



Universidade Federal do Rio de Janeiro

Desenvolvimento de Simulações Integrando Metodologias Ágeis de ES e Conceitos de IHC para o treinamento em Sistemas Complexos

DISSERTAÇÃO DE MESTRADO

Fábio Roberto Lapolli
lapollimaster@yahoo.com.br

Orientador(es): Claudia Lage Rebello da Motta
José Orlando Gomes





Desenvolvimento de Simulações Integrando Metodologias Ágeis de ES e Conceitos de IHC para o treinamento em Sistemas Complexos

Fábio Roberto Lapolli
lapollimaster@yahoo.com.br

Orientador(es): Claudia Lage Rebello da Motta
José Orlando Gomes

Área de Pesquisa:
Informática Educação e Sociedade

Rio de Janeiro
Fevereiro 2011

C957 Lapolli, Fábio Roberto.

Desenvolvimento de Simulações Integrando Metodologias Ágeis de ES e Conceitos de IHC para o Treinamento em Sistemas Complexos/ Fábio Roberto Lapolli. – Rio de Janeiro, 2011. 187 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Núcleo de Computação Eletrônica, Programa de Pós-graduação em Informática 2011.

Orientadores: Claudia Lage Rebello da Motta; José Orlando Gomes.

1. Sistemas Complexos – Teses. 2. Simulações – Teses. 3. IHC – Teses. 4. Metodologias Ágeis – Teses. I. Claudia Lage Rebello da Motta (Orient.). II. José Orlando Gomes (Orient.). III. Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica. IV. Título.

CDD. .

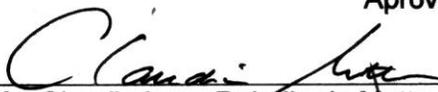
Fábio Roberto Lapolli

Desenvolvimento de Simulações Integrando Metodologias Ágeis de ES Conceitos de IHC para o treinamento em Sistemas Complexos

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do grau de Mestre em Informática.

Rio de Janeiro, 21 de fevereiro de 2011.

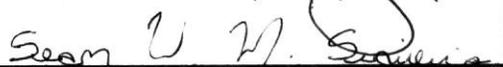
Aprovada por:



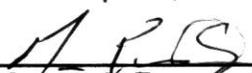
Profa. Claudia Lage Rebello da Motta, D.Sc., PPGI e iNCE/UFRJ (Orientadora)



Prof. José Orlando Gemes, D.Sc., PPGI (Orientador)



Prof. Sean Siqueira, D.Sc., PPGI/UNIRIO



Prof. Marcos da Fonseca Elia, Ph.D., PPGI e iNCE/UFRJ



Carlo Emmanuel Tolla de Oliveira, Ph.D., e iNCE/UFRJ

Dedicatória

*Ofereço este trabalho aos meus pais
Fernando e Tânia e aos meus irmãos
Fernando e Bruno, que sempre me
apoiaram e compartilharam comigo todos
os momentos de minha trajetória, pessoal
profissional e acadêmica.*

Agradecimentos

A todos aqueles que acreditaram e me apoiaram no desenvolvimento deste trabalho e contribuíram direta e indiretamente para o meu aprendizado acadêmico, profissional e pessoal, quero deixar aqui os meus sinceros agradecimentos.

A Divisão de Ensino do Instituto de Controle do Espaço Aéreo, pela atenção dispensada durante minhas visitas com a finalidade de entender o Sistema de Controle do Espaço Aéreo Brasileiro, bem como o estudo de campo.

Aos professores Claudia Motta e José Orlando Gomes, que acreditaram na minha proposta de dissertação e aceitaram o desafio da orientação. O apoio e incentivo constantes foram determinantes para o desenvolvimento e conclusão desse trabalho.

Aos professores Marcos Elia e Carlo Emmanoel Tolla de Oliveira pelas valorosas contribuições com o meu trabalho e que muito me honraram em ter aceitado o convite para serem membros na Banca.

A todos os professores do PPGI que em momentos distintos foram fundamentais para minha formação.

A professora Maria Teresa Gouvea, por me apresentar ao grupo de pesquisa do NCE.

Aos amigos do corpo discente que fizeram parte da minha trajetória. Em especial: Cristiane de Moura Cruz e Luiz Francisco Dias Pereira.

Ao amigo Gilson Fonseca, que teve uma participação fundamental na viabilização do contato com o ICEA. Seu incentivo e contribuição com as pesquisas foram determinantes para a realização deste trabalho.

Aos instrutores e alunos do curso de formação de controlador de tráfego aéreo do ICEA, que aceitaram participar do estudo de campo realizado nesse trabalho. Suas participações e sugestões foram indispensáveis para realização dessa dissertação.

Resumo

LAPOLLI, Fábio Roberto. **Desenvolvimento de Simulações Integrando Metodologias Ágeis de ES e Conceitos de IHC para o treinamento em Sistemas Complexos**. Rio de Janeiro, 2011. Dissertação (Mestrado em Informática) – Programa de Pós-Graduação em Informática, Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

Na área de estratégias educacionais, os Objetos de Aprendizagem vêm se destacando pela maneira como influenciam significativamente a forma como as pessoas aprendem ao possibilitar a representação prática da aplicação do conteúdo teórico. O desenvolvimento de objetos de aprendizagem exige modelos instrucionais e de desenvolvimento flexíveis, por se tratar de uma categoria de software com um conjunto de requisitos bastante específicos e demandas tecnológicas singulares. Neste trabalho é proposto um modelo integrado de desenvolvimento apoiado no uso de metodologias ágeis de engenharia de software aliado a conceitos de interface humano-computador para o desenvolvimento de objetos de aprendizagem baseados em simulações. O modelo proposto aborda as principais fases do projeto de desenvolvimento de software educacional, sendo, o levantamento de requisitos, o projeto instrucional, o projeto de interface e a metodologia de desenvolvimento. A integração entre metodologias ágeis de ES e conceitos de IHC pode facilitar esse processo permitindo a modelagem das funcionalidades baseadas nos requisitos de comportamento do aluno. Para a fase de levantamento de requisitos é aplicada uma adaptação do método GOMS a fim de estabelecer, metas, objetivos, métodos e regras de seleção na execução de uma tarefa ou atividade. Para o projeto pedagógico, em alinhamento com o método de levantamento de requisitos é proposta uma adaptação das práticas ágeis de desenvolvimento no modelo de projeto instrucional, possibilitando a identificação dos cenários que permitam ao aluno interiorizar o conceito de tal forma que seja possível a sua abstração e aplicação em diferentes situações, inclusive as não previstas durante o seu treinamento. No projeto interface são apresentadas diretrizes para a construção de scaffoldings para apoiar o processo de aprendizagem, através do design de interação. E finalmente, é apresentada e aplicada uma adaptação de Extreme Programming (XP) para o desenvolvimento dos objetos de aprendizagem. Neste trabalho, dois Objetos de Aprendizagem foram desenvolvidos a fim de verificar a viabilidade da solução proposta. Uma primeira avaliação procurando verificar a validação da proposta da dissertação mostra indícios de que a hipótese formulada nessa dissertação pode ser comprovada. Pelo que tudo indica a utilização de Objetos de Aprendizagem poderia apoiar a compreensão e fixação de conceitos relacionados ao controle de tráfego aéreo mostrando indícios da eficácia do modelo de objeto de aprendizagem proposto. Em segundo momento foi realizada uma avaliação procurando verificar a viabilidade do desenvolvimento desse modelo através da metodologia integrada de desenvolvimento proposta. Houve indícios também da eficácia da metodologia proposta, porém com algumas considerações com relação ao conteúdo abordado e a necessária consultoria de especialistas ao longo do processo de desenvolvimento.

Abstract

LAPOLLI, Fábio Roberto. **Development of Simulations Integrating Agile Methodologies and Concepts IHC for training in Complex Systems**. Rio de Janeiro, 2011. Dissertation (Master Degree in Computer Science) - Graduate Program in Computer Science, Institute of Mathematics, Electronic Computer Center, Federal University of Rio de Janeiro, Rio de Janeiro, 2011.

In the area of educational strategies, Learning Objects are distinguishing themselves by how significantly influence how people learn to allow the representation of the practical application of theoretical content. The development of learning objects requires instructional models and flexible development, because it is a category of software with a very specific set of requirements and technological demands of individuals. This paper proposes an integrated model of development based on the use of agile software engineering combined with concepts of human-computer interface for the development of learning objects based on simulations. The proposed model addresses the key phases of project development of educational software, and the requirements gathering, instructional design, interface design and development methodology. The integration between Agile SE and HCI concepts can facilitate this process by allowing the modeling of features based on the performance requirements of the student. For the phase of requirements gathering is an adaptation of the GOMS method in order to establish goals, objectives, methods and selection rules in performing a task or activity. For the education program, in alignment with the requirements elicitation method is proposed an adaptation of agile development practices in instructional design model, allowing the identification of scenarios that enable students to internalize the concept so that it is possible its abstraction and application in different situations, including those not covered during your training. In interface design guidelines are presented for the construction of scaffoldings to support the process of learning through interaction design. And finally, is presented and applied an adaptation of Extreme Programming (XP) for development of learning objects. In this work, two Learning Objects were developed to verify the feasibility of the proposed solution. A first assessment attempt to determine the validation of the proposed work shows evidence that the hypothesis formulated in this thesis can be proven. By all indications the use of learning objects could support the understanding and retention of concepts related to air traffic control signs showing the effectiveness of the learning object model proposed. In second place was looking for an assessment to determine the viability of this model through an integrated methodology proposed development. There is also evidence of the effectiveness of the proposed methodology, however, some considerations regarding the content addressed and the necessary advice from experts throughout the development process.

Lista de Figuras

Figura 1: Assimetria entre IHC e ES no ciclo de desenvolvimento de software.....	71
Figura 2: Processos de IHC e ES para o desenvolvimento de software	72
Figura 3: Pontos com e sem integração entre as áreas de IHC e ES	72
Figura 4 - Desafios e abordagens do uso de cenários	75
Figura 5 - Requisitos iniciais para o desenvolvimento do software.....	98
Figura 6 - Ciclo Inicial de Produção do software	101
Figura 7 - Ciclo de construção e teste de pequenas versões	104
Figura 8 - Ciclo de desenvolvimento contínuo.....	104
Figura 9 - Exemplo de <i>Story Card</i> usado no desenvolvimento do OA (Lapolli, Cruz <i>et al.</i> 2009 b).....	105
Figura 10 - Exemplo de história de usuário (Lapolli, Cruz <i>et al.</i> 2009 b).....	107
Figura 11 - Modelo de projeto instrucional dirigido pelo comportamento (Lapolli e Cruz <i>et al.</i> 2009 b)	108
Figura 12 - Produtos da etapa proposta de atividades	109
Figura 13 - Produtos da etapa definição dos cenários	109
Figura 14 - Proposta de atividade do aplicativo.....	111
Figura 15 - Modelo de objeto de aprendizagem apoiado em scaffoldings (Lapolli 2009, 2010 e).....	121
Figura 16 - Estruturação dos conhecimentos envolvidos na altimetria (Lapolli 2009).....	123
Figura 17 - Tela principal do ClearaNCE	124
Figura 18 - A estrutura interna do Altímetro	125
Figura 19 - Tela do aplicativo “A Terra no Espaço” com o subtema Orbita Elíptica	135
Figura 20 - Tela do aplicativo “A Terra no Espaço” com o subtema Orbita Elíptica	136
Figura 21 - Tela do aplicativo “A Terra no Espaço” com o subtema Inclinação do Eixo de Rotação	136
Figura 22 - Tela do aplicativo “A Terra no Espaço” com o subtema Inclinação do Eixo de Rotação	137

Lista de Quadros

Quadro 1 - Fatores cognitivos determinantes da formação da ação.....	27
Quadro 2 - Vantagens, Desvantagens, Princípios de Uso e Relações da aplicação do DCU	61
Quadro 3 - Vantagens, Desvantagens, Princípios de Uso e Relações da aplicação do DCT.....	62
Quadro 4 - Vantagens, Desvantagens, Princípios de Uso e Relações da aplicação do DCA	65
Quadro 5 - Questões e formas de integração entre IHC e ES.	74
Quadro 6 - Comparação entre metodologias de IHC.....	82
Quadro 7 - Similaridades entre XP e IHC.....	83
Quadro 8 - Similaridades entre MA e IHC.....	83
Quadro 9 - Pontos em comum entre MA, XP com IHC e princípios e práticas de design de interação.	84
Quadro 10 - Métodos de ER (Engenharia de Requisitos) classificados como ágeis.....	99

Lista de Tabelas

Tabela 1 - Matriz de correlação entre os respondentes	148
Tabela 2 - Valor registrado do Alpha de Cronbach entre os 9 respondentes em análise.....	148
Tabela 3a - Média estatística e desvio padrão dos critérios pedagógicos.....	149
Tabela 3b - Média estatística e desvio padrão dos critérios ergonômicos.....	150
Tabela 4 - Distribuição das medias das respostas dos respondentes nos 24 itens.....	151
Tabela 4 - Matriz de correlação entre critérios pedagógicos e ergonômicos restantes análise.....	152
Tabela 5 - Matriz dos Componentes Rotacionados pelo Método Varimax	153

Lista de Siglas

BDD Behaviour-Driven Development
CIP Ciclo Inicial de Produção
CTP Construção e Teste do Protótipo
DIU Design de Interface do Usuário
DIU Interface do Usuário
DCU Design Centrado no Usuário
DCT Design Centrado na Tarefa
DCA Design Centrado na Atividade
EEA Escola de Especialistas da Aeronáutica
ER Engenharia de Requisitos
ES Engenharia de Software
FDD Feature Driven Development
GOMS Goals, Operators, Methods, and Selection Rules
IHC Interface Humano Computador
ICEA Instituto de Controle do Espaço Aéreo
MA Metodologia Ágil
OA Objeto de Aprendizagem
OACI Organização de Aviação Civil Internacional
PUD Plano de Unidade Didática
QFE Pressão Atmosférica à Elevação do Aeródromo
QNE Altitude corrigida, indicada por um altímetro
QNH Ajuste de subescala do altímetro para se obter a elevação estando em terra
TDD Test Driven Development
TI Tecnologia da Informação
UML Unified Modeling Language
XP Extreme Programming
ZDP Zona de Desenvolvimento Proximal

Sumário

1.1. Problema	3
1.2. Motivações e Justificativas	4
1.2.1. Motivações	4
1.2.2. Justificativas	7
1.3. Hipótese	8
1.4. Objetivos	9
1.4.1. Objetivo Geral	9
1.4.2. Objetivos Específicos	9
1.5. Metodologia	9
1.6. Contribuições da Pesquisa	10
1.7. Organização da Dissertação	11
2.1. Fatores Humanos e Complexidade	15
2.2. Processo de Aprendizagem em Sistemas Complexos	17
2.3. Controle de Tráfego Aéreo: Um Exemplo de Sistema Complexo	19
2.4. Engenharia de Resiliência	21
2.4.1. Análise das Tarefas Cognitivas	24
2.5. Considerações Finais do Capítulo	29
3.1. Abordagem Comportamental	33
3.2. Abordagem Cognitiva	34
3.3. Abordagem Metacognitiva	41
3.4. Considerações Finais do Capítulo	47
4.1. Desenvolvimento Ágil	52
4.1.1. FDD - Feature Driven Development	53
4.1.2. TDD - Test Driven Development	55
4.1.3. BDD - Behavior Driven Development	56
4.2. Desenvolvimento de Interface	57
4.2.1. Design Centrado no Usuário	60
4.2.2. Design Centrado na Tarefa	62
4.2.3. Design Centrado na Atividade	64
4.3. Objetos de Aprendizagem Baseados em Simulações	66

4.4.	Análise comparativa entre IHC e metodologias ágeis de ES	70
4.5.	Considerações Finais do Capítulo	84
5.1.	Seleção da Amostra	88
5.2.	Instrumentos.....	88
5.3.	Procedimentos	89
5.4.	Análise das Entrevistas	89
5.5.	Considerações Finais do Capítulo	93
6.1.	Metodologia Integrada de Desenvolvimento Ágil de ES e IHC.....	97
6.2.	Modelo de Projeto Instrucional.....	105
6.3.	Modelo de Projeto de Interface	111
	6.3.1. Estilos de Interação	112
	6.3.2. Tipos de Scaffolding	114
6.4.	Implementação	126
6.5.	Considerações Finais.....	129
7.1.	Avaliação da Metodologia de Desenvolvimento	132
7.2.	Avaliação do Modelo de OA Baseado em Simulação	138
	7.2.1. Estudo de Caso	138
	7.2.2. Pesquisa Experimental	140
7.3.	Considerações Finais do Capítulo	154
8.1.	Resumo do Trabalho.....	157
8.2.	Contribuições da Dissertação	159
8.3.	Problemas Encontrados	160
8.4.	Trabalhos Futuros.....	160

Capítulo 1 - Introdução

Neste capítulo é apresentada a pesquisa documentada nesta dissertação, abordando as motivações e justificativas, o problema e a hipótese de pesquisa, os objetivos da dissertação, a metodologia utilizada e a organização do texto.

Segundo Perrow (1999), sistemas complexos podem ser definidos como um conjunto de elementos interdependentes que interagem de forma organizada, baseados em regras simples, porém não lineares, onde cada um dos elementos componentes comporta-se, como um sistema cujo resultado é maior do que as unidades poderiam obter se funcionassem independentemente. Esse comportamento produzido é complexo derivado da relação conjunto de partes coordenadas voltadas a objetivos comuns formando um todo.

Para tentar impor algum tipo de organização dentro desse processo operacional complexo e auxiliar aos operadores no desempenho de suas tarefas, houve a substituição do controle humano por ferramentas tecnológicas, a fim de automatizar tarefas rotineiras.

A automação aumenta a confiabilidade na administração das variabilidades do ambiente por meio da padronização de operações de rotina alocadas na parte máquina do sistema.

As rotinas operacionais estabelecidas para o funcionamento interligado desses sistemas automatizados pode acarretar no aumento da probabilidade de ocorrência de situações novas e, conseqüentemente, de falhas não previstas ou não percebidas no sistema.

Essas falhas que podem resultar em sérios problemas são freqüentemente relacionadas à compreensão do real estado do sistema automático num dado momento.

Esse processo de constante automação de funções e procedimentos operacionais dentro de sistemas complexos visa não só atender a demanda crescente, mas fazê-lo de forma mais rápida, precisa e segura em substituição a processos operacionais em que a sobrecarga física e cognitiva se tornou um desafio à eficiência e segurança de um sistema.

O crescimento da automação de procedimentos operacionais anteriormente desempenhados por humanos promoveu o operador a uma função mais gerencial. Com essa mudança, houve uma transferência da sobrecarga física para um sistema autômato operado através de um sistema de codificação. O gerenciamento do sistema é realizado através da mediação por signos representativos de um processo organizacional. Esse sistema de códigos traduz-se em uma linguagem própria do ambiente operacional para a qual foi desenvolvido com o intuito de estabelecer uma comunicação entre homem e máquina através de uma linguagem objetiva. Essa delegação de tarefas operacionais dos seres humanos para as máquinas levou os operadores a desempenharem um número cada vez maior de tarefas cognitivas devido à possibilidade que a automação trouxe de acumular funções e a responsabilidade de gerenciar esse sistema de códigos.

1.1. Problema

Os operadores envolvidos em sistemas complexos atuam como mediadores das relações entre as variáveis do ambiente de atuação e os artefatos tecnológicos que operam tornando cada procedimento operacional único na medida em que a correlação de significados é a estrutura organizadora principal que rege essa relação. Assim, a associação de suas habilidades cognitivas e as mensagens dos diversos artefatos tecnológicos que operam, evidencia sua ação. Essa situação coloca a prova o conhecimento adquirido pelo operador em seu treinamento e a capacidade cognitiva, de abstração, percepção visual e espacial, para a interpretação de símbolos e números de uma série de artefatos de gerenciamento desses processos operacionais. Assim, podemos concluir que embora a automação possa reduzir a probabilidade de erros humanos ao nível da habilidade, pode criar outros ao nível do conhecimento, de modo que ela não necessariamente aumenta a confiabilidade da operação, além disso, torna a análise e cálculo dos riscos envolvidos na operação muito mais complexa.

Com a evolução tecnológica, o conhecimento vem sendo codificado e traduzido através de significados a fim de compactar esse volume cada vez maior de informação. O processo de automação dentro dos sistemas vem diminuindo os desafios cognitivos e simplificando o conhecimento que o operador deveria reter a fim de sintetizar as tarefas cognitivas e operacionais durante o desempenho de sua função. Esse reducionismo diminui o tempo de resposta entre as ações, mas, se por um lado a automação diminui o tempo perdido com tarefas que exigiam um alto nível de cognição, retirou também a necessidade de reter o conhecimento elementar que está por traz do funcionamento do equipamento. Esse conhecimento, em muitos casos é a base para uma resposta adaptativa em momentos de adversidade e que poderia auxiliar na retomada da estabilidade do sistema.

O conhecimento e a habilidade dos operadores são adquiridos principalmente na prática, quando vivenciam as demandas do sistema. Os mais experientes, criam um modelo mental, representativo das variabilidades as quais já se depararam e passam a identificar invariantes operacionais que lhes propiciaram a retomada do controle do desempenho do sistema. Nesse tempo, os operadores criam um modelo mental, representando o ambiente complexo das suas tarefas. O objetivo dessa representação é diagnosticar o presente estado do sistema para poder, posteriormente, antecipar os estados futuros e planejar a intervenção.

Muitas vezes esse conhecimento tácito é gerado de forma reativa derivado de um comportamento adaptativo. Portanto, o núcleo das questões está em entender como as pessoas pensam em função da complexidade e das configurações dinâmicas que caracterizam as tarefas do mundo real.

A capacidade adaptativa, cuja importância é amplificada em um momento imprevisto, exige a rapidez de uma resposta, que para ter maior probabilidade de sucesso, dependerá da capacidade cognitiva de articulação desses conhecimentos. Para que essa resposta adaptativa seja rápida e precisa há a necessidade de proporcionar aos operadores certas condições de treinamento onde o conhecimento e variabilidade do ambiente, sejam representativas e mais adequadas aos contextos que eles atuam. Portanto a formação de profissionais que atuam dentro de sistemas complexos exige a adequação da metodologia de ensino a fim de corresponderem de forma resiliente às variabilidades do sistema e do ambiente. Para isso a metodologia de ensino deve considerar a identificação de invariâncias e promover a capacidade do aluno desenvolver através do nexos recursivo a possibilidade de estabelecer a lógica de uma regra e assim estruturar o conhecimento necessário para responder de forma bem sucedida frente às variabilidades.

1.2.Motivações e Justificativas

A automação de processos e a informatização que por um lado facilitaram o processo operacional e permitiram lidar com o aumento da demanda de forma limitada, por outro, propiciaram ao sistema a operar com uma margem de segurança cada vez menor devido à confiabilidade proporcionada pela precisão própria dos artefatos tecnológicos. É necessário, portanto também investir no aperfeiçoamento no lado humano do sistema, compreendendo onde e de que forma a tecnologia pode se inserir no treinamento desses operadores de sistemas complexos.

1.2.1. Motivações

Para o estudo de sistemas complexos, a engenharia de resiliência é uma disciplina fundamental na gestão da segurança, onde melhorias são propostas a partir da análise das

variabilidades do ambiente e das relações entre elementos heterogêneos, humanos e não humanos que impõem um alto grau de complexidade no desempenho de tarefas cognitivas.

Enquanto o funcionamento dos sistemas permanece estável, não há possibilidade de identificar as conexões instáveis que o mantêm ou que poderão levá-lo a ruptura. As variações dessas relações é que fazem emergir as conexões instáveis. Essa capacidade de se flexionar perante as adversidades e retomar seu estado natural de funcionamento é que define o grau de resiliência de um sistema.

