtos. Esta modalidade de desenvolvimento acelera a produção de famílias de sistemas relacionados destacando suas comunalidades enquanto gerencia os seus pontos de variação. Neste capítulo apresentamos o conceito de Linha de Produtos de Software, discutimos suas características mais importantes no contexto desta tese e apresentamos seu processo de desenvolvimento e as características de sua implementação.

4.1 Introdução

Entre os séculos XVIII e XIX ocorreu a Revolução Industrial. Basicamente, o trabalho artesanal deu lugar à industrialização com o objetivo de tornar a produção mais eficiente. Em consequência, os produtos que antes eram feitos um a um passaram a ser produzidos em massa, em linhas de produção; era o fim da personalização e o início da massificação. Este fenômeno, guardadas as devidas proporções, também se verifica na história do desenvolvimento de software.

No início, década de 1950, o software foi produzido de modo artesanal e "a maior parte do software era desenvolvida e, em última análise, usada pela própria pessoa ou organização. Você escrevia, colocava em funcionamento e, se o software falhasse, era você quem consertava" (Pressman, 1995). Na segunda era, em meados da década de 1960, o software passou a ser desenvolvido para a ampla distribuição e ocorreu o que foi chamado de *A nova Revolução Industrial*. O objetivo era, mais uma vez, a produção em larga escala.

O modelo massificado de consumo se desgastou ao longo do tempo e hoje é possível produzir um automóvel, o grande ícone da Revolução Industrial e da massificação em linha de produção, totalmente configurado de acordo com as preferências de seu comprador (Figura 4.1). Lanchonetes permitem que seus clientes determinem exatamente quais ingredientes devem ser colocados em seus pedidos (Figura 4.2), em oposição à outrora vitoriosa estratégia de produzir alimentos em série usada pelos grandes restaurantes de *fast food*.

Assim como ocorreu com a indústria em geral, não demorou para que a necessidade de personalização de software pressionasse o setor produtivo e ocorresse um movimento de volta à criação produtos que atendessem cada vez mais as necessidades específicas de cada organização ou indivíduo. Estava lançada a semente para a pesquisa e desenvolvimento da abordagem de linha de produtos.

4.2 Produtos de Software : Conceitos Básicos

Uma linha de produtos no catálogo de um determinado fabricante é um conjunto de produtos que compartilham consideráveis similaridades e que são, idealmente, criados a partir de um conjunto de partes reusáveis. Ao invés de oferecer um produto padrão, fabricantes buscam meios de diversificar a produção de modo a atender uma maior quantidade de clientes, preferencialmente, utilizando a mesma cadeia produtiva. O objetivo é compartilhar tantas partes quantas forem possíveis na criação

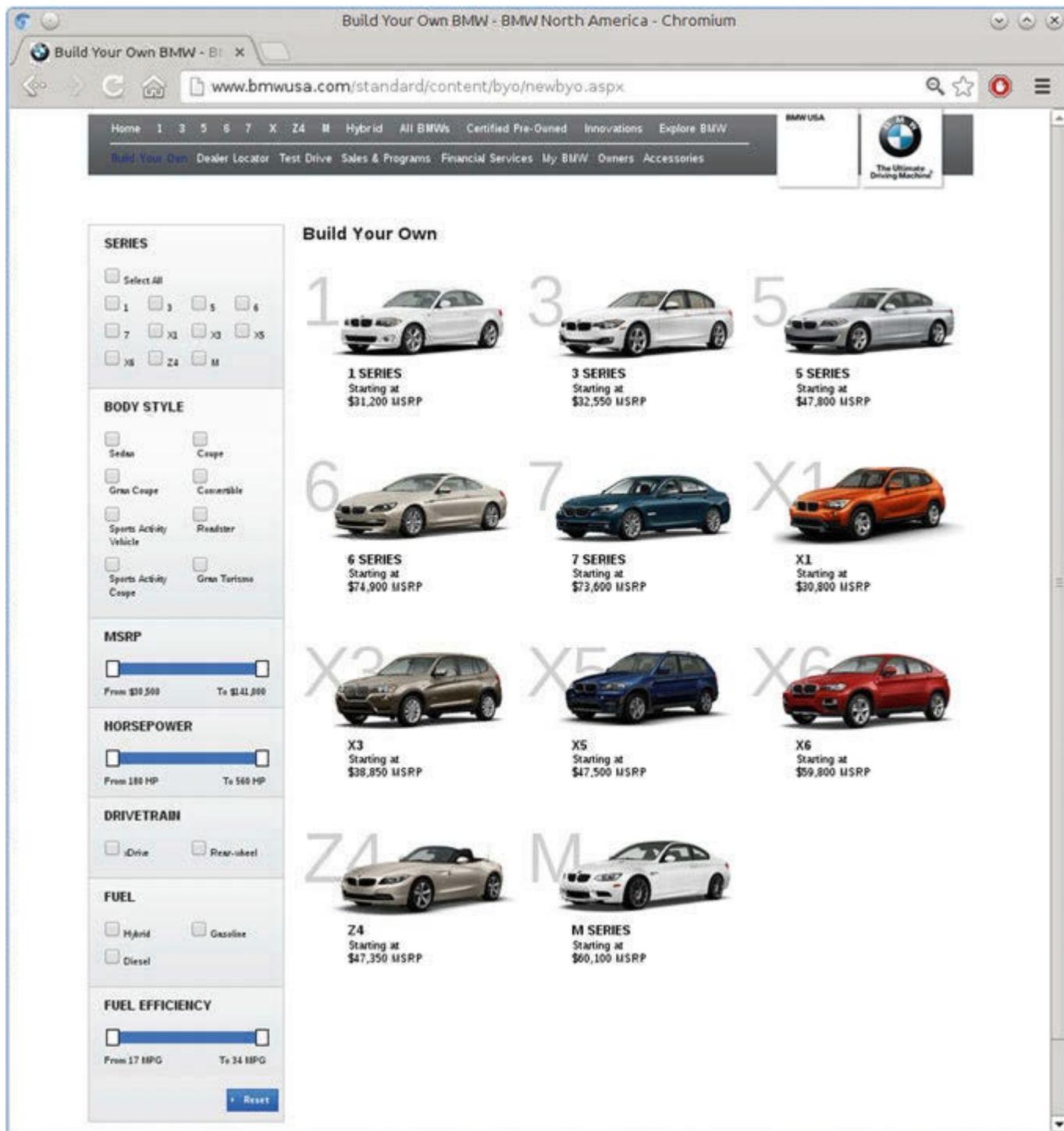


Figura 4.1: Sistema de Configuração de Autômos da BMW

dos produtos de uma linha e, ainda assim, obter variações significativas afim de gerar produtos tão diversos que permitam aos clientes escolher aquele mais adequado às suas necessidades dentre as opções disponíveis (Apel *et al.*, 2013).

De modo similar, podemos pensar em Linhas de Produtos de Software(LPS) como um catálogo de softwares similares em sua natureza, que compartilham a maior quantidade de características possível e que, no entanto, são diferentes entre si. Neste contexto, ao invés de cada sistema ser produzido desde o início, eles devem ser montados a partir das partes comuns e reusáveis. Ao invés de um sistema de software ser criado a partir de um conjunto fixo de requisitos para todos os possíveis usuários, ele deve ser adaptado aos requisitos de cada cliente. A necessidade de individualização

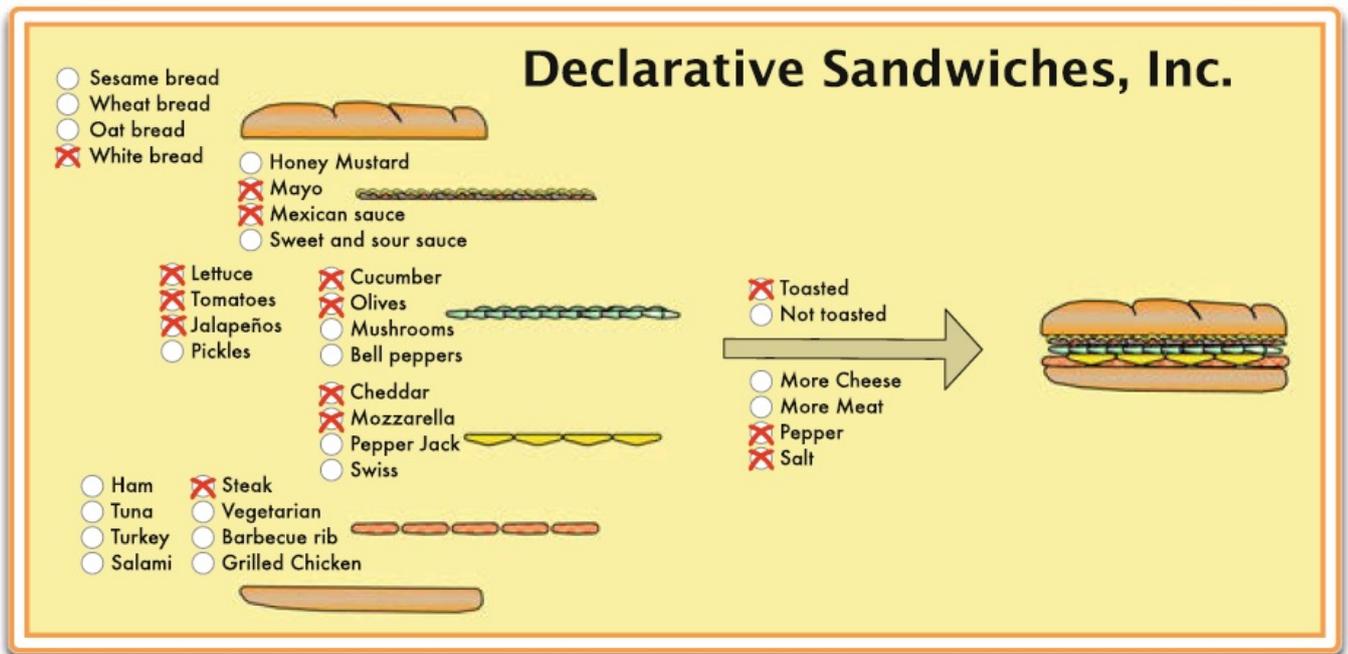


Figura 4.2: Sistema de Configuração de Lanches de um Restaurante Hipotético

surge de diversas fontes tais como funcionalidades específicas, plataforma de execução, propriedades não funcionais e etc. A abordagem de linha de produtos de software oferece uma opção de padronização em massa construindo soluções individuais baseadas em catálogo de componentes reusáveis (Apel *et al.*, 2013).

4.3 Benefícios das LPSs

A abordagem de linha de produtos oferece alguns benefícios que são, em geral, difíceis de se obter no paradigma básico de desenvolvimento de software. Segue abaixo uma lista de vantagens do uso de LPS adaptada de (Apel *et al.*, 2013, p. 9):

1. **Personalização** - A criação de produtos em LPS facilita a personalização permitindo que os requisitos específicos de uma grande quantidade de indivíduos sejam satisfeitos. Ao invés de oferecer uma pequena quantidade de produtos padronizados uma empresa de desenvolvimento de software pode oferecer um conjunto amplo de soluções personalizáveis.
2. **Redução de Custos** - Ao mesmo tempo em que atende uma gama maior de clientes, organizações que baseiam seu desenvolvimento em LPS não têm que pagar pela criação de cada sistema do início; quanto maior for a linha de produtos, menor será o seu custo efetivo.
3. **Aumento de Qualidade** - Os componentes básicos de uma LPS são testados a cada vez que são usados em um novo produto. Isso permite que erros, que porventura não tenham sido percebidos em produtos anteriores, possam ser descobertos e corrigidos. O resultado é que a partir do uso repetido dos componentes comuns falhas vão se tornando cada vez mais raras e a qualidade dos produtos da linha aumenta.
4. **Prazo de Distribuição** - Enquanto o software padronizado pode ser obtido em pacotes, o software personalizado, escrito desde o início tem um prazo de entrega, geralmente, longo. Se um cliente escolher apenas entre características que já estão implementadas em um LPS a entrega pode ser, literalmente, imediata. Mesmo que este hipotético cliente escolha funcionalidades inexistentes, a implementação e incorporação desta funcionalidade a uma LPS pode

ser muito rápida pois a base de desenvolvimento, de componentes reusáveis, provavelmente, formará a maior parte deste novo produto.