Em muitos sistemas, o que está sustentando operacionalmente a resiliência, na flexibilidade para satisfazer as variabilidades ambientais de forma adequada, é a "profissionalização" dos operadores, neste contexto, à capacidade de utilização de um conhecimento e experiência para construir e manter uma resposta adequada às diferentes demandas do ambiente operacional (algumas vezes imprevisíveis e ocasionais).

Para o desenvolvimento de programas de formação de profissionais que atuam em sistemas complexos, deve-se considerar que o aluno transforma o significado lógico do material pedagógico em significado psicológico na medida em que esse conteúdo se insere de modo peculiar na sua estrutura cognitiva. Com o controle desse processo, o aluno seria capaz de identificar invariâncias, constituir regras e assim desenvolver o raciocínio lógico necessário ao controle variável e flexível próprios de uma resposta adaptativa bem sucedida.

A incorporação de tecnologias ao ensino como recurso educacional tem sido incentivada pela capacidade de disseminação do conhecimento e principalmente pela possibilidade de criar ambientes onde a problemática, a atividade reflexiva, a atitude crítica, a capacidade decisória e a autonomia são privilegiadas e potencializadas. Dentre elas destacam-se os Objetos de Aprendizagem (OA), aplicativos educacionais que permitem ao aluno perceber onde o conteúdo instrucional se aplica em seu contexto real.

Esse recurso educacional possui características de desenvolvimento e utilização bastante flexíveis em acordo com as necessidades de treinamentos adequados a contextos variáveis. Portanto, os OAs modelados para simularem a variabilidade do cenário operacional, com a representação das condições mais atuais do estado de funcionamento do sistema e seus componentes, permitiriam também tornar mais resiliente a própria metodologia de treinamento de operadores de sistemas complexos ao viabilizar a constante atualização do conteúdo do treinamento.

Através desse recurso tecnológico e educacional, é possível recriar esse cenário operacional, onde o processo de aprendizagem seria desenvolvido a partir da retirada gradativa dos mecanismos de interação homem-máquina presentes em seu cotidiano operacional.

De acordo com o modelo de desenvolvimento de software educacional apresentado nesse estudo, seria possível estruturar o OA, através da organização do conteúdo disciplinar teórico e prático utilizando esses mecanismos de interação homem-máquina como mediadores do processo de aprendizagem. Seria possível assim, explicitar a definição dos conhecimentos que se traduzem em maior quantidade de significados e níveis de abstração, estruturando dentro do aplicativo os conhecimentos mais inclusivos aos níveis de linguagem mais altos e os conhecimentos subordinados nos níveis mais baixos.

Essa dinâmica de representação e tradução de dados do ambiente de atuação de um operador de sistema complexo seria trabalhada em um OA, através da combinação de coleta ou captura de dados dinâmicos online com a contextualização em representações gráficas e simulações interativas. Através da manipulação interativa e da provocação da interseção entre variáveis presentes na operação do sistema, o aluno seria levado a perceber a composição das relações entre os conceitos a serem aprendidos e a interação com o ambiente operacional.

A capacidade de identificar as invariâncias e constituir regras de controle do sistema a partir do entendimento de conceitos operacionais poderia assim desenvolver nos profissionais que atuam em sistemas complexos o raciocínio lógico. Esse raciocínio permitiria ao aluno, estabelecer regras de seleção para abstrair o conhecimento necessário ao controle variável e flexível, próprios de uma resposta adaptativa bem sucedida.

As características específicas dos cenários e das tarefas dos operadores de sistemas complexos junto às necessidades pedagógicas acabam por dificultar o desenvolvimento dos requisitos necessários e característicos desse tipo de aplicativo.

A dificuldade maior de desenvolver esse modelo de OA advém da necessidade de construir esses mecanismos de interação homem-máquina, onde as relações próximas entre funcionalidades e representação gráfica devem atuar em conjunto como mediadores do processo de aprendizagem.

As abordagens de desenvolvimento tradicionais já não satisfazem as necessidades das propostas educacionais atuais, principalmente as de domínio complexo, pois nem sempre

contribuem para a aprendizagem final. Portanto, é preciso pesquisar abordagens de desenvolvimento de recursos educacionais mais adequados à prática pedagógica.

Nos últimos anos, o uso de OA se tornou freqüente na prática pedagógica, entretanto, percebe-se que o atual modelo instrucional e de desenvolvimento tem generalizado as características fundamentais desse recurso. Isso é atribuído ao alto custo de produção e o vasto conteúdo educacional, o que inviabiliza a adequação das funcionalidades dos AO às necessidades específicas de cada curso. No desenvolvimento de um OA para processo de ensino e aprendizagem da disciplina Fundamentos da Meteorologia, foi possível perceber a necessidade de inovação no modelo de projeto instrucional e de desenvolvimento para atender as características específicas de treinamento dos controladores de tráfego aéreo. Motivados também pela expansão e uso das técnicas de desenvolvimento ágil de software da engenharia de software (ES) e conceitos de interface homem-computador (IHC), verificou-se que a aplicação de suas práticas poderia incorporar melhorias no projeto instrucional e, conseqüentemente, na produção de recursos didáticos.

1.2.2. Justificativas

Para nivelar a capacidade de aprendizagem entre os diferentes níveis de cognição seria preciso fornecer ao aluno estruturas de apoio ao processo de aprendizagem, mais acessíveis ao seu nível cognitivo. Essas estruturas seriam fornecidas e retiradas gradativamente proporcionando, a partir do nível mais alto de linguagem e automação em que está codificado o conhecimento, a desconstrução dos signos-significantes estabelecidos em linguagens anteriores e níveis menores de automação. Isso ocorreria até que se encontrasse um nível de invariância ao qual ele pudesse estabelecer uma correlação de significados entre a ação executada e a resposta do sistema, seja entre o nível e estrutura de linguagem e automação identificada com sua capacidade cognitiva, seja entre o nível e estrutura de linguagem e automação de nível mais alto em que se encontra codificado o conhecimento.

Baseado nesse processo de desconstrução de signos-significantes, o aluno seria capaz de identificar a regra estabelecida pela correlação entre esses níveis de significados, responsáveis pela estruturação do conhecimento. Desta forma, o aluno estaria apto a elaborar uma explicação lógica a partir da realidade percebida e transformada pela interação homem-máquina.

Essa perspectiva hierarquizada no processo cognitivo se traduziria, portanto, a cada nível de linguagem e interação, na redução fenomenológica de sentido capacitando o aluno a abstrair de forma seletiva o conhecimento adquirido. Assim, toda a realidade passaria a ter uma existência puramente fenomenal no sentido informacional. Esse reducionismo só seria percebido quando houvesse uma variação que os fizesse retroagir cognitivamente, em busca de invariantes pré-fixadas. Esses elementos de suporte os capacitariam a conectar o novo conhecimento que surge com os significados anteriormente estabelecidos e fundamentados, melhorando a capacidade de leitura e compreensão dos significados operacionais e dos conceitos a eles atribuídos.

Através da estruturação do conhecimento com mecanismos de interação homem-máquina contextualizados nos objetos de aprendizagem, é possível fornecer um modelo lógico a ser elaborado pelo aluno, permitindo a ele a modelagem cognitiva através da retirada gradativa do processo de automação. Portanto, em softwares educacionais, as funcionalidades do aplicativo, assim como os elementos de interface, assumem um caráter pedagógico. Essa característica implica na forma como esse software deve ser pensado e desenvolvido já que os elementos de interação que cumprirão esse papel pedagógico são resultantes do trabalho de design de interface e da engenharia de software.

1.3. Hipótese

Para atender ao desenvolvimento dessas funcionalidades e mecanismos de interação homem-máquina que vão além de seu propósito operacional e tomam um caráter pedagógico, foi importante integrar conceitos de áreas distintas do conhecimento como pedagogia, design e informática em uma única metodologia de desenvolvimento. Verificou-se, portanto, a necessidade de integração de duas áreas componentes do ciclo de desenvolvimento de um software, a engenharia de software (ES) e a de interface humano-computador (IHC).

Caracteriza-se assim a hipótese de que uma metodologia que integre essas áreas pode viabilizar o desenvolvimento de aplicativos educacionais apropriados ao treinamento de operadores de sistemas complexos que contemplem funcionalidades e representações da variabilidade do cenário operacional.

1.4. Objetivos

1.4.1. Objetivo Geral

O objetivo geral dessa dissertação é propor uma metodologia de desenvolvimento de objetos de aprendizagem baseados em simulações para o treinamento em sistemas complexos.

1.4.2. Objetivos Específicos

De forma a alcançar o objetivo geral, os seguintes objetivos específicos foram definidos:

Criar um modelo de desenvolvimento de simulações representando a variabilidade do ambiente de atuação dos operadores de sistemas complexos através de simulações modeladas integrando metodologias ágeis de ES e conceitos de IHC. O modelo deve contemplar às seguintes etapas da elaboração do aplicativo:

- Metodologia de Levantamento de Requisitos;
- Modelo de Projeto Instrucional;
- Modelo de Projeto de Interface;
- Metodologia de Desenvolvimento.

Verificar a viabilidade de implementação do modelo de objeto de aprendizagem proposto.

Observar se a metodologia proposta facilita o desenvolvimento de objetos de aprendizagem adequados ao treinamento de operadores de sistemas complexos que permitem visualizar diferentes pontos de vista, comprovar hipóteses, identificar e consolidar as invariâncias.

Observar se há correlação entre os critérios pedagógicos e ergonômicos no modelo de objeto de aprendizagem construído sob a metodologia proposta.

1.5. Metodologia

A presente pesquisa de dissertação foi conduzida conforme os procedimentos metodológicos apresentados a seguir.

A primeira etapa compreendeu a realização de uma revisão bibliográfica em artigos científicos, dissertações, teses e em entrevistas informais com especialistas, para compor a fundamentação teórica necessária para essa pesquisa.

Posteriormente, foram feitas entrevistas semi-estruturadas com instrutores e especialistas no domínio da atividade profissional de controle de tráfego aéreo considerada em exemplo neste estudo para realizar o levantamento dos requisitos necessários ao desenvolvimento do software educacional. Nesta etapa, também foram entrevistados alunos, a fim de traçar um quadro comparativo entre a percepção do aluno sobre a metodologia de ensino e as estratégias adotadas pelos instrutores. Esse procedimento também teve como objetivo aprofundar os conhecimentos referentes ao caso escolhido como exemplo, bem como identificar necessidades do atual curso de formação.

Em seguida foi desenvolvido um protótipo para verificar a viabilidade da hipótese que orienta essa dissertação, o qual abrangeu as seguintes fases: especificação, modelagem, descrição e implementação da ferramenta e validação preliminar com especialistas.

Para avaliar a proposta, foi aplicado um estudo de caso exploratório com o objetivo de analisar a relação experimentação - objeto de aprendizagem e verificar se a metodologia utilizada possibilitaria ou não a ação investigativa e reflexiva do aluno sobre os conceitos abordados.

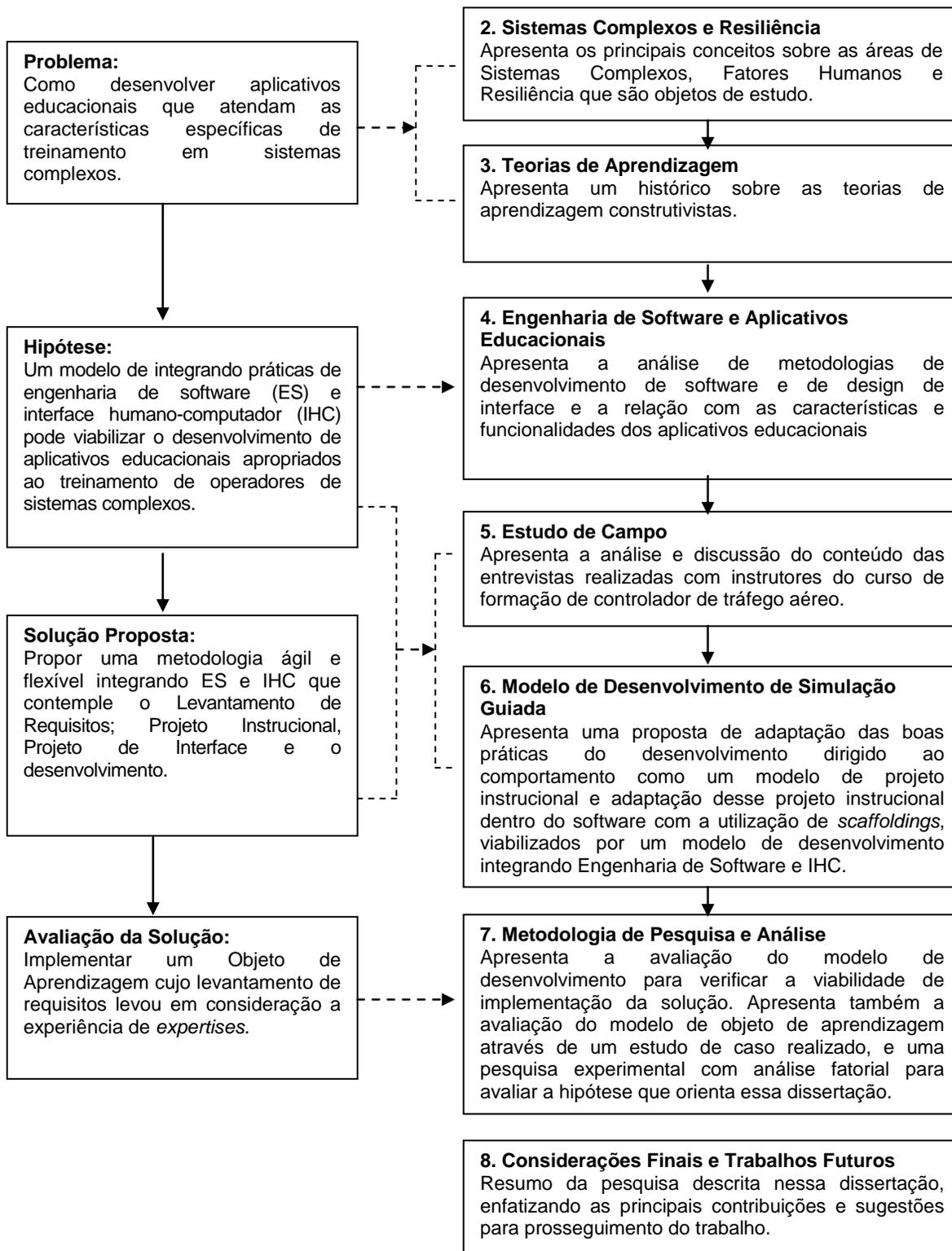
Foi realizada também uma pesquisa-ação para testar a viabilidade da proposta do modelo de desenvolvimento integrado de software educacional.

Ao final, foi realizada uma pesquisa experimental, buscando avaliar a metodologia de desenvolvimento proposta e o modelo de objeto de aprendizagem dentro da hipótese formulada a fim de verificar a correlação de critérios ergonômicos, produtos da integração das disciplinas de ES e IHC e critérios pedagógicos.

1.6. Contribuições da Pesquisa

A principal contribuição desta dissertação é a proposta de uma metodologia de desenvolvimento de software educacional baseado nas práticas ágeis da engenharia de software e conceitos de interface humano-computador. A proposta traz contribuições para as etapas principais etapas de desenvolvimento de um software sendo; metodologia de levantamento de requisitos; modelo de projeto instrucional; modelo de projeto de interface e metodologia de desenvolvimento. A proposta tem o objetivo de contribuir para que o aluno interiorize o conceito de tal forma que seja possível a sua abstração e aplicação em diferentes situações, inclusive, as não previstas durante o seu treinamento.

1.7. Organização da Dissertação



Capítulo 2 - Sistemas Complexos e Engenharia de Resiliência

Neste capítulo é feita uma revisão bibliográfica onde é definido o que são Sistemas Complexos, suas principais características e o domínio em que se baseia esta pesquisa. Além disso, são apresentados os principais conceitos e contribuições da disciplina de Engenharia de Resiliência, no estudo sobre fatores e comportamento humano dentro de Sistemas Complexos.

Ao final do capítulo são feitas as considerações finais com a apresentação de uma metodologia de levantamento de requisitos como parte da proposta desta pesquisa.

Os conceitos a respeito de sistemas complexos são elaborados a fim de compreender o funcionamento de sistemas físicos, biológicos e sociais. Essas definições estendem-se e aplicam-se em todos os campos da ciência como, por exemplo, engenharia, administração, modelagem, simulação computacional e educação.

Apesar de ser uma ciência interdisciplinar, diferentes teóricos destacam a importância de uma abordagem transdisciplinar dos fenômenos e a mudança de paradigma, abandonando o reducionismo e dando lugar a uma descrição de sistemas como um todo indissociável.

Pavard e Dugdale (2005) definem sistemas complexos como um sistema para o qual é difícil, se não impossível restringir sua descrição a um número limitado de parâmetros ou características variáveis sem perder a essência de suas propriedades funcionais globais.

Ainda de acordo com Pavard e Dugdale (2005), um sistema começa a ter comportamento complexo no instante em que suas partes interagem de modo não-linear. A característica de não-linearidade é apropriada para diferenciar entre um sistema complicado e um sistema complexo. Enquanto um sistema complicado é composto de muitas partes funcionalmente distintas, mas previsíveis, o sistema complexo é não previsível por interagir de maneira não-linear com seu ambiente e por seus componentes terem a capacidade de auto-organização.

Para Perrow (1999) sistemas complexos, são verdadeiras organizações e, mais do que isso, de uma organização de organizações, que existe uma tendência às interações, característica entre os sistemas não apenas de seus operadores. Ainda segundo Perrow (1999), esses sistemas têm o potencial de apresentar interações complexas, ou seja, aquelas em que um componente pode interagir com outros componentes em seqüência não esperada ou não planejada e, também, não visível ou não imediatamente compreensível.

Em sistemas complexos o conjunto de tarefas é desempenhado por diversos elementos, o que pode desenvolver nos trabalhadores uma percepção fracionada da importância que tem o desempenho de sua tarefa para a manutenção do funcionamento seguro de todo o sistema.

As características já convencionadas até o momento sobre sistemas complexos apontam os seguintes princípios (Rezende *et al.*, 2008) :

1. **Sistemas complexos são complexos.** Partindo do princípio de que os sistemas complexos são complexos – em essência – na sua descrição, abre-se espaço para descrições alternativas, mais simples e melhores do mundo real.

2. ***Sistemas complexos são baseados em agentes.*** Os agentes individuais constituem os blocos básicos de construção do sistema.
3. ***Sistemas complexos são caóticos.*** Existem períodos irregulares e imprevisíveis, de total aleatoriedade, onde nada é previsível ou determinado.
4. ***Sistemas complexos evoluem.*** Isto significa que variam com o passar do tempo ou com pequenas mudanças nas condições iniciais.
5. ***Sistemas complexos são abertos.*** Trocam informação, energia e massa com o ambiente. Por causa disto, não existe um ponto de equilíbrio a ser alcançado e está sujeito a constantes reorganizações.
6. ***Sistemas complexos são auto-organizáveis.*** Propriedade emergente do sistema; a habilidade de criar nova estrutura sem qualquer pressão externa.
7. ***Sistemas complexos são não-determinísticos.*** Isto é, exibem comportamento imprevisível após muito tempo, incluindo comportamento caótico debaixo de certas condições.
8. ***Sistemas complexos são sensíveis a feedback.*** O sistema evolui e sofre influências da realimentação positiva e negativa. Estas mudanças são o resultado da avaliação que os agentes recebem como resultado de suas atividades.
9. ***Sistemas complexos são interdisciplinares.*** Um aspecto importante da abordagem de sistemas complexos é o reconhecimento que diferentes tipos de sistemas incluem auto-regulação, avaliação ou adaptação em sua dinâmica e, assim, podem ter uma estrutura subjacente comum apesar das aparentes diferenças.
10. ***Sistemas complexos são hierárquicos.*** Por exemplo, um sinal luminoso que envia um estímulo à retina é tratado em diferentes níveis antes de atingir a sede cerebral da sensação visual.

Para Wilensky e Resnick (1999) a principal dificuldade para compreender os sistemas complexos vem da necessidade de conectar os fenômenos que ocorrem em nível microscópico com os que ocorrem no nível macroscópico. Para construir essa relação, entre esses dois níveis que compõem os sistemas complexos, as pessoas tendem a se apoiar em uma representação simplificada desses de seus efeitos, através do encapsulamento em significados constituídos por metáforas.

Entender a dinâmica dos sistemas complexos trata-se de compreender as interações entre as partes, promovendo o conhecimento através da percepção fenomenológica da atuação das partes individuais sobre a função final do sistema.

As relações entre elementos componentes de um sistema complexo são definidas sob a delimitação de regras de funcionamento, que estão em constante processo de ajuste para comportar a principal característica que define um sistema complexo que é a variabilidade de seu funcionamento devido a seu ambiente operacional.

2.1.Fatores Humanos e Complexidade

Uma análise abrangente sobre complexidade através de múltiplos domínios mostra muitas definições como às características de capturar o "tamanho", "contar" ou "número" de itens em um objeto (Edmonds, 1999). O número de linhas de código contido em um programa de computador, por exemplo, é uma medida comum de complexidade do programa. No entanto, como Edmonds (1999) aponta, o tamanho parece destacar um potencial para a complexidade, mas pode não ser suficiente para dar conta da ordem de grandeza do que se entende por complexidade.

De acordo com Edmonds (1999), uma segunda característica fundamental das definições de complexidade é a sua associação com os objetos, conceitos, ou problemas “compostos de partes interligadas”. A noção de "interconexões" é indicativa da importância das relações entre os elementos constitutivos de uma situação ou problema. A presença de dependências entre as partes parece ser uma condição necessária para a complexidade. Algo facilmente decomposto e com componentes não interagindo entre si geralmente não é considerada complexa. A complexidade pode ser entendida como um constructo multidimensional com atributos que engloba o número e variedade de elementos, bem como as relações entre eles.

Ainda de acordo com Edmonds (1999), a terceira característica fundamental é que a complexidade depende do modo como o objeto ou problema é representado. As representações determinam o que são consideradas as partes do objeto ou problema e as relações resultantes. A representação do mesmo objeto ou problema de duas maneiras diferentes pode alterar significativamente a complexidade. A escolha das representações é freqüentemente uma consequência da tarefa.

Esta definição capta uma noção essencial da complexidade cognitiva de muitas tarefas no controle do tráfego aéreo. Apesar da disponibilidade de informação quase total da posição dos aviões, através de um radar e onde está previsto o avião ir (por exemplo, através das strips de vôo, uma espécie de etiqueta virtual de identificação), formular expectativas precisas da evolução de uma situação de tráfego aéreo é muito difícil.

O termo complexidade é estudado em outras áreas do conhecimento. Dentro da literatura na psicologia, por exemplo, complexidade cognitiva é usada como uma definição que descreve um indivíduo que usa um grande número de processos internos para perceber e raciocinar sobre como o mundo tem uma complexidade cognitiva alta. A segunda utilização é como uma referência a uma teoria para estudar os seres humanos como processadores de informação (Hansman, 2008 b).

A complexidade cognitiva de um indivíduo irá refletir a sua capacidade de diferenciar, ou quebrar informações em unidades menores, e integrar ou combinar unidades de informação em um todo maior (Hansman, 2008 b). Definições formais de complexidade computacional são comuns na literatura de ciência da computação. A minimização do número de elementos utilizados para representar ou gerar um objeto ou conceito é muitas vezes associados com a complexidade. Por exemplo, uma informação algorítmica é uma medida curta de um programa de computador (algoritmo) que pode produzir uma dada linha de código, ou seja, o menor programa necessário para produzir uma saída especial (Edmonds, 1999). Para um dado algoritmo, a análise de complexidade algorítmica pode expressar a complexidade tanto em termos do número mínimo de passos necessários, e / ou a quantidade mínima de memória, ou o espaço necessário, para calcular uma solução para o problema.

A complexidade é um conceito geralmente associado a uma dimensão quantitativa, derivada da correlação entre o número de elementos administrados e a complexidade do processo administrativo.