4.4 Abordagem de *features*

Linhas de Produto de Software facilitam a industrialização do processo de desenvolvimento de software. Reusando componentes, um fabricante de software pode criar rapidamente novos sistemas com base nos requisitos específicos de um cliente. Para que isso ocorra, no entanto, é necessário utilizar o conceito de *features* de software. A ideia principal da orientação a *features* é organizar e estruturar todo o processo da linha de produtos e todos os artefatos envolvidos em torno de *features*. O tratamento explícito de *features* na fase de requisitos, arquitetura, implementação e verificação e validação pode melhorar o gerenciamento do desenvolvimento de *software*.

Apel *et al.* (2013, p. 18) define *feature* como uma característica ou comportamento visível ao usuário final. *Features* são usadas para distinguir os produtos de uma linha especificando comunicações e variantes para orientar todas as fases do ciclo de vida do *software*. O *portfolio* de produtos de uma LPS é definido pelas *features* que contém e pelas relações existentes entre estas. Um produto específico é identificado por um subconjunto das *features* de sua LPS, chamado de seleção de *features*.

4.5 Projetando uma LPS

Quando o objetivo é criar uma LPS se faz necessário analisar os requisitos de sistemas similares, mas não idênticos, que potencialmente farão parte da família. O principal fator de sucesso no desenvolvimento de uma linha de produtos é definir o foco apropriado sobre um domínio particular e de escopo bem definido. De modo simplificado, podemos pensar sobre um domínio como a área de conhecimento sobre a qual se desenvolve uma LPS. A importância da definição clara do escopo sobre um domínio é fundamental, pois quanto maior o escopo, menor a quantidade de pontos em comum entre as possíveis aplicações de uma linha de produtos, tornando substanciais as diferenças e, conseqüentemente, reduzindo as possibilidades de reuso. O processo de desenvolvimento de linhas de produto de software deve, portanto, levar em conta estas peculiaridades que são cruciais: o tratamento da variabilidade e o reuso sistemático da implementação de artefatos.

4.6 Processo de Desenvolvimento da LPS

As características específicas das LPS's levam, naturalmente, a uma separação entre a engenharia de domínio e a engenharia de aplicação, bem como também separa o espaço do problema do espaço da solução. A Figura 4.3 ilustra esta separação e cada um de seus elementos recebe abaixo uma descrição adaptada de (Apel *et al.*, 2013, 20).

4.6.1 Etapas do Desenvolvimento de LPS

Engenharia de Domínio é o processo de analisar o domínio de uma linha de produtos e desenvolver artefatos reusáveis. Não resulta em um produto de software específico, mas prepara artefatos para serem usados em múltiplos, se não todos, os produtos de uma linha. Seu objetivo principal é a preparação para o reuso. Engenharia de domínio é realizada apenas uma vez para a linha de produtos inteira. Em contraste, a **engenharia de aplicação** tem o objetivo de desenvolver um produto específico para as necessidades particulares de um cliente. Ela corresponde ao processo de desenvolvimento de aplicações individuais na engenharia de software tradicional, mas reusa artefatos definidos na engenharia de domínio quando e onde for possível. Busca desenvolver aplicações com o máximo de reuso possível. A engenharia de aplicação é repetida para cada produto que é derivado