Na Teoria Geral de Sistemas (Rezende *et al.*, 2008), os sistemas são analisados e classificados como multivariáveis em função da quantidade de variáveis manipuladas e das relações estabelecidas entre estas variáveis. As relações entre as variáveis, podendo ser de natureza linear ou funcional.

Essas relações, são estabelecidas pela existência de sub-sistemas dentro de sistemas e caracterizam um fluxo constante de entradas, onde o efeito sobre o sistema como um todo é

caracterizado pela unidade e sendo assim existe uma múltipla interação entre as partes (macro sistema, sistema, subsistema).

2.2. Processo de Aprendizagem em Sistemas Complexos

Processos de aprendizagem sobre sistemas complexos oferecem explicações aos alunos através da experiência com sub-representações, desenvolvendo neles a capacidade de perceber e explicar em grande escala, fenômenos macros em termos locais, através de micro eventos em cada nível.

Wilensky e Resnick (1999) atribuem a dificuldade na compreensão de sistemas complexos especificamente a necessidade de ligar os fenômenos que ocorrem em nível microscópico com aqueles que ocorrem a nível macroscópico. Wilensky e Resnick (1999) indicam que as pessoas costumam usar elementos invariantes para construir pontes entre níveis, interpretando a relação entre o macro e o micro, em termos de metáforas de hierarquia ou de contenção. Wilensky e Resnick também apontam motivos da dificuldade em entender as explicações emergentes (Wilensky & Resnick, 1999). Eles descrevem dois componentes dessa dificuldade:

- 1) Tendência para compreender fenômenos complexos como orquestrado por um líder ou concebido por uma única entidade;
- 2) Dificuldade em ver a aleatoriedade como construtora de uma ordem e de um padrão.

Para Wilensky (2008) as dificuldades estão localizadas nas limitações da carga cognitiva de interações, na computação de grande número de objetos, nas limitações da percepção de objetos em múltiplas escalas e na falta de pistas contextuais em nível micro. Quando esses obstáculos são minimizados, Wilensky (2008) constata que os alunos podem construir espontaneamente pontes entre os níveis. A interação do aluno com o ambiente através de objetos reduz as limitações cognitivas e de percepção, os alunos são capazes de usar objetos existentes e recursos para raciocinar sobre a dinâmica da situação, configurando a chamada manipulação de conceitos que são materializados pela relação entre a ação e estímulos recebidos como respostas.

Wilensky, (2008) descreve o que chamou de fases de raciocínio emergente percebido por ele no desenvolvimento de alunos do ensino médio. Os alunos normalmente começam por :

- 1) Prestar atenção a um único nível;

- 2) Movem-se entre os níveis, mas ainda vendo o nível agregado como uma simples coleção de indivíduos, por vezes, referindo-se ao nível macroscópico como uma ilusão alegando que apenas o nível individual é real;
- 3) Vêem o nível emergente como uma nova entidade e as suas propriedades como sendo diferentes dos seus constituintes;
- 4) Vêem todas as entidades estáveis como emergente e como processos com algum tipo de equilíbrio dinâmico.

Resnick & Wilensky (1999), destacam também outro aspecto difícil de superar para a percepção dos níveis é a necessidade de compreender não só as estruturas de nível inferior, mas também suas interações (e mecanismos de interação), funções e comportamento.

A maior diferença entre modelos mentais de especialistas e iniciantes que trabalham com sistemas complexos é em termos da percepção sobre como as partes que compõem o sistema funcionam e se comportam em relação umas as outras, não em termos de suas estruturas intrínsecas. Para Resnick e Wilensky (1999), é o entendimento das interações entre as partes que promove o conhecimento e verdadeira ponte para as explicações. Para este efeito, as abordagens para o ensino sobre sistemas complexos oferecem grande vantagem as descrições do sistema baseadas em equações. Essa abordagem enfoca as regras que regem as interações dos elementos permitem que a descrição seja revelada pelo sistema ao invés de partir de uma equação. Por outro lado, uma abordagem baseada em equações que explicitam qualidades macroscópicas como a população ou a temperatura pode incentivar a falsa crença de que essas propriedades macroscópicas possuem e seguem suas próprias regras independentes de pontes entre interações elementares. Uma perspectiva interativa de sistemas complexos fornece uma ponte entre as classes comportamentais do sistema e as classes formais.

Pessoas que trazem à mente um modelo esquemático do sistema complexo que atuam, quando são confrontados com a sua equação equivalente ou conjunto de regras (representações formais), tem maior probabilidade de raciocinar de forma flexível sobre o conhecimento formal. Desta forma, alunos e pesquisadores precisam de ferramentas para conectar-se a dinâmica dos sistemas complexos interligando o conhecimento formal aos fenômenos do mundo real.

2.3. Controle de Tráfego Aéreo: Um Exemplo de Sistema Complexo

Neste trabalho, foi definido como exemplo para o estudo de sistemas complexos, o controle de tráfego aéreo. O controle do tráfego aéreo pode ser definido como um sistema complexo, por ser um conjunto de elementos interdependentes que interagem de forma organizada, baseado em regras simples, porém não lineares (Perrow, 1999).

Os controladores de tráfego aéreo atuam como mediadores das relações entre os vôos sob sua orientação onde cada procedimento operacional é único. Assim, através de sua ação é possível evidenciar sua composição heterogênea, transformar, traduzir, e modificar o significado de elementos ao interpretarem através das associações de suas habilidades cognitivas, as mensagens dos diversos artefatos tecnológicos que operam e medeiam. Sua ação faz com que pilotos e outros controladores que compõem o sistema atuem.

As tarefas cognitivas já em uma configuração normal de operação do sistema exigem a capacidade adaptativa do controlador mediante a imprevisibilidade do ambiente em que trabalha. O contínuo crescimento do tráfego aéreo, com o excesso de aeronaves, há um aumento na pressão principalmente sobre os operadores devido à sobrecarga do sistema com o conseqüente congestionamento de aeroportos e aerovias levando a operação de separação vertical entre aeronaves ao limite. Essa situação coloca a prova todo o conhecimento adquirido pelo controlador em seu treinamento e a capacidade cognitiva, de abstração, percepção visual e espacial, para a interpretação de símbolos e números da tela do radar. Sua tarefa é atuar como mediador de uma série de instrumentos de gerenciamento da navegação das aeronaves monitorando as respectivas altitudes e distância em que operam onde o momento mais crítico é durante a aproximação das aeronaves em procedimento de pouso.

O controle de tráfego aéreo é um sistema dinâmico, onde os controladores recebem constantemente um grande volume de informações de várias fontes para acompanhar as mudanças no ambiente, tomar decisões e executar ações eficazes de maneira oportuna.

Os conhecimentos básicos necessários para atuar como controlador de tráfego aéreo foram definidos pela Organização de Aviação Civil Internacional (OACI), cabendo aos países-membro o planejamento e definição dos recursos educacionais que serão utilizados para apresentar, dentre outros, os seguintes assuntos:

- Normas de tráfego aéreo;

- Equipamentos para controle de tráfego aéreo;
- Conhecimentos gerais (teoria de vôo; reconhecimento e performance de aeronaves);
- Fatores humanos;
- Meteorologia;
- Navegação;
- Procedimentos operacionais.

No Brasil, o único órgão credenciado para licenciar controlador de tráfego aéreo é o Comando da Aeronáutica que oferece o curso para controladores civis e militares, respectivamente, no Instituto de Controle do Espaço Aéreo (ICEA) e na Escola de Especialistas da Aeronáutica (EEA), ambas as instituições da área de ensino da Força Aérea Brasileira.

É necessário enfatizar que, por ser o Brasil um país-membro da OACI, a estrutura do curso de formação de controlador de tráfego aéreo oferecido pelo ICEA compreende disciplinas de formação técnica, aulas teóricas e prática em simuladores como nos demais países-membro.

No ICEA o curso de controlador de tráfego aéreo tem duração de 09 meses, distribuídos entre a formação de conhecimentos básicos (disciplinas teóricas) e a formação de classificação (prática simulada de Torre de Controle, Controle de Aproximação e Controle de Área), projetadas para dar ao aluno conhecimento, compreensão e habilidades que o permitirão progredir na carreira de controlador de tráfego aéreo.

Para atingir tal propósito, o treinamento do controlador de tráfego aéreo consta de forte embasamento teórico a fim de preparar o aluno para uma formação prática correspondente, seguida de exercícios simulados que objetivam a integração das habilidades anteriormente desenvolvidas em cada disciplina.

Para se qualificar ao trabalho no sistema de controle de tráfego aéreo, os alunos devem concluir com êxito tanto as disciplinas teóricas quanto a prática simulada, onde são avaliadas as habilidades de compreender funções relevantes ao controle de tráfego aéreo e de tomada de decisões imediatas. Por isso, educação de qualidade e compartilhamento do conhecimento precisam ser vistos como requisitos fundamentais para a elaboração do treinamento fornecido aos futuros controladores de tráfego aéreo.

Depois de ter concluído o curso o controlador é encaminhado para estágio de habilitação técnica, uma espécie de adaptação na área de trabalho, em que um supervisor ou controlador mais experiente acompanha seu desempenho na comunicação com os pilotos, no repasse de informações e na tomada de decisão.

No Brasil, os controladores recém formados iniciam suas carreiras profissionais na torre de controle onde são responsáveis pela circulação de pessoas e veículos na área de manobras (pistas de táxi), além de controlar os pousos, as descolagens e o deslocamento interno nos aeródromos.

Nesta pesquisa será posta ênfase, especificamente, nas disciplinas que compõe a formação de conhecimentos básicos do curso para controladores civis.

O sucesso de uma organização (grupos e indivíduos) resiliente reside no fato de que ela reconhece, adapta e absorve as variações, mudanças, distúrbios, rupturas e surpresas que são rupturas que acontecem além dos limites de distúrbios originalmente concebidos para o sistema (Hollnagel *et al.*, 2006).

2.4. Engenharia de Resiliência

Para sistemas complexos como o controle do tráfego aéreo, a engenharia de resiliência é uma disciplina fundamental na gestão da segurança. Através desta disciplina, melhorias são propostas a partir da análise das variabilidades do ambiente e do estudo das relações entre elementos heterogeneos, humanos e não-humanos que impõem um alto grau de complexidade no desempenho de tarefas cognitivas.

A resiliência é um paradigma que se concentra em entender como as pessoas lidam sob pressão com a complexidade do sistema de forma bem sucedida (Hollnagel, 2008). A resiliência é centrada na fiscalização dos limites e condições do atual modelo de competência ou da ampliação e adaptação ao modelo que melhor acomodar novas exigências.

O aumento da demanda do tráfego aéreo vem conseqüentemente proporcionando o aumento da carga cognitiva, criando um ambiente operacional em que os fatores humanos são cada vez mais exigidos.

Para Hollnagel (2008) o nível e a qualidade do treinamento, juntos com a experiência operacional, determinam o quanto o operador está bem preparado para as várias situações, conseqüentemente o quanto variável e flexível o desempenho poderá ser.

“A idéia que eu procuro passar pra eles na meteorologia é que os fenômenos vão acontecer, nós seres humanos não somos capazes de evitá-los, mas eu sei o que fazer quando eles ocorrerem”. (Instrutor do curso de formação de Controladores de Tráfego Aéreo - ICEA, 2007)

Segundo Crandall (2006), a investigação sobre resiliência focada nos fatores humanos se apóia na combinação de três temas: (a) cognição, incluindo a cognição distribuída em situações exigidas; (b) explicação dos objetivos e metas hierárquicas nas configurações naturais e (c) características das súbitas demandas de recursos e as reações que eles evocam. Através da análise das tarefas cognitivas, é possível compreender o raciocínio humano, identificando o desenvolvimento e a aplicação da lógica na realização de tarefas.

Para Hollnagel (2008), o desempenho variável é necessário se o conjunto de sistema cognitivo, ou seja, um sistema sociotécnico, composto de homem e máquina é bem sucedido em lidar com a complexidade do mundo real.

No Brasil, o aumento do tráfego aéreo a cada ano, vem fazendo com que, em alguns pontos do território nacional, haja excesso de tráfego de aeronaves, com conseqüente congestionamento de aeroportos e aerovias. Devido a isso, o Airsafetygroup (2003) aponta que atividade de controlador de tráfego aéreo tem demandado mais atenção e aplicação do conhecimento adquirido sobre as condições de trabalho com a utilização de diferentes tecnologias.

Os controladores de tráfego aéreo estão sujeitos à situação momentânea do tráfego aéreo sendo que, de acordo com a variação do tráfego, aumenta a necessidade da “resolução rápida de conflitos”. A variação do fluxo de aeronaves em um mesmo espaço aéreo faz com que o controlador tenha de trabalhar com uma aproximação maior das aeronaves e, em alguns casos, tendo de lidar, ao mesmo tempo, com uma quantidade superior à esperada de aeronaves.

Diferentes sistemas (humanos e tecnológicos) e subsistemas estão atuando conjuntamente (combinação de humanos e tecnologias) durante o controle de tráfego aéreo. Tal controle compreende, no mínimo, a sala de controle, o espaço aéreo e as aeronaves, e envolve a equipe de controladores, que têm que tomar várias decisões e ações, com base em dados dos equipamentos e análise das informações. Diferentes situações, muitas vezes complexas, e em sua maioria dinâmica, exigem esforço cognitivo e habilidade do controlador para mudar, rapidamente, seu comportamento, a fim de se ajustar a cada situação. Portanto, uma atividade cognitiva não pode ser vista como uma atividade de um único indivíduo, pois ela é mais complexa, já que representa

uma atividade em que os indivíduos/equipamentos/trabalho interagem entre si e, em geral, está em constante transição e evolução, com novas demandas, novos limites e novas tecnologias. A incorporação de novas tecnologias para minimizar incertezas e falhas que advêm dessa complexa relação entre elementos que compõem o sistema acabam por gerar outros problemas, pois mudam a relação humano/tecnologia/trabalho. Além do mais, nem sempre estas tecnologias são usadas como o projetista concebeu, pois são moldadas pelos usuários para melhor atender às necessidades do trabalho real.

Sob esta ótica, com o foco central na segurança e, portanto, com o intuito de tornar um sistema mais resiliente, a engenharia de resiliência busca não só abordar o que deveria ter sido feito, mas explicar o que aconteceu, quando do acontecimento de uma falha (Hollnagel *et al.*, 2008). Através da Engenharia de resiliência, é possível principalmente, aprender com as quase falhas para entender as variabilidades e os limites do sistema e, assim, prever e preparar para uma ação frente as possíveis falhas.

Hollnagel *et al.* (2008) conceituam a engenharia de resiliência como um modelo na gestão de segurança focada em como ajudar as pessoas a fazer frente à complexidade dos sistemas. A fonte da resiliência está no entendimento de como as pessoas têm sucesso quando ameaçam falhar. Entender como cada trabalhador executa sua atividade mostra como ele está tentando antecipar uma situação que pode levar a uma falha. O importante é entender como ele cria e sustenta uma estratégia de percepção de uma falha, tornando assim o sistema mais resiliente (Hollnagel *et al.*, 2008).

A interdição do processamento no nível em que a ação humana se dá de forma automática, conseqüentemente exige a ativação do controle consciente não automatizado como pré-requisito para a continuidade do processo operacional (Rasmusem, 1994). Essa variabilidade de processamento permanece subjacente aos aspectos objetivos visíveis das tarefas ou atividades do trabalho. Trata-se, portanto da adaptabilidade promovida pela cognição do operador frente à variabilidade e imprevisibilidade situacional que poderia comprometer o funcionamento do sistema.

Segundo Rasmussem (1994) mesmo parecendo realizar tarefas completamente automatizadas por regras de funcionamento do sistema, essas atividades exigem o desenvolvimento de estratégias mentais hipotético-dedutivas e de indução por parte dos operadores, a fim de suprir falhas do fornecimento de dados. Portanto, os operadores de sistemas

complexos devem ser capacitados em seu treinamento, a elaborar representações cognitivas do estado do processo operacional e assim simular mentalmente a seleção de possíveis modelos cognitivos mais adequados ao objetivo de responder adaptativamente frente à variância responsável por interromper o processo. Rasmussen (1994) identifica um sistema cognitivo subjacente ao processo de trabalho como objetivo e tangível. Esse sistema opera no que ele definiu como o mais alto nível de comportamento, resultado de uma atividade cognitiva intensa pautada pela mobilização e manipulação cognitiva de conhecimento simbólico abstrato. Portanto, somente os comportamentos baseados em habilidades sensório-motoras e respostas aos sinais do ambiente caracterizados por Rasmussen (1994) como o nível mais elementar de controle, não seriam suficientes para enfrentar momentos de variabilidade operacional.

Essa capacidade em lidar com a complexidade, esforço cognitivo e habilidade para mudar rapidamente seu comportamento, ajustando-se a cada situação é a característica mais importante no aperfeiçoamento da capacitação dos operadores de sistemas complexos. Esse processo implica em adquirir novas competências, associadas a conhecimentos previamente adquiridos e relacioná-las em um processo dinâmico, resultando no eventual acesso a um nível de desenvolvimento cognitivo superior. Torna-se, portanto necessário compreender quais os fatores e, sobretudo, quais os processos e os mecanismos que estão por trás desta resposta adaptativa, pois só a partir desse conhecimento é possível saber o que é necessário para promover essa flexibilidade cognitiva.

2.4.1. Análise das Tarefas Cognitivas

Os operadores de sistemas complexos criam um modelo mental, representando o ambiente complexo das suas tarefas. O objetivo dessa representação é diagnosticar o presente estado do sistema para poder, posteriormente, antecipar os estados futuros e planejar a intervenção.

Examinar o modelo cognitivo dos operadores de sistemas complexos, assim como a percepção situacional pode levar a um entendimento de como esses operadores antevêm as relações dos diferentes sistemas envolvidos, proporcionando comportamentos adaptativos para minimizar falhas.

Com a análise das tarefas cognitivas, parte-se do plano da coleta de informações sobre a tarefa, para a constituição da modelagem do processo cognitivo. Segundo Rasmussen (1994) consiste em se

passar de uma descrição dos processos cognitivos, feita através da análise das tarefas cognitivas, para uma descrição em uma linguagem formal que permita fazer cálculos ou simulações.

Tarefas são como as pessoas organizam e estruturam a informação e o conhecimento mediante a variabilidade do sistema alcançando objetivos de forma mais eficiente.

Através da análise das tarefas cognitivas, podemos compreender o raciocínio humano, identificando o desenvolvimento e a aplicação da lógica na realização de tarefas.

Segundo Rasmussen (1994), o comportamento humano em sistemas complexos, recorre ao controle da ação em três níveis de cognição:

Comportamento baseado em conhecimento. Ações controladas com o uso da consciência e raciocínios, onde se lida com a manipulação de símbolos para solução de problemas, o qual envolve a elaboração de estratégias para a ação, baseada em conceitos e conhecimentos adquiridos e modelos mentais;

Comportamento baseado em regras. Ações controladas por um conjunto de procedimentos, a serem adotados de forma consciente, baseados em regras previamente elaboradas sobre experiências anteriores, diante de signos associados às sub-rotinas, comuns aos sistemas;

Comportamento baseado em habilidades. Ações adaptativas, controladas de maneira quase que automática em situações rotineiras; tornam-se repetitivas, proporcionando o desenvolvimento de habilidades sem a necessidade de pensar em seus componentes, gerando um conhecimento tácito, apoiado em captação de sinais do ambiente do sistema.

Certos processos chave são úteis para descrever como os seres humanos pensam e tomam decisões para o controle dinâmico de ambientes como dos controladores de tráfego aéreo. Compreender as capacidades e limitações desses processos é importante para a compreensão das fontes de complexidade cognitiva.

Vários modelos de processos cognitivos têm sido desenvolvidos, incluindo alguns específicos sobre os controladores de tráfego aéreo. Outras abordagens de modelagem têm tentado construir simulações representativas do comportamento humano com base no processamento de informações de baixo nível e tomada de decisão.

Hansman (2008 b) propõe em sua teoria, que as tomadas de decisões em ambientes dinâmicos são ancoradas no conceito de consciência da situação (CS) (*Situation Awareness*) em sistemas dinâmicos.

Nessas situações, os operadores devem compreender o significado integrado do que estão procurando, à luz dos seus objetivos, percebendo uma situação complexa em três níveis:

- 1) Elementos no ambiente operacional dentro de um volume de tempo e espaço,
- 2) A compreensão de seus significados,
- 3) Projeção do estado futuro do sistema.

O nível 1 inclui perceber a presença de uma aeronave através de identificadores (computador, as rotas atuais, altitudes etc...), procedimentos de suporte à decisão como vigilância e comunicação, pedidos de comunicação de pilotos e de outros controladores, estar ciente das condições atuais do sistema e o impacto do tempo sobre a mudança dessas condições.

O nível 2 inclui compreender distâncias atuais entre aeronaves, e a consciência da exatidão das informações observadas como a posições das aeronaves, velocidades e altitude de voo.

O nível 3 é a consciência de projeção do estado futuro do sistema, como posições futuras das aeronaves nas distâncias resultantes entre aeronaves, as mudanças no clima e o potencial impacto das mudanças de rota.

As tarefas cognitivas são ativadas, a partir da recepção de estímulos sensoriais motores onde é instituída a organização psíquica da percepção. Nesse sentido, a percepção é compreendida como estrutura básica do ato de aprender, sendo o conhecimento que promove, com base nos dados recolhidos, a coordenação da conduta. Deste modo, através do processo cognitivo, é possível ao aluno realizar a diferença dos elementos que integram a realidade e das relações por ele reconhecidas, bem como a seleção dos elementos e das relações que serão retidas, permitindo-lhe construir mecanismos mentais de antecipação. Esses elementos da cognição humana constituem a realidade psíquica, ou seja, uma espécie de “realidade virtual”, um espaço psíquico de representações, reais ou não, no ponto de vista físico, mas reais no ponto de vista psicológico como uma espécie de simulação.

Woods *et al.* (2006 b) consideram que esse processo contempla a avaliação das condições de conhecimento, dinâmica atencional e estratégia que são as classes de fatores cognitivos que regem como as pessoas formam a intenção para a ação e como esses fatores podem levar a falha ou insegurança na execução da tarefa (Quadro 1).

Fatores Cognitivos	Implicações Cognitivas	Conflitos Presentes
Conhecimento	Conhecimento impreciso, simplificação, calibração do conhecimento, modelo mental.	Domínio incompleto do conhecimento, imperfeição, contradição.
Dinâmica Atencional	Percepção situacional, fixação.	Limite atencional demandado por múltiplas variáveis.
Estratégia	Tomada de decisão, determinação do risco.	Conflito de metas, regras que não se aplicam a todos os casos, questões organizacionais.

Quadro 1 - Fatores cognitivos determinantes da formação da ação

Operadores experientes tratam a tomada de decisão como um processo de resolução de problemas de ordem lógica. Eles usam elementos que conhecem para construir uma interpretação inicial da situação. A partir dessa representação mental, a possibilidade de ser plausível esta hipótese é verificada através da avaliação crítica a esses pressupostos. Quando confrontado com um problema complexo e desconhecido, operadores experientes recolhem e avaliam criticamente as evidências disponíveis, procuram estabelecer uma correlação entre elementos do conhecimento já fundamentado a fim de estabelecer uma coerência que leva a formulação de hipóteses a serem posteriormente testadas. Eles então tentam integrar os resultados desses processos em uma abrangente, plausível e coerente explicação lógica e a configuração da real situação do problema. A partir desse momento, eles avaliam os potenciais riscos e conseqüências associados ao possível decorrer da ação, recorrendo ao modelo cognitivo correlacional de suas experiências, ou através da testagem imaginária por meio de um processo de simulação mental de um novo modelo cognitivo.

Esses modelos de funcionamento mental servem de apoio à geração e manutenção da percepção da situação, bem como a decisão de vários processos de decisão e execução. Modelos de funcionamento mental são representações cognitivas do controlador do sistema, adequadas para as necessidades da tarefa atual.

Modelos de funcionamento mental podem recorrer a abstrações, ou versões simplificadas de um sistema dinâmico. Abstrações são um meio de representar as características essenciais de um modelo mental de uma forma mais compacta cognitivamente viável dentro das limitações de recursos humanos de memória e processamento.

Como Rasmussen (1994) afirma o processo de abstração não é apenas a remoção de detalhes de informações sobre as propriedades físicas ou materiais. Mais fundamentalmente a informação é acrescentada em instâncias superiores que regem a relação das várias funções ou elementos nos níveis mais baixos.

Sem a abstração, modelos mentais detalhados podem sobrecarregar a capacidade cognitiva de um controlador de atenção limitada (por exemplo, restrito a essa informação incluída no foco de atenção). Depois de usar a abstração para simplificar a parte do modelo mental, o controlador é capaz de atender a uma versão simplificada do sistema dentro desse modelo de funcionamento mental.

O modelo de funcionamento mental opera em um nível de abstração apropriado para a atual tarefa cognitiva. Ele incorpora os modelos dinâmicos utilizados para gerar as projeções necessárias para a tarefa atual. Um nível muito baixo de abstração ou muito detalhado, de uma representação da dinâmica da situação, pode tornar ineficaz o modelo de funcionamento mental. Em um nível elevado de abstração, detalhes importantes para o bom desempenho da tarefa podem ser perdidos.

O uso de abstrações reduz a sobrecarga na memória de trabalho utilizada para armazenar e manter representações do estado atual do ambiente operacional. A memória de trabalho é descrita por Hansman (2008 b) como um “bloco”, que mantém temporariamente informação verbal e espacial, que são os principais pontos que limitam a capacidade dos controladores de processar informações. Hansman (2008 b) aponta que enquanto há um grande debate em torno da definição capacidade, existe um consenso geral de que a capacidade da memória de trabalho pode ser compreendida como sendo o limite de quantidade de “blocos” que podem ser retidos. As abstrações fornecem um importante mecanismo para controlar o número de blocos e, assim, reduzir a demanda de recursos cognitivos de um operador de sistemas complexos.

Os seres humanos são hábeis em mudar as estratégias e abordagens para uma tarefa, a fim de minimizar o descompasso entre as demandas da tarefa, os recursos cognitivos e o desempenho mínimo. A complexidade cognitiva experimentada por um controlador não é um fator externo sobre a qual o controlador não tem controle, pelo contrário, a complexidade cognitiva é uma propriedade do modelo de trabalho do controlador, que reflete a representação mental da dinâmica da situação. Assim, existem vários mecanismos pelos quais um controlador pode controlar e reduzir a sua complexidade cognitiva.

2.5. Considerações Finais do Capítulo

Controladores podem reduzir a complexidade cognitiva através de alterações sobre como a dinâmica das situações são representadas no modelo de funcionamento mental. Alterar o nível de abstração da dinâmica da situação do tráfego aéreo pode ser usado para representar de forma mais simplificada o modelo mental de trabalho. Como sugerido pelas três principais características de complexidade identificadas anteriormente, abstrações (reduzindo o número de elementos no modelo de funcionamento mental) e as interconexões entre eles, fornecem mecanismos pelos quais os controladores podem reduzir a complexidade cognitiva da tarefa. A capacidade de representar situações mais compactas e cognitivamente simplificadas é um indicador da capacidade de controle do processo cognitivo. Hansman (2008 b) conclui que a maneira em que os indivíduos representam tarefas é considerada como uma das diferenças mais significativas entre os novatos e peritos.

Mudar o modelo mental permite os controladores adaptarem o seu esforço cognitivo para o mínimo de necessidades no desempenho da tarefa. Além disso, o reconhecimento de padrões de comportamento do sistema no início da estratégia da tomada de decisão discutida anteriormente fornece cognitivamente maneiras simples para identificar soluções rapidamente. O reconhecimento de padrões no início da tomada de decisão pode prover meios de simplificar o modelo de funcionamento mental e permitir a identificação de padrões e similaridade. Essa forma de aprendizagem é recorrente dos operadores mais experientes, que são capazes de confiar no reconhecimento de padrões para tomadas de decisões em uma situação sem uma detalhada, cuidadosa e cognitivamente intensiva projeção mental da situação. Controladores experientes categorizam problemas usando menos dimensões do que os novatos. Para Hansman (2008 b), os operadores experientes parecem ter um maior conhecimento sobre as propriedades relevantes da situação do tráfego aéreo.

Mudanças nas estratégias são formas pelas quais os controladores podem atenuar a complexidade cognitiva. Estratégias e técnicas são domínios ou abordagens específicas para executar uma tarefa. Um controlador desenvolve estratégias e técnicas ao longo do tempo de experiência e através de processos de formação. Estratégias e técnicas ajudam controladores a aumentarem a possibilidade de ações de comando.

Para a recriação virtual de cenários operacionais variáveis, é necessária a participação direta dos operadores mais experientes no processo de levantamento de requisitos a fim de identificar os principais elementos mobilizados por eles que os permitem abstrair seletivamente conceitos no reconhecimento de padrões dentro da variabilidade do ambiente operacional. Nesse levantamento, são mapeadas as metas, os atos perceptivos que constituem a identificação do cenário operacional, as regras de seleção adotadas por eles para a tomada de decisão e os métodos adotados por eles para atingirem as metas.

Partindo de todo o levantamento feito nesta pesquisa sobre o comportamento humano em sistemas complexos, observou-se necessário, utilizar como referência um método da literatura de IHC cujo enfoque fosse à participação direta dos experts no levantamento desses requisitos de interface é o método **GOMS** (*Goals, Operators, Methods, and Selection Rules*). Esse método foi desenvolvido no meio industrial e no contexto da concepção da Engenharia de IHC dentro do PARC (Palo Alto Research Center) da Xerox, por Card, Moran e Newel (1986). O objetivo é integrar os fatores humanos no estudo de sistemas complexos, permitindo definir modelos prescritivos da performance humana, considerando as atividades mentais que guiam o comportamento.

Este método pretende representar o comportamento dinâmico da interação com o computador, com base num modelo do comportamento humano que possui três subsistemas de interação: o **perceptual** (auditivo e visual), o **motor** (movimentos braço-mão-dedo e cabeça-olho), e **cognitivo** (tomadas de decisão e acesso a memória). Este comportamento é definido por:

- **metas** (*goals*), que podem ser decompostas numa hierarquia de sub-metas;
- **operadores** (*operators*), que são os atos perceptivos, motores ou cognitivos básicos que os usuários devem executar para afetar o ambiente da tarefa;
- **métodos** (*methods*), que são sequências de passos para se atingir uma meta; e
- **regras de seleção** (*selection rules*), expressões do tipo “condição-ação”, que devem ser utilizadas para selecionar um determinado método para atingir uma meta, sempre que houver mais de um método disponível para tanto. Esse item é de extrema relevância, onde é possível identificar junto aos operadores experientes, padrões de comportamento e resposta do sistema e indícios do ambiente.

Condução da análise de tarefas segundo o método GOMS (Card et al. 1986):

- 1. Faça a análise *top-down*.** Comece pelas metas mais gerais, e vá acrescentando detalhes em direção às mais específicas.
- 2. Usar termos gerais para descrever metas.** Não usar termos específicos de interfaces.
- 3. Examine todas as metas antes de subdividi-las.** Isto facilita o reuso de metas.
- 4. Considere todos os cenários de tarefas.** Utilize regras de seleção para representar alternativas.
- 5. Use sentenças simples para especificar as metas.** Estruturas complexas indicam a necessidade de decompor uma meta em sub-metas.
- 6. Retire os passos de um método que sejam operadores.** Os operadores são dependentes da interface e não são tratados no método GOMS simplificado.
- 7. Pare a decomposição no limite do design da interface.** A modelagem GOMS deve terminar quando as descrições estiverem tão detalhadas que os métodos sejam operadores ou envolvam pressuposições de *design*.

Para aplicações com múltiplas funções de usuários, deve-se observar algumas orientações específicas:

- Inicie especificando metas de alto nível para cada função de usuário.
- Se uma meta for compartilhada por mais de uma função de usuário, identifique estas funções de usuário ao definir a meta. Isto se torna desnecessário se a meta for compartilhada por todas as funções de usuário.

Capítulo 3 - Teorias de Aprendizagem

Neste capítulo é apresentado um histórico sobre as teorias de aprendizagem, e a relação entre elas.

Ao final do capítulo são feitas as considerações finais com a apresentação de passos para a elaboração de um software educacional utilizando *scaffolding* como parte da proposta desta pesquisa.

De um modo geral, teoria é uma construção humana para interpretar sistematicamente uma área de conhecimento. Uma teoria de aprendizagem é uma construção humana para descrever a forma pela qual uma pessoa aprende, levando-nos a compreender o processo da aprendizagem.

De acordo com Moreira (1999), é possível identificar três correntes ou teorias básicas de aprendizado: a comportamentalista ou behaviorista, a gestáltica ou cognitivista e a metacognitiva.

Para a teoria comportamentalista, a aprendizagem concentra-se no comportamento observável, considerando a relação entre os estímulos e as respostas subseqüentes por parte do indivíduo. A teoria cognitivista argumenta que a "caixa preta" da mente deve ser aberta e entendida, enfatizando processos mentais dificilmente observáveis e a possibilidade do aluno adquirir e organizar informações. A teoria construtivista defende o indivíduo como elaborador e criador do conhecimento, instaurando a valorização do agir de quem aprende como elemento central para o aprendizado (Moreira, 1999).

Contudo, como pode ocorrer em qualquer categorização deste tipo, as divisões são um pouco arbitrárias, permitindo adições e subdivisões para as abordagens descritas a seguir.

3.1. Abordagem Comportamental

Esta abordagem se caracteriza pelo estudo do comportamento (*behavior*, em inglês), tomado como um conjunto de reações dos organismos aos estímulos externos. O princípio do behaviorismo é que só é possível teorizar e agir sobre o que é cientificamente observável.

Para Moreira (1999) as teorias comportamentalistas clássicas compartilham pelo menos três características: 1) buscam uma objetividade impecável; 2) explicam o comportamento em termos de associações estímulo-resposta elementares; 3) fazem poucas referências às intenções do comportamento, exceto no que se refere a necessidades e impulsos específicos.

Dentre os teóricos que contribuíram para a fundamentação da teoria comportamental, destacam-se: Pavlov, Watson e Skinner. Seus estudos estavam direcionados para a definição das leis que relacionam estímulos, respostas e conseqüências, sugerindo que o ambiente desempenha as funções anteriormente atribuídas a sentimentos, e estados internos do organismo, introspectivamente observados.

Nick (2006) resume a que se propõem as diversas correntes comportamentais:

(1) Limitar os estudos do comportamento às atividades diretamente relacionadas com o processo psíquico (metalismo); (2) Estudar as atitudes do organismo como um todo (unitarismo); (3) Estudar o comportamento não como quadro de referência do próprio organismo, mas como resposta global, isto é, em contraste com as atividades fisiológicas como respostas parciais; (4) Estudar o comportamento como um novo evento (emergentismo) que é produzido por integração das atividades fisiológicas ou certa Gestalt no funcionamento orgânico; (5) Proceder apenas à análise daqueles eventos que têm origem claramente psicológica ou fisiológica.

Com o passar do tempo o behaviorismo foi sendo modificado e, hoje, já não se entende o comportamento como uma ação isolada do sujeito, mas uma interação entre o ambiente (estímulos) e o sujeito (respostas). Assim, embora muito baseadas em experimentos com animais, a abordagem comportamental é considerada a precursora das teorias psicológicas sobre o comportamento humano, gerando aplicações extremamente úteis no ensino em sala de aula, fortemente influenciado por seus princípios nas décadas de 60 e 70 (Moreira,1999).

Qualquer estratégia educacional com base nas teorias comportamentais deve considerar a preocupação científica que a caracteriza, aplicando-a no planejamento, na condução, implementação e avaliação do processo de aprendizagem.

3.2. Abordagem Cognitiva

Piaget

A proposta de Piaget (1987), denominada epistemologia genética, segue a linha cognitivista e parte do estudo crítico do conhecimento científico e do conhecimento que depende das trocas entre o ser humano e o meio em que vive. Esse conhecimento é construído, a partir da sua própria ação e no modo que esta se converte em processo de construção interna (ou tradução do mundo exterior formando uma estrutura em contínua expansão) sendo o ser humano agente de seu desenvolvimento.

Em sua teoria da equilibração, Piaget (1987) percebeu que todo esquema de assimilação tende a incorporar elementos que são exteriores e compatíveis com sua natureza. Assim ele concluiu que todo esquema de assimilação é obrigado a se acomodar aos elementos que assimila,

se modificando assim em função de suas particularidades, mas, mantendo continuidade de sua evolução, assim como o conhecimento assimilado anteriormente.

Piaget (1987) parte do princípio que existe continuidade entre os processos biológicos de morfogênese e adaptação ao meio ambiente e a inteligência. Esse processo de criação e modificação do modelo cognitivo individual em que cada ser humano formula a representação do meio em que está inserido é constituído de esquemas que incluem segundo Piaget (1987), dois tipos de ações para a sua composição: a assimilação e a acomodação.

A assimilação ocorre quando é incorporado um novo conceito ou idéia ao que já é previamente conhecido. Quando os seres humanos se deparam com um problema, examinam o repertório de esquemas para resolvê-lo. Adaptar os estímulos ambientais ao conhecimento que já possuímos requer que repensemos o evento de forma correspondente ao padrão encontrado que faz parte desse repertório. Segundo Piaget (1987), assimilar um objeto a um esquema significa conferir a ele um ou vários sentidos.

A acomodação, por sua vez, ocorre quando há uma transformação sofrida pelo organismo e a necessidade em se adaptar para lidar com o ambiente. Assim, diante de um novo conceito ou idéia os seres humanos modificam seus esquemas cognitivos, estruturados anteriormente, na tentativa de adaptar-se à nova situação.

Esse processo constitui a chamada equilíbrio majorante e são oriundos de dois elementos básicos filogenéticos do desenvolvimento humano: os fatores invariantes e os fatores variantes.

Os fatores invariantes derivam da necessidade inata do ser humano de prefixar elementos de sua experiência que direcionarão seu comportamento. Piaget (1987) indica que, ao nascer, o indivíduo recebe como herança uma série de estruturas biológicas, sensoriais e neurológicas, que permanecem constantes ao longo da sua vida. São essas estruturas biológicas que irão predispor o surgimento de certas estruturas mentais. Em vista disso, na linha piagetiana, considera que o indivíduo carrega consigo duas capacidades inatas como tendência natural, que são à organização e à adaptação sendo esses os elementos motivadores do comportamento humano.

Os fatores variantes são representados pelo conceito de esquema que constitui a unidade básica de pensamento e ação estrutural do modelo piagetiano, sendo um elemento que se transforma no processo de interação com o meio, visando à adaptação do indivíduo ao real que o circunda. Com isso, a teoria psicogenética deixa à mostra que a inteligência não é herdada, mas

sim que ela é construída no processo interativo entre o homem e o meio ambiente (físico e social) em que ele estiver inserido.

Essa relação dinâmica entre fatores variantes e invariantes, regulada por um processo de equilíbrio majorante proporciona uma sucessão de desequilíbrios e reequilíbrios na passagem de estados entre a aquisição de novos conhecimentos e a relação entre conhecimentos já previamente fixados. Esse processo de reequilíbrio, portanto, seria a real fonte do desenvolvimento cognitivo.

Assim como a Epistemologia Genética proposta por Piaget (1977); a Teoria da Complexidade proposta por Morin (2005) converge na definição de que o indivíduo é um sistema biológico interagindo com outros sistemas no meio exterior a partir de trocas constantes que provocam mutuamente transformações adaptando-se a novas configurações. Sob a ótica de ambas as linhas teóricas é possível convergir a definição dessa relação tanto como descrito por Morin (2005) quanto para Piaget, ao referir-se sobre o equilíbrio dos sistemas cognitivos. Segundo ambos os autores, há um desequilíbrio no fluxo energético que alimenta os sistemas e que, sem este fluxo, haveria desordem organizacional levando rapidamente à ruptura do sistema. Portanto, esse desequilíbrio que retroalimenta o sistema e o mantém em funcionamento e aparentemente estável, só se deteriorará se o sistema se fechar provocando a ruptura dessas relações.

Assim é possível concluir, que as explicações sobre o funcionamento do sistema, emergem da dinâmica da configuração dessas relações e, portanto, a aprendizagem sobre sistemas complexos só poderia ser promovida, a partir da representação dessas relações.

Ambas às teorias convergem com relação às formas de interação do sistema cognitivo com o meio interno e externo, e também em relação à importância do processo de equilíbrio, onde os desequilíbrios atuam como força motriz dos avanços cognitivos. Piaget (1977) aponta que na perspectiva da equilíbrio deve procurar-se nos desequilíbrios uma das fontes de progresso no desenvolvimento dos conhecimentos. Para Piaget (1977), só os desequilíbrios obrigam o sujeito a ultrapassar o seu estado atual e procurar seja o que for em direções novas.

Para Morin (2005), duas consequências capitais decorrem da ideia de sistema aberto. A primeira é que leis de organização da vida não são de equilíbrio, mas de desequilíbrio, recuperado ou compensado, de dinamismo estabilizado. A segunda consequência, talvez ainda maior, é que a inteligibilidade do sistema deve ser encontrada, não apenas no próprio sistema, mas também na

sua relação com o meio ambiente, e que esta relação não é uma simples dependência, ela é constitutiva do sistema.

Piaget (1995) teorizou ser possível construir conhecimento por operações concretas antes das operações formais e que, portanto, o conhecimento não é uma simples cópia da realidade. Para conhecer um objeto, é preciso agir sobre ele, transformá-lo e compreender este processo.

Para Piaget (1995), o conhecimento tem origem na atividade do sujeito sobre o meio e não apenas nas propriedades objetivas da realidade. Tal conhecimento não resulta da abstração das propriedades do objeto, mas sim, das propriedades que a ação do sujeito introduz nos objetos, ou seja, da abstração reflexionante.

O conceito de abstração tem um papel central na teoria de Piaget (1995) e refere-se às propriedades introduzidas a partir da ação do sujeito em um objeto. Com interiorização dessas operações, elas passariam a ser entendidas e executadas simbolicamente pelo sujeito de forma dedutiva. Diferentemente da abstração empírica que leva apenas à constatação a partir do que é observado das propriedades da natureza do objeto. A abstração reflexionante atinge maior profundidade e leva à compreensão. Na abstração reflexionante, construção e reflexão atuam juntas formando a compreensão, e assim, projetando estruturas comportamentais e cognitivas a nível superior, de forma consciente.

Vygotsky

A teoria de Vygotsky (1991) também converge com as demais teorias cognitivistas no que se refere às relações mediadas por instrumentos. Segundo Vygotsky (1991), as relações sociais são convertidas em funções mentais através de mediadores como instrumentos e signos. Instrumentos e signos são construções sócio-históricas e culturais que são apropriadas através da interação social. Ou seja, quanto mais o sujeito usa signos e instrumentos, mais se modificam as operações psicológicas que ele é capaz. Para internalizar signos, o ser humano precisa captar significados já compartilhados socialmente e certificar-se de que os significados que capta são aqueles compartilhados para os signos em questão.

Para Vygotsky (1991), a linguagem é um importante sistema de signos para o desenvolvimento dos seres humanos, porque a libera dos vínculos contextuais imediatos. O desenvolvimento dos processos mentais superiores depende de descontextualização e a

linguagem serve muito bem para isso, na medida em que o uso de signos lingüísticos permite o afastamento cada vez maior de um contexto concreto, flexibilizando o pensamento conceitual e proposicional.

Para Vygotsky (1991), o desenvolvimento das funções mentais superiores requer a internalização de instrumentos e signos em contextos de interação e a aprendizagem se converte na condição principal para o desenvolvimento dessas funções. Para medir a capacidade cognitiva do indivíduo real e a sua capacidade potencial, Vygotsky (1991) desenvolveu o termo que fundamenta sua teoria ao qual classificou de zona de desenvolvimento proximal, que define as funções que ainda não amadureceram, mas que estão em processo de maturação. Vygotsky (1991) define assim três fases no processo de formação de conceitos: Agregação desorganizada ou amontoado, pensamento por complexos, conceitos potenciais.

O estabelecimento desta zona permitirá a um mediador (pessoa detentora de um conhecimento superior, reconhecido pelo aluno) prever o acesso a atividades que exercitem o que está latente. A mediação ocorrendo fora da ZDP, não produziria nenhum desenvolvimento, pois, ou o aluno já sabe o que está sendo proposto pelo mediador, ou ainda não é capaz de entender o que o mediador está sugerindo, sendo, portanto, sua determinação fundamental no processo de aprendizagem.

Ausubel

Através das pesquisas de Ausubel (2003), é possível obter uma primeira visão da importância da mediação na transição entre o processo cognitivo e o metacognitivo e como este último pode ser estimulado. Em suas pesquisas, ele identificou que o conhecimento prévio do aluno interage de forma significativa com o novo conhecimento a ser adquirido provocando mudanças na estrutura cognitiva já existente no aluno. Essa mudança ou processo de assimilação é designado por Ausubel (2003) como Aprendizagem Significativa.

Em sua teoria, a Aprendizagem Significativa, é provida do conjunto de resultados de experiências de aprendizagem de um aluno e está organizado de forma hierarquizada em sua estrutura cognitiva. Nesse processo de interação entre o novo material e a estrutura cognitiva já existente, promove a mudança na estrutura cognitiva do aluno, caracterizando assim a

assimilação de novos significados através da correlação com antigos significados já internalizados, ocorrendo assim a aprendizagem.

Ausubel (2003) em sua teoria aponta princípios facilitadores e seqüenciais e construção do conhecimento. Para ele, a aprendizagem ocorre primeiramente com a diferenciação progressiva de conceitos e significados com a apresentação das idéias mais gerais e centrais do conteúdo logo no início, para que o aluno possa identificar as diferenças entre o que ele já sabe e o que ele deve aprender. Posteriormente, o aluno reconhece novas relações correlacionando o novo e antigo conhecimento, novos e antigos conceitos e significados na reconciliação integradora de aparentes inconsistências, criando um nível de ordenação entre conceitos. A internalização desses conceitos ocorre em fases distintas e podem ser facilitadas pedagogicamente pelo professor com a organização seqüencial de relações, ocorrendo assim a consolidação de um novo conhecimento.

Para Ausubel (2003), a organização dos conceitos na estrutura cognitiva dá-se de três maneiras: aprendizagem subordinada, aprendizagem superordenada e aprendizagem combinatória.

- **Subordinada** – Acontece quando a nova idéia é um exemplo, uma especificação de algo que já se sabe.
- **Superordenada** – Ocorre quando a nova idéia que se aprende é mais geral do que uma ou um conjunto de idéias que já se sabe
- **Combinatória** – É a aprendizagem de uma nova informação, potencialmente significativa, relacionada à estrutura cognitiva como um todo e não com aspectos específicos dessa estrutura como ocorre nas aprendizagens subordinada e superordenada.

Para facilitar a aprendizagem significativa, o papel do professor envolve os seguintes cuidados fundamentais:

- **Fase 1 (Estágio inicial)** – Conceitos previamente formados e respectivas ligações à estrutura cognitiva do aluno.
- **Fase 2 (Apresentação do novo conceito)** – Assimilação do novo conhecimento à estrutura cognitiva do aluno;
- **Fase 3 (Diferenciação progressiva)** – Ancoragem do novo conhecimento ao conhecimento pré-existente e já internalizado na estrutura cognitiva do aluno;

- **Fase 4 (Reconciliação integrativa)** – Reorganização da estrutura cognitiva do aluno e internalização de novos conceitos configurando a aprendizagem significativa.

É possível perceber que ao longo das fases a construção do conhecimento, a Aprendizagem Significativa é apoiada no que Ausubel (2003) identificou como *organizadores prévios*, para ligar o que o aluno já sabe e o que ele deveria saber para que esse conteúdo mostre a relação do novo com o conhecimento prévio.