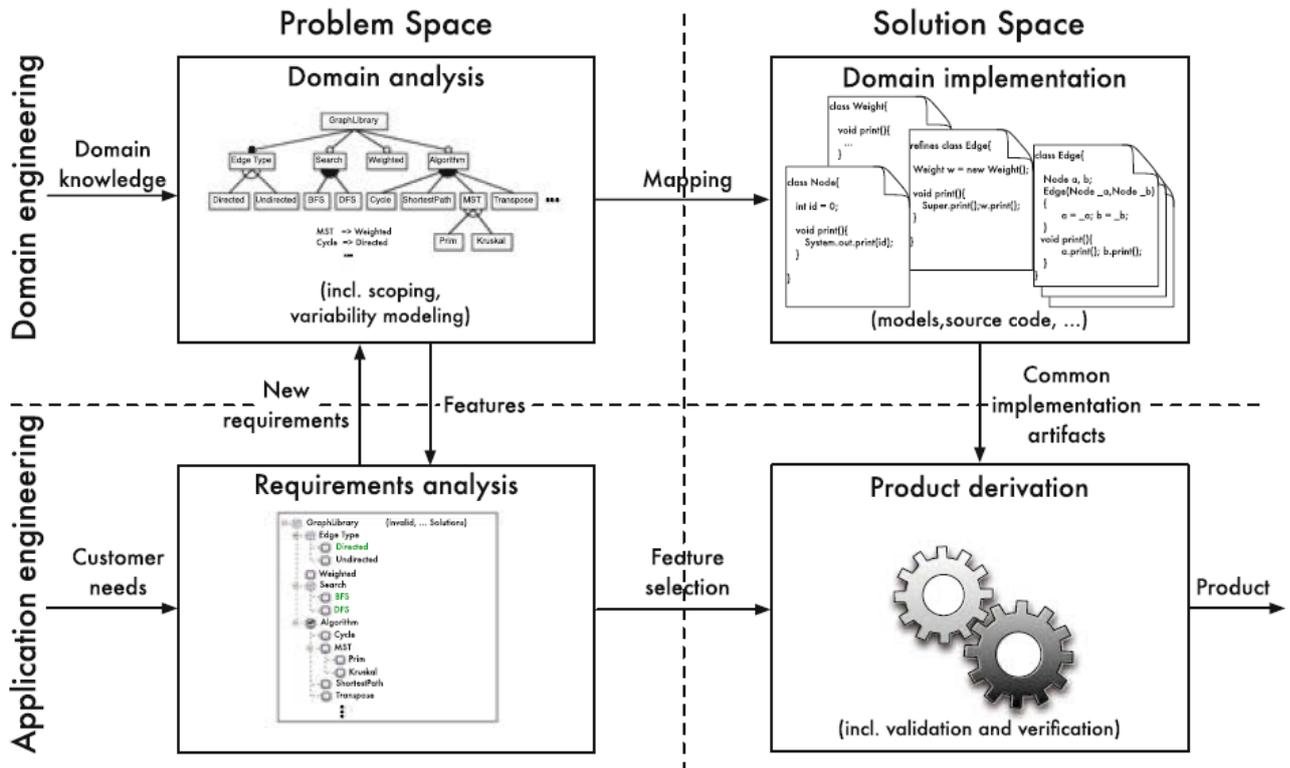


Figura 4.3: Processo de Desenvolvimento de Linhas de Produto de Software

em uma linha.

A distinção entre o espaço do problema e o espaço da solução destaca duas diferentes perspectivas. O **espaço do problema** assume a perspectiva dos interessados no produto e seus problemas, requisitos e visões sobre o domínio como um todo e sobre os produtos individuais. *Features* são, na verdade, abstrações do domínio que caracterizam o espaço do problema. Por outro lado, o **espaço da solução** representa as perspectivas dos desenvolvedores e vendedores dos produtos. Se caracteriza pela terminologia de desenvolvedor que inclui nomes de funções, classes e parâmetros de programa. O espaço de solução cobre o design, implementação, verificação e validação de *features* e suas combinações de modo coerente visando facilitar o reuso sistemático.

4.6.2 Tarefas do Desenvolvimento de LPS

As distinções entre engenharia de domínio e de aplicação, bem como a separação entre espaço de problema e de solução dão origem a quatro grupos de tarefas no desenvolvimento de LPS's.

A **análise de domínio** é uma forma de engenharia de requisitos para uma linha de produtos inteira. Nesta fase, é necessário decidir o escopo do domínio, isto é, decidir quais produtos devem ser cobertos pela linha de produtos e, conseqüentemente, quais *features* são relevantes e devem ser implementadas como artefatos reusáveis (modelagem do domínio) e quais devem ser consideradas como variantes do sistema. Os resultados da análise de domínio são normalmente registrados em um documento chamado modelo de *features*.

A **análise de requisitos** investiga as necessidades de um cliente específico como parte da engenharia de aplicação. No caso mais simples, os requisitos de um cliente são mapeados para uma seleção de *features* baseada nas features identificadas durante a análise de domínio. Se novos requisitos são descobertos, eles podem ser introduzidos nos resultados da análise de domínio, o que pode

resultar em mudanças no modelo de *features* e no domínio de artefatos reusáveis.

A **implementação do domínio** é o processo de desenvolver artefatos reusáveis que correspondem às *features* identificadas na análise de domínio, sejam eles, *design*, teste, documentação ou código-fonte, por exemplo. Pensando apenas no código-fonte, primeiro seria necessário selecionar uma estratégia de implementação ou estrutura de reuso (*reuse framework*). Poderíamos usar diretivas de pré-processador ou criar uma estrutura de *plugins* que pudessem ser combinados sob demanda. Discutiremos a estratégia de implementação em detalhes na seção 4.7