“Estes organizadores são apresentados com antecedência ao aprendizado por si mesmo, e também são apresentados em um nível mais elevado de abstração, generalidade e abrangência. E como o conteúdo substantivo de um dado organizador, ou de uma série de organizadores, é selecionado com base na sua conveniência para explicar, integrar e inter-relacionar o material que eles precedem, esta estratégia satisfaz simultaneamente o substantivo, assim como o critério de programação para aumentar a força organizacional da estrutura cognitiva.” (Ausubel, 2003).

Ausubel (2003) para representar a função desses organizadores prévios ou subsunções dentro Aprendizagem Significativa, os relaciona metaforicamente a estruturas de apoio utilizadas na construção civil, chamadas andaimes ou *scaffoldings*, para demonstrar que desequilíbrio cognitivo no aluno pode ser provocado, empregando material diferenciado:

“(...) andaimavam a aquisição de material mais diferenciado (...) e incrementavam a discriminação entre os novos conceitos e os similares ou em conflito na estrutura cognitiva.” (Ausubel, 2003).

Para Ausubel (2003), os *scaffoldings* são andaimes intelectuais que servem de apoio à formação da estrutura cognitiva do aluno, servindo de ancoragem para o novo material.

O *scaffolding* possui duas dimensões determinadas por sua representação metafórica estabelecida, composta por extensão e altura. A extensão é definida pela diversidade de métodos de ensino utilizados pelo professor na tentativa de facilitar para o aluno a aprendizagem de um determinado conteúdo através de organizadores prévios e elementos subsunções já existentes na estrutura cognitiva do aluno que permitem conectar o conhecimento prévio ao novo conhecimento. Já a altura é estabelecida pelo nível de abstração ou elevação do grau de

dificuldade de aprendizagem do conteúdo estabelecida de forma sequencial ao longo do processo, até que o aluno atinja o nível máximo de proficiência, de uma determinada área de conhecimento.

3.3. Abordagem Metacognitiva

A metacognição se constitui no uso reflexivo da cognição, onde através da prática do controle desse processo, os seres humanos são capazes de estruturar a lógica e assim instituir o nexos recursivo como o caminho reverso ao da construção do conhecimento. Utilizando o próprio conceito para compreender a metacognição, é necessário recorrer ao nível anterior de significado até a definição de metaproceto que se configura na capacidade de controlar de forma arbitrária e generalizada através da metalinguagem, a regra de significação que estrutura um conhecimento. Retroagindo um nível derivativo a mais da palavra, está o significado de metalinguagem que se caracteriza em uma linguagem ou forma própria do aluno descrever as outras linguagens ou sistemas de significação de acordo com seu mecanismo cognitivo. Portanto, as capacidades de controle reflexivo e consciente de uma regra identificada através da descrição da composição dos significados que a compõem caracterizam a metacognição. Vista como cerne da capacitação para a resposta adaptativa, a metacognição é aplicável a todos os níveis do processo comportamental.

A metacognição fornece uma visão altamente dinâmica da tomada de decisão. Os problemas expostos pelo pensamento crítico permitem levar a correção de etapas, que envolvem a modificação e elaboração de modelos e planejamento das situações. O pensamento crítico pode levar à criação e à detecção de outros problemas, desencadeando novos ciclos de correção e avaliação. O processo para quando o teste dinâmico conclui que o modelo ou plano é satisfatório, ou que os custos de uma maior reflexão superam os potenciais benefícios.

A partir do pensamento crítico, que levanta as questões de forma antecipatória, o ser humano seria capaz de responder através do processo de recursão, estabelecendo a chamada metacognição. A metacognição é considerada como a capacidade humana de tomada de consciência de processos e das competências para desenvolvê-los, de forma controlada ou autorregulada, onde o indivíduo deve ser capaz de avaliar a execução de tarefas e fazer correções quando necessário. A metacognição refere-se não somente a consciência reflexiva do ser humano sobre seus processos cognitivos, mas também na capacidade de controlá-los.

Flavell

Flavell (2001), baseando-se nos trabalhos de outros teóricos principalmente Piaget e Ausubel, procura focar seus estudos sobre a possibilidade de reversão do processo cognitivo, identificado como sendo característico do processo de desenvolvimento cognitivo a partir do período operacional concreto.

Flavell (2001) fundamenta a metacognição a partir da metamemória, termo em que ele define como um sistema de desenvolvimento que leva em conta a sensibilidade e o conhecimento das variáveis da pessoa, da tarefa e da estratégia.

A variável das pessoas refere-se ao conhecimento sobre si mesmo, as diferenças entre si e os outros e o conhecimento dominante de uma cultura.

A variável tarefa trata do conhecimento sobre a natureza da informação e sobre os critérios da tarefa a realizar.

Em termos de estratégia são tratadas as informações sobre os meios, processos ou ações que permitem ao sujeito atingir os objetivos com maior eficácia numa determinada tarefa.

Estas três variáveis apresentadas interagem entre si e o modelo global de controle cognitivo contendo os aspectos do conhecimento metacognitivo, as experiências metacognitivas, os objetivos e as ações (ou estratégias).

As experiências metacognitivas consistem em impressões ou percepções conscientes que podem ocorrer antes, durante ou após a realização de uma tarefa. Geralmente relacionam-se com a percepção do grau de sucesso em que se pode alcançar na realização de uma tarefa.

Já com relação às ações que derivam do propósito definido por objetivos, implícitos ou explícitos, impulsionam e alimentam o processo cognitivo, que traduzidos em comportamentos correspondem às estratégias utilizadas para potencializar e avaliar o progresso cognitivo. Para Flavell (2001), enquanto as estratégias cognitivas podem levar o sujeito a um objetivo cognitivo, as estratégias metacognitivas propõem-se a avaliar a eficácia dessas estratégias traçadas.

Seminério

Seminério (1984, 1985), fundamenta sua teoria propondo que através do estudo dos passos filogenéticos que constituem a estrutura cognitiva dos seres humanos é possível compreender a função organizadora da informação captada do ambiente pelos nossos sentidos mais

desenvolvidos. Essa função se divide em dois planos, o audiofonético e o visiomotor, onde através dos três primeiros níveis de linguagem, os *inputs* e signos-significantes são construídos, constituindo o que é definido por Seminério (1984, 1985) como quarto nível que compõe a lógica reguladora da estrutura de todos os níveis anteriores de linguagem.

Para Seminério (1985), o conhecimento é constituído por níveis de linguagem onde a partir do estudo da estrutura de cada nível de linguagem é possível observar que a organização e a captura do conhecimento se dão com o encapsulamento do conteúdo através da forma, onde signos-significantes são construídos a cada nível de linguagem (L1, L2, L3 e L4) formando invariâncias capazes de serem identificadas, controladas e reaplicadas em forma de regras instituindo no aluno o processo metacognitivo.

Através do estudo dos passos filogenéticos que constituem a estrutura cognitiva dos seres humanos é possível compreender a função organizadora da informação captada do ambiente pelos nossos sentidos mais desenvolvidos. Essa função se divide em dois planos, o audiofonético e o visiomotor, onde através dos três primeiros níveis de linguagem, os *inputs* e signos-significantes são construídos, constituindo o que é definido por Seminério (1985) como quarto nível que compõe a lógica reguladora da estrutura de todos os níveis anteriores.

A linguagem e principalmente a metacognição se estrutura em quatro níveis:

L1 - Linguagem mais elementar e arcaica do processo cognitivo, que utiliza os *inputs* psicofisiológicos da visão e cinestesia geral, bem como da audição e cinestesia da fonação, como signos-significantes para organizar em conjuntos combinatórios seus contextos significativos. Trata-se da capacidade inata da percepção.

L2 - Linguagem que utiliza os *inputs* das estruturas figurais decorrentes da linguagem anterior, como signos-significantes para organizar em conjuntos combinatórios seus contextos mais significativos. Representa a possibilidade associativa de se acoplar a uma estrutura figural presente um sentido experimental, ou seja, associar formas L1 a modos virtuais de ação elementar, bem como às formas de atribuição de significado empírico. Trata-se da capacidade de criar significados a partir da percepção.

L3 - Linguagem que utiliza os *inputs* das figuras estruturais decorrentes da linguagem anterior, como signos-significantes aptos a formar com eles contextos significativos organizados com sentido episódico: em termos imaginários no canal visomotor, em termos de frases discursivas no canal

audiofonético. Trata-se da capacidade de criar sentido a partir de significados e assim construir o raciocínio.

L4 - Linguagem que utiliza os *inputs* decorrentes de todas as linguagens anteriores notadamente a partir da L3 como signos-significantes, aptos a formar invariâncias captadas como regras controláveis e reversivelmente reaplicáveis às suas fontes. Constitui em si a atividade metaprocessual que permite desenvolver controle consciente sobre qualquer processo cognitivo, incluindo a reflexividade da regra por ele gerada, tornando-se assim, fonte para estruturação de qualquer tipo de lógica. Trata-se da capacidade de controlar de forma consciente significados, sentidos e raciocínio constituindo a lógica.

Nesse modelo, o aluno é treinado à compreensão da existência de regras ou invariantes, em quaisquer situações tornando implícita a capacidade de descobri-las e transferi-las do inconsciente para a consciência. Esse nível de identificação e consciência das regras de controle dos sistema pré-estabelecidas através da elaboração dirigida e percebidas pelo aluno como um conjunto de invariantes pode ser decodificada através das estruturas de linguagem que fornecem uma via de acesso aos diferentes mecanismos cognitivos com os quais cada aluno é dotado.

Para nivelar a capacidade de aprendizagem entre os diferentes níveis de cognição é preciso oferecer ao aluno estruturas de linguagem mais acessíveis ao seu nível cognitivo. Essas estruturas devem ser capazes de proporcionar, a partir do nível mais alto de linguagem em que está codificado o aprendizado, a desconstrução dos signos-significantes estabelecidos em linguagens anteriores. Isso ocorre até que se encontre um nível de invariância ao qual ele possa estabelecer uma correlação de significados, seja entre o nível e estrutura de linguagem identificada com sua capacidade cognitiva, seja entre o nível e estrutura de linguagem de nível mais alto em que se encontra codificado o conhecimento.

Baseado nesse processo de desconstrução de signos-significantes, o chamado salto metacognitivo permite ao aluno identificar a regra que proporciona a correlação entre esses níveis de significados, responsáveis pela estruturação do conhecimento, constituindo a capacidade de elaborar uma explicação para realidade percebida e transformada.

Essa perspectiva hierarquizada no processo cognitivo se traduz, a cada salto de linguagem, na redução fenomenológica de sentido, reproduzindo a maneira como os operadores experientes desenvolvem a capacidade de abstração. Quando o conhecimento é estruturado, reduz-se o que é

captado do ambiente pelos sentidos em informação. Em seguida, essa informação é transformada em significação e que por sua vez, compõem hierarquicamente as mensagens construídas ao longo das linguagens superpostas. Assim, toda a realidade passa a ter uma existência puramente fenomenal no sentido informacional. Esse reducionismo só é percebido quando há uma variação que nos faz retroagir cognitivamente um ou mais níveis de linguagem (L1, L2, L3 e L4), em busca de invariantes pré-fixados em todas as linguagens anteriores, capazes de conectar o novo conhecimento que surge com os significados anteriormente estabelecidos e fundamentados, possibilitando uma reversibilidade reflexiva para a identificação e construção de uma regra, caracterizando o aprendizado.

Ao invés de esperar que o aluno realize, através da própria ação, as assimilações necessárias a provocar a acomodação e, subseqüentemente, a determinar equilíbrios como preconizado por Piaget, Seminério aponta ser viável estimular a metacognição postulada por Flavell, através não só da equilíbrio de Piaget, mas também associado a um sistema de códigos como estudado por Vygotsky. Assim, seria possível fornecer um modelo lógico, para que possa ser elaborada pelo aluno como uma autêntica modelação, através da imitação provocada, a idéia de observação e reprodução aperfeiçoada caracterizada na teoria de Vygotsky. Assim, um procedimento pedagógico de elaboração dirigida, segundo Seminério deveria permitir a passagem de regras potenciais do inconsciente cognitivo para a consciência.

Bruner

Assim como Seminério, Bruner (*apud.* Azevedo 2005 a, 2005 b, 2008) faz uma crítica ao modelo educacional, ao observar que eles tendem a negar a capacidade dos próprios alunos “construírem andaimes” (*scaffoldings*).

A partir do conceito concebido por Ausubel, Bruner (*apud.* Azevedo 2005 a, 2005 b, 2008) define que o *scaffolding* se caracteriza pela progressiva diminuição da regulação realizada pelo tutor, na medida em que o aluno começa a captar o significado funcional das ações a serem realizadas.

O *scaffolding* é considerado um “andaime” que o professor ou alguém na função de orientador do processo de aprendizagem fornece ao discurso que o aluno constrói e que progressivamente deve ser desmontado, à medida em que ele começa a ter condições de dominar

aquelas modalidades de raciocínio. Assim como Seminério, Bruner (*apud.* Azevedo 2005 a, 2005 b, 2008) aponta que o conhecimento é estruturado em níveis de linguagem e como Ausubel, ele acredita que eles são elementos subsunçores de apoio à construção da estrutura cognitiva do aluno.

Nesse sentido, o ensino dialógico surge como uma valiosa ferramenta no processo de aquisição de habilidades necessárias para a percepção da organização e o controle de um sistema complexo. Esse modelo proporciona um maior benefício ao aluno ao depositar maior ênfase a interação produzida em meio aos contextos de produção do conhecimento.

O ensino dialógico relaciona-se com outras perspectivas teóricas que também levam em conta o papel fundamental da interação para os processos de produção do conhecimento. Tais perspectivas defendem a idéia de participação guiada em meio aos processos de ensino e aprendizagem, orientando-se pela concepção de *scaffolding*, no que diz respeito ao papel do professor que encontra-se diante de uma intervenção pedagógica voltada para a construção guiada do conhecimento.

Tal processo também pode ser compreendido de acordo com as proposições de Vygotsky em relação a sua teoria de zona de desenvolvimento proximal onde ocorre a negociação social dos significados em que os alunos apropriam-se das ações e interpretações geradas através da interação com o ambiente.

É nesse espaço que se pode estabelecer o vínculo dos alunos com a interatividade, convergindo no plano do funcionamento interpsicológico. A possibilidade de uma compreensão compartilhada de uma tarefa depende da definição da situação, ou seja, o modo pelo qual o ambiente é representado por aqueles que ali operam. Sendo a representação ativamente construída por cada aluno, ela é também diferente para cada um, assim o objetivo da interação e da instrução é atingir uma “redefinição compartilhada da situação” por meio de níveis progressivamente maiores de intersubjetividade.

Bruner (1986) assim aponta em sua obra “*Vygotsky's zone of proximal development: The hidden agenda*” aponta a retirada progressiva dos *scaffoldings* como forma de estimular a autonomia do aluno no processo de aprendizagem onde o conhecimento é construído através da dedução lógica da relação entre o modelo referencial apresentado ao aluno e o novo cenário.

Apoiado no trabalho de Vygotsky, Bruner (1986) indica que o aluno deve ser avaliado constantemente pelo professor, verificando os dois níveis do desenvolvimento cognitivo, o real e

potencial. Bruner (1986) orienta que ao constatar que o aluno atingiu a zona de desenvolvimento real, o scaffolding deve ser retirado, incentivando a ação autônoma do aluno. Sem os scaffoldings, o aluno seria capaz de restabelecer a conexão entre esses cenários de aprendizagem com a identificação dos elementos invariantes.

3.4. Considerações Finais do Capítulo

Os invariantes desempenham função de destaque no processo de aprendizagem permitindo aos indivíduos reconhecer os elementos relevantes em uma dada situação, bem como definir o curso da ação a ser realizada.

Um invariante pode ser explícito ou implícito, dependendo do sujeito ter ou não consciência de sua existência no momento em que faz uso dele. Os invariantes explícitos organizam as ações do usuário, podendo também ser expressos através do uso de representações simbólicas ou palavras. Os invariantes implícitos também estão presentes na ação do sujeito, mas pelo fato de serem conhecimentos tácitos não podem ser compartilhados e discutidos por um aluno em seu ambiente de ensino.

Os invariantes desempenham um papel chave também na avaliação do aprendizado. A inferência dos invariantes presentes no raciocínio de um indivíduo pode ser realizada a partir da análise das ações executadas pelo mesmo durante a realização de uma atividade educacional. Essa análise permite avaliar os esquemas que o aluno desenvolveu para lidar com situações relacionadas a um campo conceitual específico.

O conhecimento do conjunto de situações e conceitos necessários ao aprendizado facilita o desenvolvimento de interfaces educativas na medida em que traz à tona componentes necessários ao aprendizado desse conceito. O conhecimento desses componentes permite ao desenvolvedor vislumbrar as necessidades e possibilidades relativas ao aprendizado de um conceito que deverão ser atendidas através da interface.

Os invariantes identificam as propriedades de um conceito. Conhecer os invariantes relacionados a um conceito viabiliza a apreensão por parte do aluno do significado do mesmo. Sob essa perspectiva, fica clara a importância da interface e das funcionalidades em um software educacional que deve desempenhar a função de mediador e então favorecer a percepção dos invariantes relacionados aos conceitos tratados no software, necessários para o aprendizado.

Esses mesmos invariantes devem ser levados em consideração no momento de avaliar a aprendizagem decorrente da utilização da interface. A utilização de *scaffolding* em um software educacional requer a compreensão do papel a ser desempenhado pelo software enquanto mediador, não só apoiando como também provocando o aluno no processo de reequilíbrio para construção do conhecimento.

Uma forma de adequar o *scaffold* utilizado em um software educacional às necessidades dos alunos é investigar o processo de execução das tarefas por eles realizadas. A partir do registro dos passos, procedimentos, observações e dificuldades apresentadas pelos alunos durante a execução da tarefa é possível formular uma ajuda adequada ao aprendizado do conteúdo em questão. Outras fontes de inspiração a serem consideradas no projeto do *scaffolding* podem ser encontradas no próprio conteúdo a ser veiculado na interface, através da identificação dos conceitos e invariantes necessários à aprendizagem do assunto bem como na análise das estratégias bem sucedidas aplicadas pelos instrutores em sala de aula ou a partir do relato de *experts* a partir de suas experiências reais. Uma vez implementado, o *scaffolding* pode ser avaliado a partir da observação dos usuários utilizando a interface educativa.

A partir do que foi exposto, é possível apontar os seguintes passos para a elaboração de um software educacional utilizando *scaffolding*:

- Definir o campo conceitual a ser veiculado na interface;
- Identificar os conceitos necessários ao aprendizado deste campo conceitual;
- Identificar as situações que tornam esses conceitos significativos e oferecer essas situações ao usuário;
- Definir as diferentes representações que serão utilizadas para lidar com essas situações e permitir ao usuário fazer uso delas durante o uso da interface;
- Definir se existirá conexão entre algumas das representações utilizadas e de que forma o usuário poderá utilizar estas conexões para passar informações de uma representação para outra;
- Identificar os invariantes que devem ser capturados pelos usuários ao lidar com essas situações;

- Projetar o *scaffolding* (ajuda para o usuário) a ser utilizado na interface de forma a facilitar a percepção desses invariantes bem como auxiliar o usuário no processo de resolução de problemas com a interface;
- Desenvolver um protótipo para realização de testes com usuários;
- Avaliar os invariantes mobilizados pelo usuário ao utilizar o protótipo bem como a eficácia do *scaffolding* em auxiliar o aluno durante o uso da interface educativa.

Capítulo 4 - Engenharia de Software e Aplicativos Educativos

Neste capítulo é feita uma revisão bibliográfica com a apresentação de conceitos sobre Engenharia de Software e a apresentação das principais metodologias de desenvolvimento ágil. Também são apresentados conceitos de Interface Humano-Computador e suas principais metodologias de desenvolvimento. Além disso, é definido o que são Objetos de Aprendizagem e suas características como software educacional.

Ao final do capítulo é apresentada uma análise comparativa entre as disciplinas, cujo resultado é apresentado como parte de uma metodologia integrada de desenvolvimento de software proposta nesta pesquisa.

A Engenharia de Software é definida por Pressman (2001) como a aplicação de um processo disciplinado, sistemático e quantitativo para desenvolvimento, operação e manutenção de um software. Em uma tradução livre e mais contemporânea, a definição de Engenharia de Software talvez possa ser definida como sendo um conjunto de processos, técnicas e ferramentas que têm por objetivo apoiar o processo de implementação e operação de um software. A Engenharia de Software é, então, a aplicação de conhecimento científico para criação e manutenção de um software, o que envolve a aplicação de processos, métodos e ferramentas.

Os processos se constituem na ligação entre os métodos e as ferramentas, garantindo que haja uma seqüência em que os métodos sejam aplicados e que os produtos que se exigem sejam entregues, ou ainda estabelecendo controles que ajudem a assegurar a qualidade e coordenar as alterações de requisitos e marcos de referência que possibilitam administrar o progresso do software.

Os métodos proporcionam os detalhes de como fazer para construir o software, seja com relação ao planejamento e à estimativa de projeto, ou à análise de requisitos de software e de sistemas, ou ao projeto da estrutura de dados, entre outros. Por fim, as ferramentas dão o suporte automatizado aos métodos.

A Engenharia de *Software* tem como objetivos finais:

- melhorar a qualidade de produtos de *software*;
- aumentar a produtividade do pessoal técnico;
- aumentar a satisfação do cliente.

Sob o ponto de vista da engenharia de software, os Objetos de Aprendizagem apesar de apresentarem características e princípios que os diferenciam de outros sistemas computacionais, como, software básico, utilitários, aplicativos, poderiam ser considerados como software comum, porém com funcionalidades e objetivos específicos que os diferenciam dos demais eles configuram uma categoria a parte, pertencentes à classe dos software educacional, onde se englobariam aplicações educacionais interativas, tais como atividades multimídia, animações, simulações, etc. Nesta linha de raciocínio, os Objetos de Aprendizagem também necessitam de um modelo de desenvolvimento que contemple a construção de mecanismos de interação adequados à proposta instrucional.

Assim, partindo da premissa que Objetos de Aprendizagem interativos podem ser classificados como software, então se faz necessário que, para o desenvolvimento destes, se utilize metodologias de análise, projeto e desenvolvimento de software.

As metodologias de análise, projeto e desenvolvimento de propósito geral não são direcionadas especificamente para a produção de software educacional e Objetos de Aprendizagem, embora estejam sendo utilizadas para este propósito ou adotadas como fonte de referência e pesquisa para o desenvolvimento de outras metodologias adaptáveis ao contexto dos OAs. Dentre os diversos métodos de propósito gerais, é possível citar: modelo em cascata, em espiral, por prototipação, orientados a objeto, ágeis, entre outros. Contudo, é preciso ampliar a discussão sobre o uso dessas tecnologias visando modificar a forma com que a Educação propõe o ensino e como os materiais educacionais são projetados, desenvolvidos e entregues aos usuários finais.

É válido questionar se, pelas características peculiares dos OA como produtos educacionais, os modelos de processo atualmente aplicados ao desenvolvimento desses produtos são, de fato, efetivos no que tange à qualidade dos produtos gerados e principalmente, a viabilidade de seu desenvolvimento.

4.1. Desenvolvimento Ágil

Os métodos denominados “ágeis” são uma proposta de contraponto às metodologias de desenvolvimento tradicionais usadas há décadas, pois visa criar um processo de desenvolvimento de software “leve”, não tão voltado a grande quantidade de documentação do projeto do software, adaptável as mudanças de requisitos ocorridas durante o desenvolvimento e de grande interação entre as equipes que irão produzir o software, bem como com o cliente.

A partir de 2001, com a publicação do Manifesto Ágil (Beck, 2001), começaram a surgir alguns métodos que sintetizam os pontos desta visão de prática de desenvolvimento, os chamados “princípios ágeis”. Este manifesto foi proposto em 2001 por um grupo que se autodenominou a aliança ágil (*agille alliance*) - de 17 pesquisadores e especialistas em métodos leves de desenvolvimento de software, e defende um conjunto de valores, práticas ágeis e princípios.

Uma das principais diferenças dos métodos ágeis em relação aos métodos tradicionais de análise e desenvolvimento de software é a previsibilidade versus a adaptabilidade característica

relevante para o desenvolvimento de OA dirigidos ao treinamento de operados de sistemas complexos. Pesquisadores questionam até que ponto a previsibilidade é possível ser alcançada no desenvolvimento de software, porém a essência desta diferença está no contraponto do planejamento detalhado e prévio de todo o projeto (alta previsibilidade) versus o desenvolvimento de software incremental com ciclos curtos de tempo, escopo variável onde “alterações nos requisitos são bem-vindas” e alta interação da equipe, portanto, alta adaptabilidade.

Dentre os métodos ágeis, um que tem sido bastante utilizado e pesquisado é o Extreme Programming (XP), proposto por Beck (2000) e também um dos autores do Manifesto Ágil. Com relação aos “conceitos XP”, os quais devem nortear também a prática do desenvolvimento, destacam-se: as histórias de usuário – que são funcionalidades do sistema desejadas pelo cliente, que, em outras abordagens, é similar aos requisitos do sistema ou casos de usos –, o planejamento de entrega e a versão compacta (versão funcional do sistema).

4.1.1. FDD - Feature Driven Development

Feature Driven Development (FDD) é uma metodologia ágil para desenvolvimento de software, voltado para o cliente e orientado a modelagem.

Segundo Hartmann (2006) o Feature Driven Development é dividido em 5 atividades, cada uma destas possuindo sub atividades que compõem o processo como um todo. Essas 5 atividades são (em ordem): Desenvolver um modelo geral, gerar uma lista de funcionalidades, planejar por funcionalidade, modelar por funcionalidade e construir por funcionalidade. As três primeiras são realizadas uma vez antes da realização do projeto e as 2 restantes são realizadas a cada iteração. Essas atividades e suas sub atividades segundo Hartmann (2006) são:

Desenvolver um modelo geral: O projeto começa com uma análise superficial do escopo do sistema e seu contexto. Assim, são estudados os domínios de negócio do sistema e criado um modelo geral baseado nestes. Depois é criada uma modelagem superficial para cada área de domínio existente. Cada um desses modelos é revisado por um grupo aleatório de membros do projeto e melhorias são discutidas. É escolhido o modelo de domínio melhor projetado, ou senão é escolhido um modelo que é a junção de mais de um domínio projetado. Finalmente, os modelos

escolhidos para cada área de domínio são fundidos para gerar um modelo do domínio como um todo (ou domínio principal) do sistema.

Gerar uma lista de funcionalidades: O conhecimento obtido na fase de desenvolvimento do modelo geral é essencial para esta fase. Nesta, é elaborada uma lista de funcionalidades do sistema decompondo as áreas de domínio obtidas. Cada funcionalidade é uma pequena tarefa a ser implementada que gere valor para o cliente. Devem seguir o formato: ação, resultado, objeto, por exemplo: “Gerar relatório de vendas” ou “Validar a senha do usuário”. Essas funcionalidades são muito importantes para o processo como um todo. A não identificação delas causa um impacto enorme sobre o projeto, pois significa que a primeira fase não foi feita com sucesso e conseqüentemente os efeitos aparecem em cascata nas fases seguintes.

Planejar por funcionalidade: É realizado o planejamento de desenvolvimento de cada funcionalidade da lista obtida da fase anterior. São designados classes ou código específico para os programadores-chefe tomarem conta. Daí em diante, os programadores-chefe são responsáveis pelo código produzido nessas classes.

Modelar por funcionalidade: Os programadores-chefe escolhem algumas funcionalidades para que, junto com os clientes de código, sejam feitos os diagramas de seqüência e a modelagem completa das funcionalidades. A diferença desta modelagem para a modelagem feita na primeira fase é que esta é específica para cada funcionalidade em questão. Ou seja, nesta etapa pensa-se nas classes, métodos e atributos que irão existir. Ao final da modelagem, é realizada uma inspeção do modelo pela equipe que a fez ou por outra equipe designada.

Construir por funcionalidade: Após a modelagem, o código é finalmente implementado e os testes finalmente escritos. Assim, o código fonte é produzido e as funcionalidades ganham vida. O programador-chefe designa um programador para implementar uma certa funcionalidade. Logo, este último será o proprietário do código escrito. Após uma inspeção no código escrito, a funcionalidade é terminada.

4.1.2.TDD - Test Driven Development

O *Test-Driven Development* (TDD) é uma metodologia ágil de desenvolvimento de software introduzida por Beck (2003) para produzir o que se chama de “código limpo que funciona”. Apesar de ser um processo simples, alguns desenvolvedores têm dificuldades de aplicar essa técnica no dia-a-dia e não conseguem guiar um projeto orientado a objetos a partir de testes.

Essa metodologia de desenvolvimento é baseada em testes sistemáticos ao longo do desenvolvimento incremental do software e auxiliam os desenvolvedores a tomarem as decisões corretas e embasadas na análise dos resultados aferidos. Desta forma, o TDD não deve ser compreendido como uma ferramenta para testes, mas sim como uma metodologia de desenvolvimento, voltada para a criação de software de uma maneira simples e incremental.

O TDD, juntamente com a refatoração e a programação em pares, são práticas que fundamentam a metodologia ágil de desenvolvimento de software chamada de *Extreme Programming* (XP). No método XP, os testes contínuos ao longo do ciclo de desenvolvimento do software são característicos desta metodologia e de fundamental importância para a implementação correta das funcionalidades do software. No TDD, o desenvolvimento é centrado nessa característica, onde casos de teste são criados e aplicados de acordo com a funcionalidade que será implementada. Esses casos de teste são agrupados em uma lista de casos de teste ou simplesmente lista de testes.

Considera-se que um teste foi bem sucedido ao verificar que a funcionalidade avaliada foi corretamente implementada. Esse processo ocorre com código sendo produzido em rápidas e pequenas interações.

Diferentemente de outras metodologias de desenvolvimento, em que os testes são criados após a implementação de todas ou parte da funcionalidade, no TDD os testes são criados antes do desenvolvimento, permitindo a projeção e a correção de eventuais erros de implementação.

O desenvolvimento guiado por testes é um dos princípios e das principais práticas existentes em *Extreme Programming* (XP). A proposta do TDD é de que sistemas complexos podem ser desenvolvidos em pequenos passos incrementais utilizando-se de testes para guiar o projeto e a implementação do software. A principal dificuldade na utilização desta metodologia está na adaptação dos desenvolvedores sobre a forma de pensar o projeto. Ele não deve mais

projetar como o código será gerado para ter uma estrutura já definida, mas sim criar testes que definam o funcionamento de pequenas partes do sistema.

No TDD os testes são escritos antes do código de produção, compilação e verificação de falhas, provendo o que é preciso para escrever um código de produção que satisfaça esses testes. Segundo Beck (2003), o ciclo básico do TDD é formado pelas seguintes etapas:

1. Projete o que seu código deve fazer;
2. Escreva um teste que represente como uma pequena parte do sistema deve funcionar;
3. Execute o teste, provocando a falha e de forma rápida possível sem se preocupar com detalhes implementação;
4. A partir da verificação nos testes do funcionamento do código, realize uma refatoração organizando a estrutura do código gerado sem alterar o seu comportamento externo.
5. O TDD é um processo contínuo e iterativo, onde estas etapas anteriormente descritas devem ser repetidas até a homologação do código gerado.

A metodologia TDD, deve ser aplicada com a decomposição dos requisitos do sistema em um conjunto de comportamentos a serem cumpridos através das funcionalidades implementadas, onde para cada comportamento do sistema é necessário elaborar um teste de unidade para avaliá-lo.

4.1.3.BDD - Behavior Driven Development

Dave Astels (2003) observou que a grande vantagem da prática do TDD não está nos testes gerados, mas sim no fato de se pensar no design antes de escrever a primeira linha de código, no momento em que se descreve o comportamento do sistema. Assim nasceu outra metodologia ágil, sendo o desenvolvimento orientado a comportamento (BDD, do inglês *Behaviour-Driven Development*), que com o tempo evoluiu para um processo que engloba desde a análise de requisitos, até o desenvolvimento do código.

O desenvolvimento orientado a comportamento é semelhante ao TDD em vários aspectos, mas embora as diferenças pareçam sutis influenciam grandemente no modo de criar um sistema. O BDD transfere o foco dos testes de implementação para os comportamentos que o sistema

expõe. É possível afirmar que o software é desenvolvido “de fora para dentro”, onde os testes são construídos baseados nos requisitos do cliente ou em um roteiro de aceitação (também conhecidos por estórias). O diferencial está que BDD aborda o comportamento e o resultado como um todo, não se preocupando necessariamente com as classes, métodos e atributos de forma isolada.

Uma vantagem da abordagem BDD é que os comportamentos se modificam menos vezes do que os testes e, tipicamente, tais modificações refletem a necessidade de novas funcionalidades do sistema.

Os comportamentos do sistema também podem ser descritos em vários níveis de granularidade. Por exemplo, é possível falar sobre os comportamentos que o sistema deve apresentar no conjunto ou caracterizar comportamentos que caracterizam componentes individuais do sistema.

De modo geral, o BDD apresenta um framework baseado em três princípios (Astels, 2003):

1. A área de negócios e a de tecnologia precisam se referir a mesma parte do sistema da mesma forma;
2. Toda parte do sistema precisa ter um valor identificável e verificável para o negócio;
3. Analisar, projetar e planejar tudo de cima a baixo tem retorno decrescente.

Portanto, é possível definir o BDD como a união de várias práticas consideradas ágeis e úteis no desenvolvimento de software, cuja ênfase está nas funcionalidades de alto valor e na redução dos custos de mudança por meio da identificação do que de fato está sendo testado.

4.2. Desenvolvimento de Interface

As metodologias de desenvolvimento de software que estão surgindo, vêm mostrando que a engenharia de software tem buscado uma aproximação cada vez maior com o usuário através da Ergonomia com o propósito de fornecer através da tecnologia, formas de ação mais intuitivas na realização de tarefas e atividades (Preece, 2005).

Segundo Wisner apud Fialho & Santos (1995), Ergonomia é o conjunto dos conhecimentos científicos relativos ao homem e necessários à ferramentas, máquinas e dispositivos que possam ser utilizados com o máximo de conforto, segurança e eficácia. Essa definição evidencia dois

aspectos fundamentais na prática da ergonomia: o conjunto de conhecimentos científicos sobre o homem e a aplicação desses conhecimentos na concepção de ferramentas e máquinas, dispositivos que o homem utiliza na atividade de trabalho.

A Ergonomia para a Interação Humano-Computador (IHC) propõe-se a estudar as dificuldades encontradas nesta relação, procurando aperfeiçoar a capacidade cognitiva do usuário na realização de suas tarefas e atividades. Assim, a Ergonomia enfoca as seguintes características (Preece, 2005):

Conhecimento do usuário:

- Cognição Humana;
- Personalidade;
- Comunicação humana e semiótica.

Conhecimento do Trabalho:

- Objetivos, restrições e estratégias do usuário;
- Perspectiva do trabalho – tarefa (trabalho prescrito) e atividade (trabalho efetivo);
- Lógica envolvida – funcionamento (interna/projetista) e utilização (externa/operador)

Assim, a parte humana também vem sendo estudada como componente dos sistemas e, portanto, para o desenvolvimento bem sucedido das funcionalidades do software é necessário não só prover soluções que atendam as necessidades dos usuários como, também é importante viabilizar e facilitar a comunicação entre homem e máquina.

O conceito de interface é atribuído ao que interliga dois sistemas. Essa interligação aplicada à comunicação entre homem-máquina é estabelecida através de um artefato que reúne um conjunto de dispositivos sensíveis cujas funções são ativadas pelas ações do usuário, estimulando sua percepção. Esse processo de envio de ações e recepção de estímulos permite ao usuário controlar e avaliar o funcionamento da máquina. No processo de interação usuário-sistema a interface é o meio que através da combinação de software e hardware viabiliza e facilita os processos de comunicação entre o homem e máquina. Diferentemente das convencionais interfaces de máquinas, a interface de sistemas computacionais diferencia-se por exigir maior esforço cognitivo dos usuários na interpretação e expressão de informações processadas pelo sistema (Norman, 1988). A interface pode ser definida como um artefato que possui um

componente físico, onde o usuário interage através da percepção e manipulação, e outro conceitual, que o usuário interage cognitivamente interpretando, processando e raciocinando. Portanto, a interface se configura como um sistema de comunicação, que a sendo tanto um *meio* para a interação usuário-sistema, quanto uma *ferramenta* que oferece os instrumentos para o processo comunicativo.

A interface é a parte funcional do sistema, responsável por implementar processos computacionais necessários (a) para controle de hardware, (b) para a construção de objetos virtuais para interação, (c) para a geração de símbolos e mensagens para a comunicação do usuário com o sistema do sistema (Preece, 2005).

O Design de Interação como disciplina, tem o propósito de examinar comportamentos e interações, focando os complexos diálogos que ocorrem no espaço físico ou virtual, entre os usuários e os artefatos tecnológicos de uma interface. A característica multidisciplinar do Design como disciplina, amplia sua área de atuação com o Design de Interação. Este segmento do Design surge da necessidade de estudar e viabilizar o processo comunicacional dentro do espaço virtual, avançando assim em áreas como a Informática e Ciências da Computação.

O Design de Interação ainda que possa tocar em alguns pontos a Engenharia de *Software*, vai um pouco mais além. Enquanto a Engenharia de Software, disciplina de áreas relacionadas à computação se ocupa, fundamentalmente, de produzir soluções específicas para atender a automação de tarefas, o Design de Interação possui um espectro maior de metas a atingir, ligadas à usabilidade e às expectativas dos usuários para a realização dessas tarefas.

Essa nova área surge da busca pelo equilíbrio desses dois grupos de metas, o conhecimento do contexto de uso, da tarefa a ser realizada e do público-alvo e suas necessidades.

O objetivo do Design de Interação é permitir ao usuário quantificar e analisar a informação que os sentidos são capazes de transmitir. Enquanto metodologia, o Design de Interação é hoje considerado como um processo que efetua a modificação de interações entre os elementos de um sistema, de tal modo que a sua estrutura seja substancialmente alterada, originando novos comportamentos.

Uma característica importante de uma interface são as *affordances* do sistema, propriedades percebidas e reais de um artefato, que indicam ou determinam como um artefato pode ser utilizado (Norman, 1988). Para o autor, as *affordances* fornecem elementos indicativos sobre o processo operacional e a função dos artefatos.

Diferentemente das interfaces de software convencionais, o objetivo da interface de um software educacional além de apenas ajudar usuários a realizar mais rapidamente ou de maneira mais efetiva suas atividades ou tarefas, pode também, engajá-los a refletirem sobre o processo (Wilensky e Resnick, 1999).

Quando a atividade educacional está implícita na tarefa, pode não ser desejável ou necessário minimizar a carga cognitiva da mesma. A redução da carga cognitiva, geralmente é exigida e indicada para a realização de tarefas que não estejam relacionadas à aprendizagem e ao conteúdo didático. Essa é a principal e fundamental diferença entre um projeto de interface para software convencional e um projeto interface para software educacional. Portanto, interfaces concebidas e pautadas a partir exclusivamente de critérios ergonômicos com o objetivo de tornar as tarefas mais fáceis para resolução de problemas podem não ser as melhores interfaces para aprendizagem.

4.2.1.Design Centrado no Usuário

A condição prévia para a concepção de interfaces para sistemas interativos é o desenvolvimento de um modelo de usuário. Conhecer o usuário é, portanto, um fator importante e crítico para as especificações de interfaces. Os usuários têm necessidades, objetivos e capacidades que têm que ser consideradas. O modelo de Design Centrado no Usuário (DCU) permite identificar e projetar na interface vários aspectos relevantes (Norman 1986), como por exemplo, as preferências do usuário, capacidades individuais, como eles reagem aos diferentes problemas de uso e assim, entender como as respostas do sistema podem ser adaptadas para ao usuário.

A forma de levantamento de requisitos utilizada na metodologia de design de interface centrado no usuário foi trazida da metodologia de desenvolvimento ágil XP, que introduziu a prática de expressar os requisitos sob a forma de histórias de usuários ajudando na identificação e elaboração das funcionalidades e elementos de interface relevantes para ele.

Princípios de Uso	Início de planejamento; Estimativa de esforço / risco; Descrever funcionalidades; Entender o usuário através de suas características principais.
Vantagens	Dados suficientes apenas para estimativas; Comunicação verbal para registrar os casos de uso; Detalhamento progressivo; Terminologia do usuário.
Desvantagens	Os entrevistados não são necessariamente usuários finais; Levantamento das características podem ser vagas ou imprecisas.
Relacionado ao	Casos de uso, funções de usuário, cenários de uso.

Quadro 2 - Vantagens, Desvantagens, Princípios de Uso e Relações da aplicação do DCU

O design centrado no usuário parte da elaboração de um perfil de usuário para a caracterização do público-alvo de um sistema fornecendo informações úteis para tomada de decisões de projeto. Conhecer o usuário é fundamental, a proposta de um perfil de usuário é estabelecer as necessidades e exigências de um grupo determinado de usuários e considerá-los durante as especificações de design. A caracterização desse perfil pode ser feita com a ajuda de algumas propriedades fundamentais:

- Características psicológicas (por exemplo, a atitude, a motivação).
- Conhecimento e experiência (por exemplo, experiência profissional).
- Trabalho e as características da tarefa (por exemplo, frequência de uso, a estrutura da tarefa).
- Características físicas (por exemplo, critérios de acessibilidade).

Para a determinação das características do usuário, o designer de interação pode usar técnicas de coleta de dados como entrevistas e questionários, para que os aspectos mais importantes possam ser analisados. A partir dos dados resumidos, o designer pode extrair informações importantes para o design da interface, onde um perfil de usuário servirá de matriz de base para vários outros usuários, permitindo que a equipe compreenda mais claramente as funções e características em comum a cada usuário.

4.2.2. Design Centrado na Tarefa

Para Preece *et al.* (2005), a análise de tarefas é utilizada principalmente para investigar uma situação existente, não prevista para novos sistemas ou dispositivos. Ela é usada para analisar a lógica existente em uma ação do usuário e a finalidade desta ação: o que eles estão tentando fazer, porque eles estão tentando fazer e como eles estão conseguindo fazer?

Uma das características principais do Design Centrado na Tarefa (DCT) para o desenvolvimento da interface do usuário é dar suporte ao usuário na execução de sua tarefa operacional. Para isto, há diversas maneiras de registrar essas ações, no entanto, todos esses registros devem permitir a especificação dessas tarefas, de forma hierárquica e seqüencial. O objetivo é definir o caráter seqüencial dos pontos da tarefa do usuário e os passos que devem ser percorridos para atingir um determinado objetivo. Geralmente, uma seqüência pode ser subdividida em partes menores. O aspecto hierárquico de uma tarefa também deve considerar as limitações da cognição humana e, portanto, destinar-se a estruturação de uma tarefa em partes específicas.

Princípios de Uso	Descrever um caminho através de um caso de uso; Descrever o comportamento interativo, descrição de novas idéias através de funcionalidades.
Vantagens	Fácil de produzir pode ser desenvolvido por todas as partes envolvidas; Aumenta a compreensão de um caso de uso através da narrativa do usuário.
Desvantagens	Não é muito preciso, é necessária a transformação dessa narrativa em um modelo estruturado para o desenvolvimento (de casos de uso para cenários)
Relacionado a	Personas, casos de uso, diagramas de seqüência

Quadro 3 - Vantagens, Desvantagens, Princípios de Uso e Relações da aplicação do DCT

Através da descrição de casos de uso é possível elaborar um exemplo real de como as pessoas interagem com um sistema. Eles descrevem as etapas, eventos e / ou ações que ocorrem durante a interação e como o usuário trabalha com o software, podendo se elaborada uma descrição seqüencial que servirá de modelo de construção da estrutura da interface.

Casos de uso permitem discutir caminhos alternativos através de uma tarefa. Para descrever todos os caminhos possíveis através de uma atividade, a equipe de desenvolvimento tem portanto, que escrever vários cenários de uso. Posteriormente através de um diagrama de seqüência é possível determinar a lógica de implementação para cada tarefa a ser elaborada na interface.

Para tornar eficiente esse processo de design de interface centrado na tarefa é necessária a transformação dos casos de uso baseados nas descrições do usuário em cenários. A adaptação deve seguir algumas regras simples: (1) escrever pelo menos um cenário para cada parte envolvida, (2) analisar pelo menos uma de cada cenário, (3) escrever vários cenários de cada tarefa para usuários que desempenham múltiplas funções.

A principal diferença entre os casos de uso e os cenários e que diz respeito à modelagem das ações do usuário na interface é que os casos de uso normalmente referem-se a ações genéricas, já os cenários remetem a exemplos de ações.

Escrever cenários de atividades é, portanto, a primeira fase do design centrado na tarefa onde o design é focado em identificar e representar as práticas cotidianas do usuário na realização de tarefas identificando os problemas e oportunidades no processo operacional do usuário para serem transformados em novas formas interação. Além disso, os cenários de atividades são similares a outros tipos de cenários, e, portanto, podem ser descritos de forma muito breve.

Durante a atividade de design, cenários servem como um meio flexível e barato para a criação e discussão de idéias e propostas. O foco em situações realistas ajuda os designers a gerarem soluções específicas para atenderem as necessidades do usuário na realização de tarefas. Os cenários são avaliados e refinados, identificando as principais características e suas conseqüências para o uso através da análise de problemas.

Ao escrever os cenários das tarefas, os designers podem identificar objetos que desempenham um importante papel dentro de um determinado cenário possibilitando construir um ponto de vista para cada objeto a partir da perspectiva imaginária que o software permite materializar no ambiente virtual.

4.2.3.Design Centrado na Atividade

Na especificação de interface do usuário, o modelo de domínio da atividade ou o modelo operacional (Constantine *et al.* 1999; 2002) descreve o contexto e o ambiente em que o processo de interação com o usuário ocorre. O Design Centrado na Atividade (DCA) baseia-se principalmente na identificação, das condições organizacionais que influenciam o processo de especificação de interface.

Além de considerar no modelo as necessidades da atividade organizacional, essa característica é também parte da estratégia para a fase de definição de objetivos que devem conduzir o design. Nessa fase, a definição de metas claras e os objetivos ajudam a construir uma base comum para todas as partes envolvidas identificarem os problemas para o desenvolvimento do software e assim apontarem soluções baseadas no domínio do conhecimento específico das atividades de cada área.

O Design Centrado na Atividade descreve os objetivos das atividades das organizações e quais resultados devem ser atingidos ao utilizar o software. Assim é necessária a utilização de especificação interativa para fornecer informações sobre a razão e os motivos para a concepção de uma funcionalidade específica no projeto de interface. O modelo de atividade representa os elementos que existem ou eventos que acontecem e interagem no ambiente de negócios.

Um modelo de atividade identifica os tipos fundamentais de atividades da organização e as relações entre elas dentro de seu contexto do negócio, definindo assim o alcance do sistema.

Princípios de Uso	Capturar os mais importantes tipos de objetos no contexto das atividades; Compreender as práticas da atividade.
Vantagens	Podem orientar a programação no desenvolvimento a partir de testes de resolução de problemas; Um cenário por partes envolvidas ou vários cenários se as partes envolvidas estão envolvidas em muitas atividades; Detalhamento da interface de forma progressiva.
Desvantagens	Especificações técnicas são difíceis de ler para pessoas que não são de TI; Os entrevistados não têm necessariamente que serem os utilizadores finais; Demorado para criar.