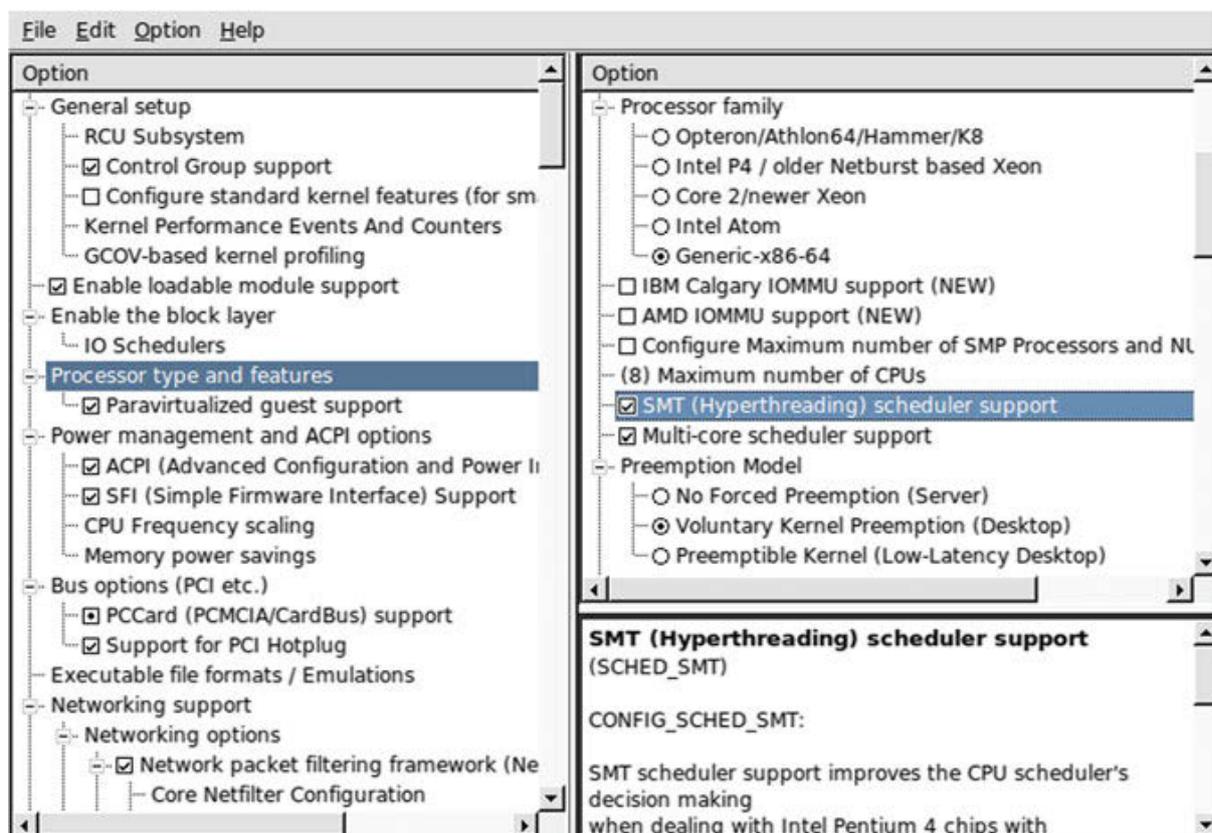


Figura 4.4: Exemplo: seleção de *features* do Kernel do Linux

A **geração de produto** é o passo de produção da engenharia de aplicação no qual artefatos reusáveis são combinados de acordo com os resultados da análise de requisitos. Dependendo da abordagem de implementação este processo pode ser mais ou menos automatizado, possivelmente envolvendo várias tarefas de desenvolvimento e personalização. O principal objetivo de uma LPS baseada em *features* é automatizar, completamente, a geração de produtos. A derivação automática de produtos ocorre quando um produto, especificado por uma seleção de *features* (Figura 4.4), é montado automaticamente a partir de artefatos reusáveis, ou seja, após o usuário fazer a seleção das *features* desejadas, a LPS executa algum mecanismo de composição para combiná-las criando um produto final (Figura 4.5), sem a necessidade de novas intervenções manuais.

4.6.3 Escalando uma LPS

Em uma situação ideal a análise de requisitos para a criação de um novo produto deveria se reduzir a uma seleção de *features* existentes e tal produto poderia ser montado com implementações de artefatos reusáveis associados a estas *features*. Eventualmente, no entanto, pode ocorrer que novos clientes venham a solicitar produtos que contenham *features* que não foram previstas na análise de

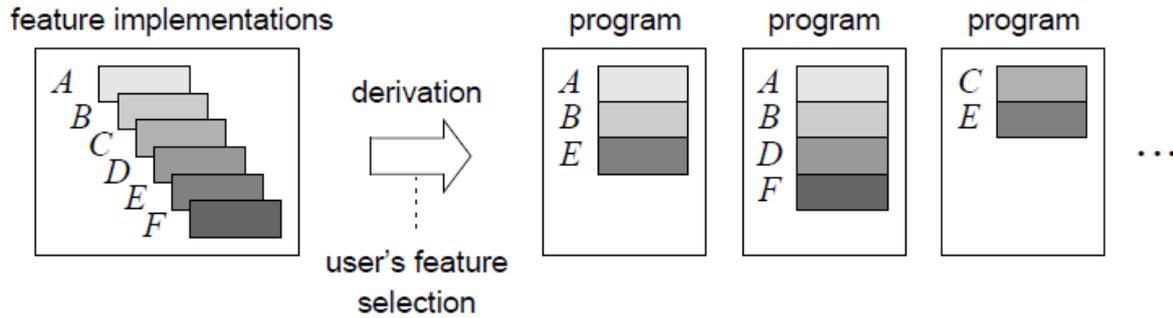


Figura 4.5: *Geração de Produtos em uma Linha de Produtos de Software*

domínio. Neste caso, são três as possíveis estratégias:

- Considerar o requisito fora do escopo da LPS e, simplesmente, deixar de atender a solicitação de produto;
- Criar a maior parte do produto com os artefatos reusáveis existentes e complementar este produto com extensões personalizadas. Neste caso será investido um esforço adicional de implementação durante a fase de engenharia de aplicação que não será integrado à linha de produto;
- Modificar o escopo da LPS e incluir o requisito adicional de modo que uma nova feature ou modificações nas *features* existentes. Neste caso, voltamos à fase de engenharia de domínio e fazemos as alterações necessárias em artefatos existentes ou criamos artefatos complementares. As novas *features*, neste caso, passam a estar disponíveis, e portanto podem ser integradas, a produtos pré-existentes da família.

4.7 Implementando uma LPS

Capítulo 5

Instanciando o iPlan

Capítulo 6

Experimentos com o uso do iPlan

Apêndice A

Protocolo da Revisão Sistemática

Apêndice B

Questionários Utilizados nos Experimentos

Apêndice C

Avaliação de software para Learning Design

C.1 Introdução

Durante a pesquisa para o desenvolvimento desta tese foi necessário fazer um levantamento das ferramentas existentes que se propunham a ajudar o professor no trabalho de preparar suas aulas. Após este levantamento, precisamos verificar e, eventualmente, comparar as funcionalidades oferecidas. Este trabalho de comparação foi realizado com base nas ideias do *framework* apresentado abaixo.