Relacionado a	Tarefa de modelagem (por exemplo, casos de uso essencial); Fluxos de dados / informações; Diagramas de atividade; Análise das partes envolvidas; Análise de tarefas.
---------------	--

Quadro 4 - Vantagens, Desvantagens, Princípios de Uso e Relações da aplicação do DCA

Para o design centrado na atividade, é necessário definir as atividades comuns a serem reproduzidas na interface que incluem o desenvolvimento de (1) um modelo de contexto (geralmente um diagrama de fluxo de dados ou informações), mostrando como o sistema se encaixa em seu ambiente global (2), um modelo de alto nível dos requisitos do negócio (muitas vezes um elemento essencial de casos de uso modelo), (3), um glossário de termos técnicos do negócio (4), um modelo de domínio (geralmente um diagrama de classes ou de dados), representando as classes organizacionais importantes ou entidades e (5) um modelo de processo de negócio (muitas vezes um diagrama de fluxo de dados ou diagrama de atividade), representando uma visão macro do processo de negócio a ser representado na interface do sistema.

O design centrado na atividade tem como base o desenvolvimento de cenários problemas (Carroll *et al.* 2002), onde o levantamento dos requisitos a serem trabalhados no projeto são sempre ligados a análise de problemas no desempenho de uma atividade. As atividades descrevem as características de uma situação que tem uma influência sobre as partes envolvidas. Assim, cenários são constituídos de problemas que descrevem como funcionalidades específicas podem ter impacto sobre as atividades e experiências do usuário.

Problemas podem ser vistas como casos importantes para o domínio do negócio:

- Problemas descrevem o impacto de características específicas do cliente.
- Registros dos problemas contribuem para a análise de cenários porque são escritos para isolar as características mais importantes das narrativas dos usuários.
- Extensão dos problemas nos cenários, apontando para os possíveis efeitos que uma característica pode ter em outros cenários.
- Análise dos problemas promove uma visão equilibrada da situação (os impactos positivos e negativos).

- Cenários problemas ajudam no design a aumentar os impactos positivo e diminuir os negativos da interface.
- Um cenário pode ser definido como uma série de ações e respostas do sistema ao tentar alcançar a meta. Durante essa tentativa, os obstáculos podem impedir que o objetivo seja alcançado. Estes obstáculos devem ser documentados no cenário para orientar os projetistas em suas tentativas de identificar soluções gráficas ou de interação que ajudem a resolver questões relevantes e, assim, melhorar a qualidade do software.

4.3. Objetos de Aprendizagem Baseados em Simulações

O software educacional através da interatividade aliada a recursos multimídia permite ao aluno vivenciar a exploração de conceitos impossíveis sem a utilização de representações virtuais em uma sala de aula ou laboratório, testarem diferentes caminhos e acompanhar a evolução temporal das relações entre causa e efeito. A possibilidade de relacionar conceitos representados virtualmente através dos objetos de aprendizagem facilita ao aluno, visualizar diferentes pontos de vista e comprovar hipóteses, o que contribui para a prática pedagógica, despertando novas idéias e curiosidade e ao estimulá-los a resolverem problemas.

Através da interatividade o aluno é encorajado a descrever ou implementar alguns aspectos do fenômeno a ele ensinado com a representação de algumas situações já previamente definidas e outras que devem ser complementadas por ele. Esse processo de interação requer que o aluno se envolva com o fenômeno ou conceito, procure descrevê-lo em termos de comandos, ações ou facilidades fornecidas pelo software educacional, observando assim através de estímulos as variáveis que atuam no fenômeno e como elas influenciam o comportamento do mesmo. Nesse envolvimento com o fenômeno, o aluno elabora uma série de hipóteses e idéias que deverão ser validadas por intermédio do processo de simulação do fenômeno no computador. Portanto, o papel do computador nesse caso é o de permitir a elaboração do nível de compreensão dentre outras formas, por meio do ciclo de descrição, execução, reflexão, depuração e descrição.

O software educacional desenvolvido a partir da integração de conceitos de interação homem-máquina e estruturado através de metodologias ágeis da engenharia de software pode criar condições para o aluno fazer a transição entre a simulação e o fenômeno no mundo real.

Os objetos de aprendizagem podem ser ferramentas úteis para intermediar a linguagem a qual o conhecimento está estruturado, amenizando assim as dificuldades em compreender determinados conceitos, tratados em contextos familiares aos alunos a fim de facilitar a conexão com os fatos e a aplicabilidade do conhecimento, por meio de situações problema.

Segundo Wiley (2000), objetos de aprendizagem recursos digitais que podem ser reutilizáveis no apoio a aprendizagem e que possuem o caráter de construção e aplicação de conceitos, que:

- Permitam mudanças de parâmetros físicos
- Permitam exploração e interação
- Possam ser utilizados em diferentes projetos pedagógicos

Para Wiley (2000), os objetos de aprendizagem podem ser considerados atividades interativas que possibilitam vivenciar a exploração de conceitos através de recursos de multimídia. É uma ferramenta educacional que permite ao professor se aproximar do mundo de interesse dos alunos mais facilmente. Através dele, o aprendizado se dá de forma mais participativa com maior interação do aluno que passa a ter um papel mais ativo no processo com a transmissão do conhecimento deixando assim de ser unilateral.

O conceito de objeto de aprendizagem (“*learning object*”) surgiu no final da década de 90 como uma proposta de paradigma na elaboração de materiais instrucionais. A idéia era a de construir repositórios de materiais didáticos, devidamente catalogados e disponibilizados, para que os usuários (instrutores e aprendizes) fizessem economia (de tempo e dinheiro) ao preparar seus cursos e processos instrucionais. O conceito envolvia a idéia que quando os professores acessam materiais instrucionais, eles freqüentemente os quebram em suas partes constituintes (Wiley 2000) e os reorganizam de forma a apoiar os seus próprios objetivos educacionais. Portanto, se houvesse um conjunto de objetos instrucionais reutilizáveis e granulares, como componentes individuais, o passo inicial da decomposição já estaria dado, aumentando a eficiência do processo (Wiley 2000).

Todos esses recursos apresentam os atributos fundamentais de um objeto de aprendizagem: que ele seja reutilizável, que seja digital, que seja um recurso e que esteja envolvido com a aprendizagem.

O conceito de reutilização possibilita que aplicativos sejam compartilhados com múltiplos usos em diferentes situações de aprendizagem. Para que essa utilização se torne viável, é necessário pensar, durante a produção, na chamada granularidade do aplicativo - se ele é uma peça minúscula dentro de uma aula, de um currículo, ou se é uma aula em si, uma disciplina completa.

As noções associadas de granularidade e recombinação – a abrangência dos conceitos tratados em um único aplicativo e a possibilidade de combinação dos aplicativos por parte do professor que o utiliza, são consideradas de forma a adaptá-los à sua proposta de trabalho, à sua visão pedagógica, ao seu plano de curso e de aula.

Por princípio, a granularidade proposta é a menor possível: aborda-se um único conceito de física de cada vez. Por definição, privilegia-se uma discussão conceitual deste tópico, buscando aspectos que a literatura e a prática docente indicam apresentar problemas de aprendizagem.

Os objetos de aprendizagem derivam da idéia da programação orientada ao objeto onde é possível modelar a aprendizagem através do agrupamento de partes fracionadas do conhecimento.

Esse tipo de abordagem estrutural do conhecimento permite organizar hierarquicamente o conhecimento, considerando prioritariamente as idéias mais inclusivas e posteriormente os conceitos subordinados ou específicos, ocorrendo à aprendizagem significativa, e, portanto, conceitos simbolicamente expressos são relacionados de maneira substantiva e são especificamente relevantes a algum aspecto de sua estrutura cognitiva para a aprendizagem. Assim utilizando objetos de aprendizagem desenvolvidos sob simulações torna-se mais fácil construir o conhecimento quando se inicia de uma idéia mais geral e inclusiva e se encaminha para idéias menos inclusivas através de mecanismos de interação homem-máquina.

Através de simulações gráficas e interativas é possível representar as forças que atuam em um objeto. Quando esse objeto se movimenta, ele revela a relação entre as forças que proporcionaram o seu deslocamento, que se modifica de acordo com a alteração das forças que elas representam. Assim, através da análise do modelo configurado pelo aluno, é possível uma representação visual concreta das suas nuances abstrata, e através da simulação gráfica e interativa é possível tornar essa experiência empírica, concreta.

Uma simulação gráfica e interativa representa a evolução temporal de um modelo da realidade tornando possível a exibição de elementos abstratos em sua representação figurativa e concreta. Na medida em que possibilita a percepção visual de variações temporais de grandezas

físicas (abstratas ou não), as simulações gráficas e interativas conduzem a um nível de abstração da realidade que permite o nivelamento dos requisitos cognitivos para a compreensão e aquisição do conhecimento por alunos de variados níveis de cognição.

As simulações gráficas e interativas podem facilitar a compreensão a partir da possibilidade oferecida ao aluno de visualizar a representação gráfica de um modelo lógico de percepção da realidade, transformando conceitos abstratos em uma representação gráfica da realidade, e assim, através da possível interação transformar o conteúdo lógico em conteúdo psicológico.

A construção do conhecimento se dá a medida em que o aluno interage com a informação, permitindo que ele faça conexões importantes entre significados o que desse modo possibilita a aprendizagem significativa.

Para Longmire (2001), algumas características tecnológicas, favorecem pedagogicamente o uso de objetos de aprendizagem na educação, tais como:

Reusabilidade /Flexibilidade - os OAs são construídos de forma modular, seguindo o modelo de programação orientada a objeto que tem como premissa a flexibilidade de reaproveitamento do código, de forma que, podem ser reutilizáveis com baixo custo de manutenção.

Granularidade/Facilidade para atualização – sendo os mesmos objetos utilizados em diversos momentos e montados de forma modular, a atualização dos mesmos em tempo real é relativamente simples, caso o acesso e carregamento dos dados seja estruturado a partir de um banco de informações;

Adaptabilidade/Customização – a modularização torna os objetos independentes, de fácil manutenção e reaproveitamento. A flexibilidade de desenvolvimento permite principalmente que o aplicativo seja customizado de acordo com as necessidades verificadas na fase de levantamento de requisitos e também modificados posteriormente para se readequarem a novas necessidades de treinamento.

Interoperabilidade - os OA podem ser utilizados em qualquer plataforma de ensino, sem a necessidade de hardware de alto desempenho;

Aumento do valor de um conhecimento - ao ser reutilizado e adaptado a um novo conhecimento, os OAs vão sendo melhorados ao longo do tempo, levando a uma melhora significativa na qualidade do ensino de um novo conteúdo.

Por serem digitais os objetos de aprendizagem podem ser tratados como produtos de software. Portanto, seu desenvolvimento requer o conhecimento das técnicas utilizadas em engenharia de software, como a ES.

4.4. Análise comparativa entre IHC e metodologias ágeis de ES

Para projetar sistemas é necessário que especialistas na área de IHC tenham habilidades e conhecimentos distintos, que vão desde uma compreensão da psicologia humana para entender a lógica de interação dos usuários, até os requisitos de modelagem e de design de interface a serem projetados.

Profissionais de IHC são conhecidos como designers de interface, especialistas em usabilidade, designers gráficos, especialistas em experiência do usuário, etc., um profissional que combina o conhecimento de usabilidade, projeto gráfico e interação. Considerando que IHC incide sobre o Design da Interface do Usuário (DIU) questões como a facilidade de uso, a lógica de interação, o desempenho e satisfação do usuário ou a estética são os principais pontos de direcionamento que o projeto deve considerar no desenvolvimento de um software.

Já a Engenharia de Software (ES) entende como os requisitos funcionais são traduzidos e executados no desenvolvimento de um sistema. Os engenheiros de software são geralmente formados em áreas como algoritmos, estruturas de dados, arquitetura e design do sistema ou banco de dados.

Essas duas áreas que se complementam, geralmente são tratadas separadamente e de forma assimétrica dentro do ciclo de desenvolvimento de um software. (Figura 1)

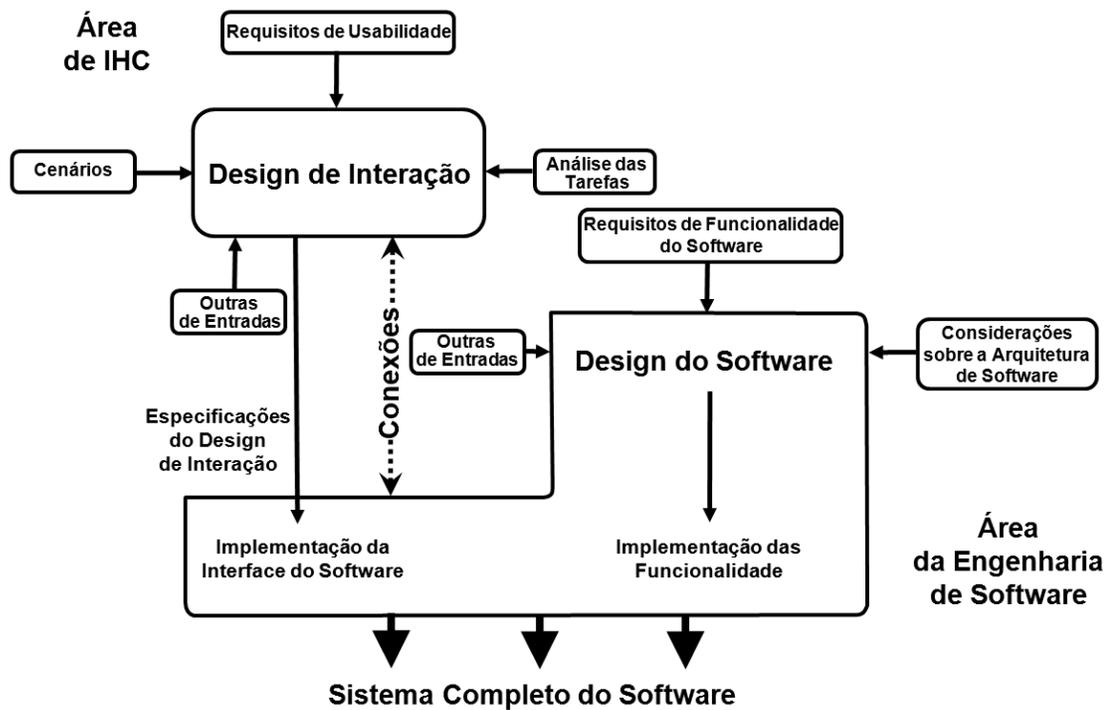


Figura 1: Assimetria entre IHC e ES no ciclo de desenvolvimento de software

IHC e ES são reconhecidas como áreas compostas por profissionais de habilidades distintas, onde cada conjunto de habilidades é essencial para a produção de software de qualidade, mas esse conjunto dessas habilidades, separadamente não são auto-suficientes. Como engenheiros de software também são responsáveis pela aplicação de especificações de todo design de interface e execução de códigos para viabilizar interações e funcionalidades de interface, existe a necessidade de se comunicarem com a equipe da área de IHC.

Para atender aos requisitos específicos de um software educacional é necessário que as duas áreas, IHC e ES, converjam e se complementem para desenvolver funcionalidades onde a lógica de interação é estabelecida como um processo pedagógico desenhado no projeto instrucional, por um professor, orientador ou desenhista instrucional.

Algumas das fases que compõem os processos de ambas as áreas, IHC e ES, possuem pontos em comum. Já outras possuem pontos de convergência que precisam ser integrados. (Figura 2)

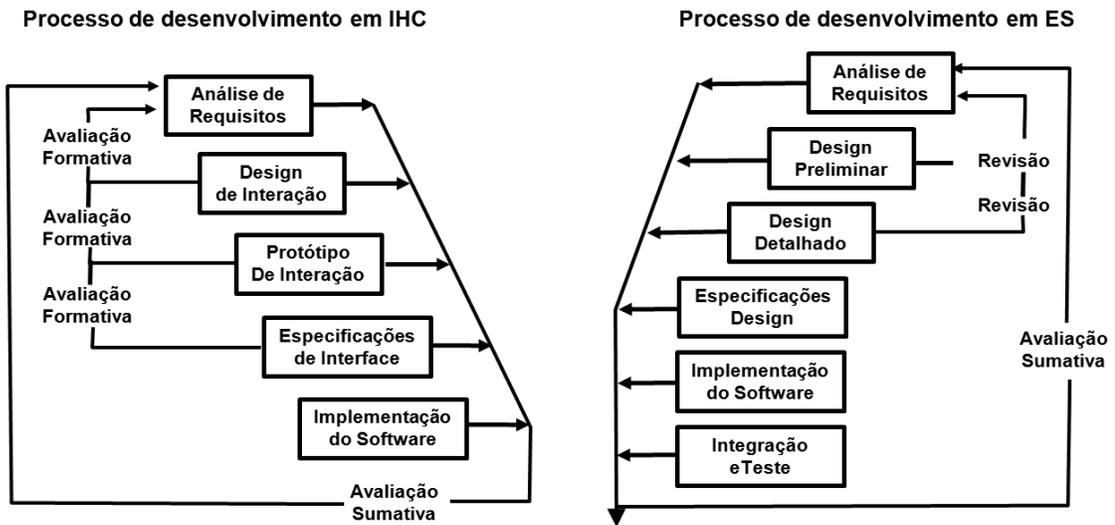


Figura 2: Processos de IHC e ES para o desenvolvimento de software

A interface entre o sistema e o usuário é a área onde os processos IHC e ES convergem e as equipes são obrigadas a trabalharem juntas, a fim de garantir que o software resultante se comporte como especificado inicialmente na Engenharia de Requisitos (ER). Para proporcionar um elevado nível de interação com usuário, a equipe de engenharia de software tem que trabalhar com pessoas com experiência em IHC, mas o ponto de interseção e colaboração tem que ser mais claro. Por isso, ainda há falta de integração clara de métodos e processo de IHC com métodos tanto clássicos quanto ágeis de ES (Figura 3).

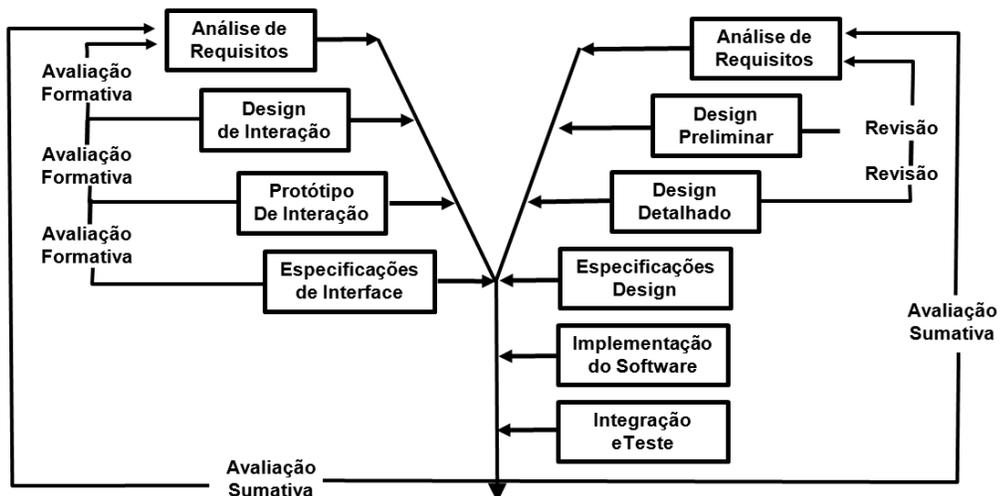


Figura 3: Pontos com e sem integração entre as áreas de IHC e ES

Várias linguagens (como a UML, por exemplo) têm sido desenvolvidas como base, a fim de superar esta limitação e necessidade de comunicação entre as áreas de ES e IHC, mas essas linguagens criaram um maior formalismo e diminuiu a compreensão para os clientes.

A linguagem UML mesmo sendo uma das mais poderosas linguagens de modelagem para ES, em questões de design de interface, a UML só permite descrever interações formais sem que seja possível prover a visualização do comportamento do usuário real. Conseqüentemente, o cliente e o analista de TI (Tecnologia da Informação) podem inicialmente pensar que estão de acordo em um projeto, mas só depois perceberem que tinham visões muito diferentes sobre a implementação de funcionalidades e comportamentos do sistema. Através dessa convergência de opiniões e visões do processo de aprendizagem e nos modelos didáticos é possível identificar as lacunas no ensino que podem ser preenchidas pelo apoio da tecnologia.

IHC também possui modelos formais de projeto, como casos de uso, modelos de tarefa ou de modelos. No que diz respeito ao uso de artefatos informais como *storyboards*, cenários Carrol *et al.* (2002) ou de protótipos, IHC é geralmente mais informal e, portanto, mais aberta à colaboração com as demais partes envolvidas no projeto. Como há uma necessidade de ambas as áreas (IHC e ES) se integrarem, é preciso estabelecer certo grau de formalidade para a descrição dos componentes da interface do usuário. No Quadro 5 são apresentadas de forma resumida as principais questões de integração levantadas por diversas publicações, onde para cada questão de integração apontada, são sugeridas formas de integração dos processos.

Questões de Integração	Formas de Integração
Mediar e melhorar as linhas de comunicação entre os usuários, especialistas em usabilidade e desenvolvedores.	Usar artefatos de meio peso, trabalhar com ferramentas adequadas para o projeto colaborativo, falar a mesma língua, trabalhar em pares.
Estender a engenharia de software para especificações e conceitos de IU.	Uso de artefatos conhecidos por ambas às profissões para alinharem as representações.
Reforçar a notação de modelos orientados a objeto da engenharia de software.	Usar anotações e modelos que podem ser compreendida por todas as partes envolvidas no projeto.
Estender métodos de ER para a coleta de informações sobre usuários e usabilidade	Utilização de princípios como métodos e práticas ágeis de IHC e ER.

Desenvolvimento de novos processos para projetos de sistemas interativos.	Reconsiderar os métodos aplicados, considerando prazos e demandas.
Representação do design utilizando diferentes formas como, por exemplo, protótipos.	Aplicar a prototipagem como método de design participativo, permitindo a todos os envolvidos no projeto reunirem os requisitos de teste do sistema sem codificação.
Transportar atividades do design centrado no usuário (UCD) e não apenas ferramentas e métodos.	Integrar as ferramentas comuns a ambas as profissões na implementação do sistema incorporando padrões de programação, princípios e conceitos de design, etc.

Quadro 5 - Questões e formas de integração entre IHC e ES.

Linguagens de modelagem muito formais ou complexas são inadequadas para a comunicação, especialmente em processos ágeis de design colaborativo com a participação das partes envolvidas. A utilização de *scaffoldings* para o desenvolvimento de software educacional exige uma forte integração entre as áreas envolvidas. Com a integração entre diferentes visões do projeto é que são percebidas as oportunidades e surgem as melhores soluções na modelagem desses *scaffoldings* dentro da interface do software.

Como linguagem de modelagem interdisciplinar para a elaboração dos requisitos, a pesquisa semi-estruturada realizada com profissionais do controle de tráfego aéreo e a revisão teórica sobre Sistemas Complexos realizada no capítulo 2 deste trabalho, apontam cenários conhecidos como as histórias do usuário (cenários leves) no desenvolvimento e protótipos como pontos principais de integração entre as técnicas de IHC e ES.

Cenários

Em ES, cenários são seqüência de eventos disparados pelo usuário e geralmente usados para os requisitos de coleta e análise de dados. Cenários, servem para identificar um segmento de uso para o sistema a ser construído, fornecendo uma descrição de como o sistema será utilizado (Pressman, 2001). Em contrapartida, aplicam-se cenários para IHC descrever o contexto de utilização do software, usuários, funções de usuário, tarefas e interação Carrol *et al.* (2002). Assim, os cenários podem ser usados como uma visão do sistema, explicando a funcionalidade e o comportamento interativo, baseados na descrição dos usuários e de suas tarefas. “Cenários podem apoiar a concepção do processo (...) como meios para testar hipóteses e estimular a

criação. (Eles) podem formar a base comum entre as disciplinas e, ainda podem ser integrados (...) em um ciclo de desenvolvimento" (Carrol *et al.* 2002).