C.2 Modelo usado nas avaliações

A fim de analisar ferramentas criadas para apoiar os processos previstos na especificação de Learning Design, Britain (2003) criou um conjunto de questões cujo objetivo é sistematizar a avaliação do software. As tabelas abaixo apresentam estas questões e seus objetivos. As questões são divididas em três grupos:

C.2.1 Questões sobre a finalidade do software

C.2.2 Questões sobre projeto e funcionalidades do software

C.2.3 Questões sobre aspectos técnicos do software

| <i>Finalidade do Software</i> | |
|--|---|
| 1. Descrição das Finalidades | |
| | Uma breve introdução sobre a finalidade do software. Esta informação pode, geralmente ser obtida no site do projeto. |
| 2. Pra quem o sistema foi criado? Quem mais está envolvido? | |
| | Algumas ferramentas são criadas para usuários finais, isto é, professores e estudantes, outras são criadas para o uso de desenvolvedores de software ou instrucional designers com um alto nível de conhecimento técnico. É importante estar consciente desta diferença. |
| 3. Perspectiva ou ponto de vista dos projetistas | |
| | É útil conhecer a formação dos desenvolvedores do software para obter uma compreensão aprofundada sobre a orientação pedagógica do sistema. Se eles têm um viés comercial ou de treinamento sua ferramenta pode ter uma ênfase em aspectos diferentes daquelas projetadas especificamente para a educação básica ou superior. |
| 4. Escopo | |
| | Algumas ferramentas são ambientes de autoria ou editores, outras são ambientes de execução ou tocadores (players). Algumas têm ambas as funcionalidades. Algumas ferramentas são voltadas para o trabalho individualizado, outras possibilitam o trabalho em grupo. |
| 5. Integração | |
| | Algumas ferramentas são projetadas para serem usadas de modo independente de outras aplicações; outras são projetadas para funcionar como parte de um conjunto mais amplo de ferramentas. O software pode ser integrado a outras ferramentas? |
| 6. O software implementa IMS-LD? (se sim, em que nível: A, B or C?) | |
| | Se o software não implementa IMS-LD, tem a intenção de fazê-lo no futuro? |

Tabela C.1: Questões sobre a finalidade do software

| <i>Especificidades do Projeto</i> | |
|---|---|
| 1. Quais são os principais conceitos ou entidades modelados no software? | |
| | Uma grande quantidade de informações pode ser obtida da análise do modelo de negócio do software. Se este modelo não estiver explícito, é útil observar quais são suas principais entidades, objetos e conceitos. Por exemplo, para learning design queremos saber se o sistema foi construído em torno do modelo de objetos ?Atividade? e ?Fluxo de Trabalho? ou se foi construído em torno do modelo de objetos ?Conteúdo? ou ?Objetos de Aprendizagem?. Naturalmente, estas são apenas diretrizes e uma análise mais aprofundada pode ser necessária visto que desenvolvedores de software podem rotular uma entidade de ?Atividade?, com operações como ?criar nova atividade? e ?salvar atividade? quando, de fato, esta atividade é meramente um bloco de conteúdo. |
| 2. Qual é o modelo de ?Atividade? implementado no software? | |
| | Para Learning Design o conceito de ?Atividade? é um dos mais importantes, logo pode ser útil saber mais sobre o que os desenvolvedores consideraram como atividade na implementação do software. |
| 3. Qual é o modelo de Workflow implementado no software? | |
| | O conceito de workflow também é muito importante. Logo, também é importante conhecer seus detalhes de implementação. Apenas sequenciamento simples é permitido ou workflows mais complexos são possíveis? |
| 4. Quais são as características da interface gráfica software? | |
| | Um aspecto chave da usabilidade do software por diferentes grupos é a natureza da sua interface gráfica. |
| 5. Para ambientes de execução ? quão interativo é o design quando em execução? | |
| | O quê os estudantes realmente podem fazer neste ambiente? O quê professores podem fazer? |

Tabela C.2: *Questões sobre projeto e funcionalidades do software*

| <i>Características Técnicas</i> | |
|--|---|
| 1. Qual é o formato do software (ambiente online, aplicativo para desktop, etc...)? | |
| | Que tipo de software é este? Ele pode ser executado em diferentes plataformas? |
| 2. Quais são os requisitos técnicos para a execução do software? | |
| | Listar qualquer outro requisito técnico ou software adicional requerido para a execução da aplicação. |

Tabela C.3: *Questões sobre aspectos técnicos do software*

Referências Bibliográficas

- ABED (2012)** ABED. Censo ead brasil 2012. http://www.abed.org.br/censoead/censoEAD.BR_2012_pt.pdf, 2012. Último acesso em 27/08/2015. Citado na pág. 1
- ABED (2013)** ABED. Censo ead brasil 2013. http://www.abed.org.br/censoead2013/CENSO_EAD_2013_PORTUGUES.pdf, 2013. Último acesso em 27/08/2015. Citado na pág. 5
- Alexander et al. (1977)** Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King e Shlomo Angel. *A pattern language: town, buildings, constructions*. Oxford University Press. Último acesso em 27/08/2015. Citado na pág. 22, 23
- Alsofyani e bin Aris (2011)** Mohammed M. Alsofyani e Baharuddin bin Aris. Design and development of tp ack template - planning effective blended courses. Em *IEEE 2011 International Conference on Information Society (i-Society)*, London, EN. Citado na pág. 8, 11
- Apel et al. (2013)** Sven Apel, Don Batory, Christian Kastner e Gunter Saake. *Feature-Oriented Software Product Lines*. Springer-Verlag Berlin Heidelberg. Citado na pág. 26, 27, 28
- ASK-LDT (2005)** ASK-LDT. Ask learning designer toolkit. [http://edutechwiki.unige.ch/en/ASK_Learning_Designer_Toolkit_\(ASK-LDT\)](http://edutechwiki.unige.ch/en/ASK_Learning_Designer_Toolkit_(ASK-LDT)), 2005. Último acesso em 27/08/2015. Citado na pág. 18
- Baehr (2012)** Craig Baehr. Incorporating user appropriation, media richness, and collaborative knowledge sharing into blended e-learning training tutorial. *IEEE TRANSACTIONS ON PROFESSIONAL COMMUNICATION*, 55(2):175–184. Citado na pág. 5, 6, 10
- Britain (2003)** Sandy Britain. A review of learning design: Concept, specifications and tools - a report for the jisc e-learning pedagogy programme. Relatório técnico, JISC Joint Information Systems Committee. Citado na pág. 14, 18, 43
- Cadmos (2011)** Cadmos. The learning design grid. <http://www.ld-grid.org/resources/tools/cadmos>, 2011. Último acesso em 27/08/2015. Citado na pág. 2
- Carle et al. (2007)** Andy Carle, Michael Clancy e John Canny. Working with pedagogical patterns in pact: Initial applications and observations. Em *SIGCSE'07 Technical Symposium on Computer Science Education*. Citado na pág. 21, 22, 23
- Cheon et al. (2002)** Jong-Pil Cheon, Jang-Mi Paek, Sun-Gwan Han e Chul-Hwan Lee. Automated lesson planner system for ict education. Em *Proceedings of the International Conference on Computers in Education (ICCE'02)*. Citado na pág. 7, 10
- Chrome (2015)** Google Chrome. Navegador google chrome. <https://www.google.com.br/chrome/browser/desktop/>, 2015. Último acesso em 27/08/2015. Citado na pág. 10
- Clayer et al. (2013)** Jean-Pierre Clayer, Claudine Toffolon e Christophe Choquet. Patterns, pedagogical design schemes and process for instructional design. Em *IEEE 13th International Conference on Advanced Learning Technologies*. Citado na pág. 9, 22

- Collage (1990)** Collage. Collaborative learning design editor. <http://www.w3.org/XML>, 1990. Último acesso em 27/08/2015. Citado na pág. 18
- Coopercore (2003)** Coopercore. Coopercore the ims learning design engine. <http://coppercore.sourceforge.net>, 2003. Último acesso em 27/08/2015. Citado na pág. 2, 18
- de Vord e Pogue (2011)** Rebecca V. de Vord e Korolyn Pogue. Teaching time investment: Does online really take more time than face-to-face? *IRRODL The International Review of Research in Open and Distance Learning*. Citado na pág. 5, 6, 8, 11
- Derntl e Calvo (2011)** Michael Derntl e Rafael A. Calvo. E-learning frameworks: facilitating the implementation of educational design patterns. *International Journal of Technology Enhanced Learning*, 3(3):284–296. Citado na pág. 5
- Derntl et al. (2010)** Michael Derntl, Susanne Neumann, Dai Griffiths e Petra Oberhuemer. Investigating teachers' understanding of ims learning design: Yes they can! *Sustaining TEL: From Innovation to Learning and Practice*, 6383:62–77. Citado na pág. 8, 18
- Durand e Downes (2009)** Guillaume Durand e Stephen Downes. Toward simple learning design 2.0 - simple interoperability for learning activities. Em *Proceedings of 2009 4th International Conference on Computer Science and Education*. Citado na pág. 1, 2, 13
- Durand et al. (2010)** Guillaume Durand, Luc Belliveau e Benjamin Craig. Simple learning design 2.0. Em *10th IEEE International Conference on Advanced Learning Technologies*. Citado na pág. 13
- Dutra et al. (2005)** Renato L. S. Dutra, Liane Margarida Rockenbach Tarouco e Mary Lucia Pedroso Konrath. Learning design, evoluindo de objetos de aprendizagem para atividades de aprendizagem. *Novas Tecnologias na Educação - RENOUE*, 3(1). Citado na pág. 18
- Eckstein et al. (2001)** Jutta Eckstein, Mary Lynn Manns e Markus Voelter. Pedagogical patterns: Capturing best practices in teaching object technology. *Software Focus*, 2(1):9–12. Citado na pág. 21, 22, 23
- Explorer (2015)** Internet Explorer. Navegador internet explorerer. <http://windows.microsoft.com/pt-br/internet-explorer/ie-9-worldwide-languages>, 2015. Último acesso em 27/08/2015. Citado na pág. 10
- Filho e Derycke (2009)** César O. Moura Filho e Alain Derycke. Pedagogical patterns and learning design: When two worlds cooperate. Em *8th International Conference on Information Technology Based Higher Education and Training (ITHET)*. Citado na pág. 22, 23
- Firefox (2003)** Mozilla Firefox. Navegador mozilla firefox. <https://www.mozilla.org/pt-BR/firefox/new/>, 2003. Último acesso em 27/08/2015. Citado na pág. 10
- Gamma et al. (2000)** Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides. *Padrões de Projeto - Soluções Reutilizáveis de Software Orientado a Objeto*. Bookman. Citado na pág. 23
- Gibson et al. (1998)** Jane W. Gibson, Charles W. Blackwell e Richard M. Hodgetts. Communicating with online students: Lessons from the front. *A Contemporary Renaissance: Changing the Way We Communicate*. Citado na pág. 6, 7, 8
- Herman e Banister (2007)** Terry Herman e Savilla Banister. Face-to-face versus online coursework: A comparison of costs and learning outcomes. *Contemporary Issues in Technology and Teacher Education*, 7(4):318–326. Citado na pág. 5, 7, 8
- Hermans et al. (2003)** Henry Hermans, Jocelyn Manderveld e Hubert Vogten. Educational modelling language. Relatório técnico. Citado na pág. 12

- Hernandez-Leo (2007)** Davinia Hernandez-Leo. *A pattern-based design process for the creation of CSCL macro-scripts computationally represented with IMS LD*. Tese de Doutorado, Departament de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu Fabra, Barcelona. Citado na pág. 3
- IBM (2004)** IBM. Patterns for e-business. <http://www.redbooks.ibm.com/redpapers/pdfs/redp3836.pdf>, 2004. Último acesso em 27/08/2015. Citado na pág. 23
- IDOrg (2013)** IDOrg. Instructional design. <http://www.instructionaldesign.org/index.html>, 2013. Último acesso em 27/08/2015. Citado na pág. 1
- IMS-BP (2003)** IMS-BP. Ims learning design best practice and implementation guide. http://www.imsglobal.org/learningdesign/ldv1p0/imslld_bestv1p0.html, 2003. Último acesso em 27/08/2015. Citado na pág. 2, 9
- IMS-LD (2003)** IMS-LD. Learning design best practice and implementation guide; ims learning design information model. <http://www.imsglobal.org/learningdesign>, 2003. Último acesso em 27/08/2015. Citado na pág. 1, 13
- IMS-SS (2001)** IMS-SS. Ims simple sequencing information model, best practice and information guide, version 1.0 final specification. <http://www.imsglobal.org/simplesequencing>, 2001. Último acesso em 27/08/2015. Citado na pág. 14
- Jaffry (2003)** Syed W. Q. Jaffry. Pedagogical pattern language for in time student confidence in studied material. Em *9th IEEE International Multitopic Conference, (INMIC)*. Citado na pág. 21, 22
- Jones et al. (2000)** David Jones, Sharonn Stewart e Power L. Patterns: Using proven experience to develop online learning. Em *Ascilite 99 The 16th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*. Citado na pág. 9, 21, 22
- Koper e Manderveld (2004)** Rob Koper e Jocelyn Manderveld. Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology*, 35(5):537–551. Citado na pág. 14
- Lams (2003)** Lams. The learning activity management system (lams). <http://www.lamsinternational.com>, 2003. Último acesso em 27/08/2015. Citado na pág. 2, 18
- Medeiros et al. (2007)** Flavio M. Medeiros, Arturo Hernández-Domínguez, Fábio N. de Medeiros e Anderson G. da Silva. Um sistema de ensino na web baseado no padrão pedagógico exposição teórica-exemplos-atividade-apresentação-avaliação. Em *XVIII Simpósio Brasileiro de Informática da Educação - SBIE - Mackenzie*. Citado na pág. 21, 22, 23
- Mylonakis et al. (2013)** Manolis Mylonakis, Polyxeni Arapi, Nektarios Moumoutzis, Stavros Christodoulakis e Maria Ampartzaki. Octopus: A collaborative environment supporting the development of effective instructional design. Em *Second International Conference on e-Learning and e-Technologies in Education (ICEEE)*. Citado na pág. 23
- Nascimento et al. (2013)** Maurício Garcia Franco Nascimento, Leônidas de Oliveira Brandão e Anarosa Alves Franco Brandão. A model to support a learning object repository for web-based courses. Em *43rd ASEE/IEEE Frontiers in Education Conference*. Citado na pág. 4
- PPP (2000)** PPP. Pedagogical patterns project. <http://www.pedagogicalpatterns.org>, 2000. Último acesso em 27/08/2015. Citado na pág. 1, 24
- Pressman (1995)** Roger S. Pressman. *Engenharia de Software*. Pearson Education. Citado na pág. 25

- Rawlings et al. (2002)** A. Rawlings, van Rosmalen P., Koper R., Rodriguez-Artacho M. e Lefrere P. Survey of educational modeling languages. Em *CEN/ISSS WS/LT Learning Technologies Workshop*. Citado na pág. 1
- Reload (2003)** Reload. Extensible markup language. <http://www.w3.org/XML>, 2003. Último acesso em 27/08/2015. Citado na pág. 2, 18
- Rodrigues e Brandão (2010)** Patrícia A. Rodrigues e Leônidas O. Brandão. i-tarefa: incrementando a interatividade em cursos via web com o moodle. Em *Anais do XVI Workshop de Informática na Escola ? WIE*. Citado na pág. 4
- Siviter e Wheeler (2004)** Douglas Siviter e Andrew Wheeler. Patterns in education systems: Are there lessons from the enterprise domain? Em *ASCILITE Conference in Perth, Western Australia*. Citado na pág. 22
- SLeD (2004)** SLeD. Service-based learning design. <http://sourceforge.net/projects/ldplayer/>, 2004. Último acesso em 27/08/2015. Citado na pág. 18
- Stark et al. (2002)** Carolyn Stark, Kathy J. Schmidt, Linda Shafe e Mary Crawford. Creating e-learning programs: A comparison of two programs. Em *32nd ASEE/IEEE Frontiers in Education Conference*. Citado na pág. 1, 5, 6, 7, 8, 11
- Turani (2007)** Aiman Turani. *A Pedagogical Application Framework for Synchronous Collaboration*. Tese de Doutorado, School of Electrical and Information Engineering, The University of Sydney, Australia. Citado na pág. 10, 22
- UML (1995)** UML. Unified modeling language. <http://www.uml.org>, 1995. Último acesso em 27/08/2015. Citado na pág. 2
- Valentín et al. (2007)** Luis F. Valentín, Abelardo Pardo Sánchez e Carlos Delgado Kloos. Using learning design to deploy and administer engineering courses. Em *37th ASEE/IEEE Frontiers in Education Conference*, October 10 2007, Milwaukee, WI. Citado na pág. 2, 18
- Verpoorten et al. (2006)** Dominique Verpoorten, Marianne Poumay, Séverine Delcomminette e Dieudonné Leclercq. From expository teaching to first e-learning course production: Capture in a 17 online course sample of a pedagogical pattern facilitating transition. Em *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06)*. Citado na pág. 9, 21, 22, 24
- Word (2015)** Microsoft Word. Editor de texto. <https://www.microsoft.com/pt-br/download/office.aspx>, 2015. Último acesso em 27/08/2015. Citado na pág. 11
- Worley e Tesdell (2009)** Wanda L. Worley e Lee S. Tesdell. Instructor time and effort in online and face-to-face teaching: Lessons learned. *IEEE TRANSACTIONS ON PROFESSIONAL COMMUNICATION*, 52(2):138–150. Citado na pág. 6, 7, 8
- XML (1990)** XML. Extensible markup language. <http://www.w3.org/XML>, 1990. Último acesso em 27/08/2015. Citado na pág. 2
- Zhi-ying e Hong (2010)** Gao Zhi-ying e Liu Hong. Effective english teaching and learning via web-based electronic english lesson plan design. *2010 Second International Workshop on Education Technology and Computer Science*, páginas 358–361. Citado na pág. 6, 10