Cenários são propostos por Carrol *et al.* (2002) como um meio de representar, analisar e planejar como um sistema computacional pode causar impacto nas atividades e experiências do usuário. Um cenário é uma descrição em geral narrativa baseada nas histórias dos usuários, mas também em outros formatos, que as pessoas fazem e experimentam conforme elas imaginam ou tentam fazer durante o processo operacional de sistemas complexos.

O cenário ajuda a identificar as motivações do usuário no uso do sistema, descrevendo as ações e tomadas de decisões para essas ações. Para o designer, ajuda a visualizar aspectos da atividade, da experiência ou necessidades do usuário. (Figura 4)

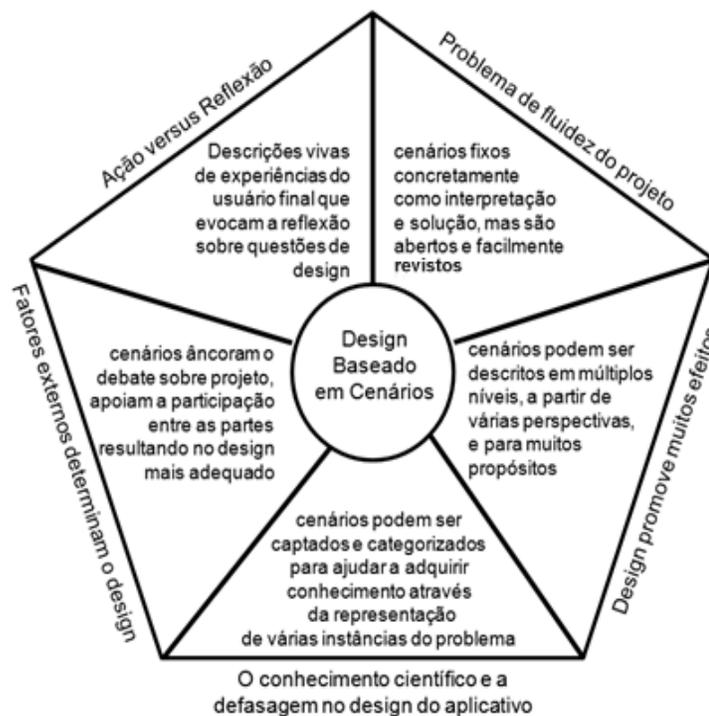


Figura 4 - Desafios e abordagens do uso de cenários

Durante a fase de análise de requisitos, designers e usuários negociam explicitamente os cenários e descrições do domínio de uso do sistema. Descrições de cenários hipotéticos facilitam a descoberta das necessidades do usuário que não são óbvias ou aparentes até para eles mesmos.

Cenários podem ser analisados para identificar os objetos centrais no domínio do problema e articular o estado, o comportamento e a função dos objetos que serão desenvolvidos para a representação de conceitos. A abordagem de casos de uso em análise orientada a objetos faz uso

desse tipo de visualização de situações concretas com o objetivo de identificar objetos computacionais essenciais em um sistema.

Na fase de avaliação, cenários podem ser usados para se coletar informação detalhada de como os usuários percebem o sistema. Designs de telas podem ser apresentados a usuários potenciais que tentam explicar o que pensam ser possível fazer e os efeitos esperados de suas ações.

Na fase de implementação, o uso de cenários apresenta benefícios, uma vez que ajudam a manter os designers focados em dar suporte às atividades dos usuários.

A documentação de um sistema pode facilitar, também, o próprio treinamento do usuário; as pessoas fazem melhor uso da documentação se ela é apresentada no contexto de tarefas típicas que eles têm que realizar especialmente as mais críticas.

Na abordagem estabelecida tradicionalmente, as descrições são abstratas, com foco em tipos genéricos, o processo é completo, exaustivo e orientado à tecnologia, os métodos são formais e rigorosos e os resultados bem especificados. Na abordagem baseada em cenários, as descrições são concretas, com foco em instâncias particulares, o processo é fragmentado e aberto, orientado ao trabalho.

Uma contribuição importantíssima dos cenários no processo de design é o estabelecimento de um canal de comunicação de mão dupla, usuário-designer.

A proposta de utilização de cenários pressupõe a visão de sistemas computacionais como transformadores das tarefas do usuário e de suas práticas sociais. Estórias são elementos coesivos importantes em qualquer sistema social; os cenários que a equipe compartilha motivam e direcionam a construção do software, representando a busca pelo equilíbrio entre intuição criativa e análise, entre a flexibilidade e a informalidade, necessários para a evolução sistemática do projeto.

Protótipos

Protótipos também são aplicados por ambas as áreas, embora o seu papel sempre tivesse uma menor importância, até esse momento em que as abordagens ágeis redirecionam a atenção sobre eles como forma de inspeção e testes no desenvolvimento de software. Protótipos em ES também podem ser usados para testar modelos e especificações funcionais, assim como na

identificação e compreensão de problemas, apontados através de inspeções realizadas junto ao usuário. Protótipos são reconhecidos como um tipo de versão de pequeno porte do software, que pode ser continuamente mudado e aperfeiçoado. Em IHC principalmente eles são classificados como artefatos para o design iterativo de interface de usuário e distingue diferentes dimensões dos protótipos.

A base que favorece a proposta de um modelo integrado é que alguns métodos informais de desenvolvimento ágil são semelhantes a práticas de IHC e, portanto, podem viabilizar uma linha comum de ação e consequente integração. Com protótipos e cenários como os veículos comuns, IHC e ES (ágil) podem se integrar em outros princípios e práticas compartilhadas utilizando métodos conhecidos por ambas as áreas.

Confrontado com as demandas dos atuais ambientes de projeto, uma forte dependência de métodos pesados, tais como modelos formais de ES, ou documentações pesadas, tais como guias de estilo de IHC são muito caros para o design de produtos interativos em qualquer caso. Isso fica mais evidente em software educativo como objetos de aprendizagem, onde cada pequena unidade de conteúdo funciona de forma independente e auto-suficiente do resto como uma parte fracionada de uma grande área do conhecimento. Essa característica dos objetos de aprendizagem favorece a utilização de metodologias ágeis e consequentemente a redução do volume de documentação exigida em detrimento à utilização de protótipos.

Uma das principais necessidades de um modelo integrado é a questão do custo que se torna mais evidente no desenvolvimento de software educativo que apresenta demandas de funcionalidades e características muito específicas que podem dificultar o desenvolvimento.

Facilidade e flexibilidade de desenvolvimento estão entre os principais argumentos para a elaboração e implementação de um modelo integrando IHC e métodos ágeis de ES aplicados ao desenvolvimento de software educacional.

Em contra ponto aos métodos pesados de desenvolvimento, surgiram métodos de ES como casos essenciais de uso, porém são abstratos demais para serem compreendidos por todos e também não são bem estruturados para orientar a Engenharia de Requisitos (ER) na especificação de um sistema. Compartilhar modelos ágeis característicos das áreas de IHC e ES, como protótipos podem facilitar a integração das áreas. Construídos com ferramentas adequadas, eles podem ser gerados e mudados rapidamente (prototipagem rápida), de forma contínua e serem suficientemente detalhados para agirem como referência para a comunicação com as demais

partes envolvidas no projeto. Eles podem preservar a experiência do projeto em uma linguagem visual entendida por todos.

Em comparação com os modelos clássicos e pesados de engenharia de software, os métodos ágeis começam a codificação, já na fase inicial do desenvolvimento, embora tenham uma curta fase de ER de interface e menos documentação.

Seguindo o paradigma do XP, a implementação do código ocorre em pequenos incrementos e iterações, sendo fornecidas ao cliente pequenas versões após cada ciclo de desenvolvimento. Durante uma breve análise de requisitos chamada fase de exploração, a equipe de desenvolvimento anota as histórias dos usuários em uma tentativa de descrever as necessidades dos usuários, transformá-las em cenários e assim definir as funcionalidades do software.

Os métodos ágeis são menos rígidos do que o XP e são mais cuidadosos na coleta inicial de requisitos que podem proporcionar diagramas de atividade, de caso de uso e conseqüentemente protótipos. No entanto, a fase de análise é concluída logo sem que os requisitos tenham sido declarados em um nível horizontal, porque o processo iterativo pressupõe que a falta de informação será preenchida em fases posteriores.

O Desenvolvimento de pequenos incrementos pode funcionar corretamente, desde que o software não tenha seu desenvolvimento centrado na interface com usuário (IU). Alterações na arquitetura do software geralmente não têm nenhum impacto sobre o que o usuário vê e interage. Com a interface do usuário, no entanto é diferente, principalmente em software educacional em que a interface é o elemento principal por funcionar como mediadora do processo de aprendizagem. Assim, o desenvolvimento ágil pode funcionar como base para a elaboração de um modelo integrado de desenvolvimento de software.

Ao projetar a interface com o usuário, as constantes alterações, devido à agilidade do processo de design de interação podem dar origem a conflitos frente às expectativas do usuário e a capacidade de aprendizado, causando inconsistência e levando ao usuário insatisfação (Constantine, 2004). Essa inconsistência, portanto é reflexo da falta de adequação das características do ambiente como elemento de contextualização do processo operacional de um sistema complexo levando a dificuldade de aprendizagem, não permitindo que o aplicativo cumpra o papel de facilitador do processo de aprendizagem a que se propõem. A avaliação de versões com a participação das partes envolvidas no projeto, não garante que todo o sistema forneça um modelo conceitual consistente de navegação ou conteúdo.

Ambas as áreas, ES e IHC, têm que lidar com um menor tempo de produção, sem comprometer a qualidade do software entregue. A redução contínua dos ciclos de desenvolvimento é, portanto, um grande desafio tanto para a gestão quanto para a metodologia de desenvolvimento e para a aplicação de ferramentas.

O desenvolvimento de software educacional tem demandas muito específicas no que se refere principalmente na integração da linguagem gráfica e do modelo de interação, onde a interface tem um papel mais importante ainda do que o desempenhado em software de outra natureza. A interface para aplicativos educacionais tem uma função mediadora onde a integração entre a linguagem gráfica definida no modelo de IHC e o processo de interação definido no modelo de ES configura o modelo instrucional que conduz o processo de aprendizagem.

Especialistas em IHC ou em ES, que se deparam com a necessidade de ter domínio de ambas as áreas, enfrentaram o problema através do desenvolvimento de abordagens leves ou chamadas de interação humano-computador ágil, por exemplo, a chamada *Extreme Usability* (Holzinger, 2006).

A proposta é um processo equilibrado híbrido, e ao mesmo tempo ágil de ES e IHC, que seja coerente com os princípios e práticas das duas áreas. A fim de identificar as similaridades entre ES ágil e IHC ágil, destacam-se diferentes abordagens para o design de interface (como DCU, DCT e DCA), com o objetivo de analisar o seu potencial ágil e suas diferentes contribuições para um processo de integração entre as áreas.

Como o XP, o Design Centrado no Usuário (DCU) originalmente é um processo altamente interativo. Difere dos métodos ágeis, no entanto, desde que os usuários reais sejam levados em conta e que as equipe de desenvolvimento tentem entender as necessidades do usuário e as tarefas por eles desempenhadas, antes de qualquer linha de código começar a ser escrita.

Constantine (2004) argumenta que o DCU produz soluções de design transformadas em protótipos visuais sem se basear nos dados reunidos ou requisitos, mas em um processo não registrado que promove transformações no projeto sem que seja compreensível nem rastreável. Tal processo geralmente é desenvolvido de forma intuitiva e produz soluções criativas, sem ser capaz de explicar ou ilustrar o que se passa no processo de criação. Além disso, DCU tenta convergir o resultado e alternativas de projeto em uma solução única, que é então continuamente avaliada e refinada. O DCU pode, portanto, ser demorado, ou mesmo falho, se muitos usuários estiverem envolvidos no processo reduzindo o tempo de produção e dificultando o design. A

interação com o usuário pode criar a ilusão de progresso, mesmo que o projeto realmente esteja em desenvolvimento às soluções são imperceptíveis.

A incorporação total de processos e métodos de DCU geralmente acaba sendo inadequada para a utilização da metodologia de desenvolvimento ágil. Constantine (2004), portanto, sugere a mudança do foco em vez dos usuários para as tarefas e apresenta métodos adequados para análise de requisitos. Sua abordagem centrada no uso retoma a base da filosofia da metodologia ágil e concentra-se no essencial e em modelos fáceis de compreender, como o modelo de tarefa ou modelo de atividade. Com a aplicação destas medidas os métodos de IHC tornam-se mais formais, mas a simplicidade de sua sintaxe ainda permite a colaboração com as partes envolvidas. O modelo de design centrado na tarefa pode ser aplicado por ambas as áreas, tanto IHC quanto ES. A sua aplicação em comum pode ser efetivada e validada no desenvolvimento de interface mais como um processo de engenharia do que como um processo de tentativa e erro (Constantine, 2002). Constantine também sugere protótipos como uma ferramenta para o levantamento de requisitos, mas argumenta que o detalhamento muito cedo, muitas vezes conduz a decisões prematuras. Ele recomenda protótipos abstratos, que podem ser classificados como prototipagem de baixa fidelidade. Embora possa ser bem sucedida, a aplicação do design centrado na tarefa, os produtos tendem a apoiar o desempenho do usuário em vez da experiência do usuário. O modelo tradicional de desenvolvimento de software que tem por objetivo o cumprimento bem sucedido de uma tarefa, não se aplica a software educativo, o enfoque no desempenho do usuário, muitas vezes não se traduz no aprendizado real e sim na aquisição do conhecimento de forma mecanizada através da simples memorização pelo processo de repetição da tarefa. Já o desenvolvimento sob o enfoque da experiência do usuário é um modelo mais adequado para software educacional por ser um processo pautado sob o entendimento do processo de interação do usuário com o ambiente.

Segundo o que é definido no design centrado na tarefa, o projeto pode ser considerado bem sucedido quando os usuários são capazes de atingir os objetivos da sua tarefa. Isso, no entanto, não pode ser a única aspiração de uma abordagem de design de interfaces para software educacional. Para uma integração do ciclo de desenvolvimento de software educacional, IHC deve considerar outros tópicos além de apenas o desempenho do usuário. É onde Donald Norman (2005) propôs a abordagem do design centrado na atividade (DCA) que como Constantine

(2002), ele argumenta que, concentrando-se nas tarefas, sem considerar a atividade, o design de interface pode não ser confiável e consumir muito tempo de aprendizagem.

O autor também afirma que o tempo de conclusão de tarefas provavelmente não é sempre o principal fator que leva o software a ser bem sucedido, principalmente tratando-se de software educacional.

Produtos que proporcionam uma experiência mais próxima das expectativas do usuário têm uma grande aceitação e utilização por parte do usuário mesmo quando há pouca usabilidade. Lowgren *et al.* (2004) apontam como conceber sistemas altamente utilizáveis e estéticos, alterna entre 3 níveis de abstração: a visão, a imagem operacional e especificação.

Se o designer é confrontado com uma situação de projeto, num primeiro momento freqüentemente surge uma visão fragmentada e difusa. Diversas propostas elaboradas competem para ser implementadas eventualmente resultando em um caótico conflito de visões. Por isso, em um projeto de desenvolvimento de software educacional, principalmente os direcionados ao desenvolvimento de habilidades operacionais é importante a identificação e a elaboração da chamada imagem operacional. Através dela, é definido o processo de interação entre conceitos e objetos. A versão inicial da imagem operacional é a exteriorização da primeira visão, capturada, por exemplo, na elaboração de protótipos interativos (alta fidelidade). Esses protótipos permitem a manipulação, simulação, visualização e tomada de decisão para tornar o projeto bem sucedido. A imagem operacional se suficientemente detalhada é eventualmente, transformada em uma especificação (visual) do projeto final. Em contraste com o Design Centrado na Tarefa, Lowgren *et al.* (2004) descrevem o projeto como sendo linear, não uma divisão há clara entre concepção e construção, mas sim uma contínua interação entre a visão, a imagem operacional e a especificação. Assim o desenvolvedor pode aprender e identificar bem suas possibilidades dentro do projeto, mantendo alto o número de possíveis soluções e aproximando o projeto de uma melhor solução o mais tarde possível, oferecendo assim uma proposta sempre atualizada às demandas mais recentes do usuário.

O quadro 6, mostra uma comparação das abordagens. Durante o desenvolvimento, um processo comum de design deve ser capaz de resolver as metas de usabilidade, bem como as experiências e metas de desempenho do usuário.

Características de Metodologias de IHC			
Design Centrado no Usuário (DCU)	Design Centrado nas Tarefas (DCT)	Design Centrado na Atividade (DCA)	Design sobre a Lógica de Interação
O foco é sobre os usuários: a experiência e satisfação do usuário	Foco está no uso: apoiar os objetivos tarefa	Foco é sobre as atividades	Foco definido pelo designer e é dependente de situação de projeto
Impulsionado pela entrada do usuário	Impulsionado por modelos	Impulsionado por atividades	Impulsionada pelo designer da interface
Envolvimento substancial dos usuários	Envolvimento seletivo do usuário	Envolvimento impositivo do usuário	Envolvimento perceptivo usuário
Estudos de utilização Design participativo O feedback dos usuários Teste de utilização	Modelagem exploratória Validação do modelo Inspeções de Usabilidade	Atividade seqüencial / Análise de Tarefas Design Intuitivo	Percepção visual com protótipos Design participativo Especificação Visual
Design por prototipagem iterativa	Design pela modelagem e prototipagem resumida	Design pelo entendimento atividades	Design pela alternância entre abstração e detalhamento
Muito variada, informal, ou “caixa preta” do processo de design	Sistemática, processo totalmente especificado, “caixa branca”	Não especificada, a regra de abertura da “caixa-preta” do processo de design	Depende da orientação e autoria
Design por tentativa e erro	Design por engenharia	Design por atuação	Design pela percepção visual

Quadro 6 - Comparação entre metodologias de IHC.

O ciclo de desenvolvimento deve ser criado sobre o núcleo dos métodos de todas as abordagens de design apresentadas, como, por exemplo, pontos como, a participação do usuário e o design sobre a atividade vistos em DCU e DCA, prototipagem, bem como modelagem leve com (casos de uso) cenários ou mapas de tarefas característicos de DCT. Todos os designers em um projeto precisam ter um entendimento similar da visão e da totalidade do sistema (DCA). Assim, a discussão contínua é necessária dando um aspecto prioritário a metodologia XP.

Comunicações informais através das fronteiras organizacionais devem ser facilitadas, e as equipes devem compartilhar escritórios e espaços comuns como sala de um projeto XP.

Em ES ágil, a escrita da documentação deve ser mantida a um mínimo suficiente. Anotações sobre noções abstratas exclusivamente através do uso de texto são difíceis. Uma das habilidades mais importantes para o designer de uma interface é a capacidade de tornar visíveis

idéias, permitindo que participantes olhem, sintam, analisem e avaliem. Basicamente, ambas as metodologias XP e MA (Modelagem Ágil) também tentam visualizar casos e tarefas o mais cedo possível.

Práticas de XP	Práticas de IHC
Iteração, pequenos incrementos	Prototipagem
Planejamento Jogo	<i>Focus Groups</i>
<i>Story Cards</i> , Cartões de Tarefas, Histórias do usuário	Cenários, Papel do usuário e Modelo de Tarefas
Participação do cliente	Design Centrado no Usuário, Manual de Participação
Teste, Teste de História	Avaliação, Inspeções de Usabilidade
Metáfora	Modelo conceitual e Mental, UI Metáfora

Quadro 7 - Similaridades entre XP e IHC.

O processo deve ser controlado por uma pessoa que deve ter um profundo entendimento de ambas as áreas de ES e IHC, confirmando que o XP e MA são os seus mais importantes critérios para o sucesso de um projeto (Beck, 2000). O líder navega pelo processo de desenvolvimento, propõe soluções para a concepção, critica, questiona e aplica o modelo adequado com relação à engenharia e os métodos de desenvolvimento. As lacunas entre a ES e IHC tornam-se menos significativas quando o especialista em IHC também é um programador e analista.

A opção pelo XP como sendo fundamental para os nossos pensamentos sobre integração entre ES e IHC, permite que através de seu princípio da programação em pares, pessoas com diferentes domínios de conhecimento, mas com capacidades em comum, projetem um sistema juntos.

Práticas de Metodologias Ágeis	Práticas de IHC
Prova com o código	Prototipagem
Criação de diversos modelos em paralelo, modelagem com outros	Modelagem simultânea, (Papel do Usuário - Tarefa - Conteúdo - Modelo)
Participação ativa dos <i>Stakeholders</i>	Design Centrado na Tarefa, Participação do Usuário, <i>Focus Group</i>
Considera a Testabilidade	Avaliação, Inspeções de Usabilidade
Disponibilização Pública de Modelos	Sala de Projeto

Quadro 8 - Similaridades entre MA e IHC.

A base da para a integração entre as disciplinas é, portanto, a identificação de semelhanças entre XP e IHC (Quadro 7), Metodologias Ágeis (MA) e IHC (Quadro 8), bem como DCA e DCT em relação ao IHC, MA e XP (Quadro 9). Para isso foram traçadas algumas semelhanças importantes, para comparação e destacadas outras relações entre essas disciplinas. Apesar de diferentes na sua formulação, princípios e práticas ágeis são comparáveis e mostra uma sobreposição significativa, como o design iterativo, pequenas versões, prototipagem, cartões com histórias de usuários e participação dos usuários e cenários, ou testes e avaliações. As abordagens modernas de design de interface não se contrapõem a colaborar com ES, ao contrário, salientam as correlações.

MA & XP Prática	IHC Prática	DCT / DCA & Prática
Único documento que é regularmente atualizado e de fácil manutenção.	Documentação através de modelos e gráficos compreensíveis.	Documentação minimalista, representação interativo.
Mostrar resultados iniciais, liberar pequenas versões.	Baixa-/Alta-Fidelidade prototipagem, Reutilização.	Tornar visíveis as idéias, protótipos abstratos e detalhados.
Comunicação em pequenas equipes, salas de projeto.	Salas de projetos e guias de estilos.	Comunicação informal, discussão.
Participação ativa de clientes	Participação colaborativa do usuário no projeto	Entendimento das idéias através da exteriorização
Desempenho do usuário	Desempenho do usuário, experiência do usuário,	Desempenho do usuário, experiência do usuário, qualidade estética.

Quadro 9 - Pontos em comum entre MA, XP com IHC e princípios e práticas de design de interação.

A explicação do modelo integrado se concentra nas questões que precisam ser trabalhadas de forma colaborativa por especialistas em IHC e em ES.

4.5. Considerações Finais do Capítulo

É possível definir a seqüência de desenvolvimento dos objetos de aprendizagem, iniciando-a pela identificação do núcleo conceitual que será trabalhado. Esses núcleos são conceitos, não necessariamente curriculares, da experiência em sala de aula dos autores ou da literatura disponível, e que são identificados como de difícil apreensão pelos alunos.

Escolhido o núcleo ao qual vamos nos concentrar, é preciso identificar qual o diferencial que o uso de recursos multimídias interativos pode trazer ao conceito. Em geral, quando uma estratégia de visualização, explicação ou exploração pode ser implementada sem perdas utilizando recursos instrucionais mais tradicionais (material didático impresso, aula expositiva, laboratórios e outros) e eles estão disponíveis aos alunos, essas estratégias devem ser evitadas. O objetivo é explorar ao máximo o potencial e o diferencial do computador na elaboração do aplicativo. A interface computacional introduz ruídos no aprendizado e, portanto, só é justificada quando seu uso possibilita um maior acesso ao conteúdo ou o uso de novas estratégias.

O passo seguinte é a definição de um objetivo primário do aplicativo e do público alvo. Como todo material instrucional, um objetivo instrucional deve ser uma sentença mensurável, que permita a verificação da eficácia do material no aprendizado. No caso específico de aplicativos, definimos um objetivo primário como sendo aquele que pode ser alcançado por qualquer aluno dentro do público-alvo definido que o utilize dentro do tempo planejado para o aplicativo. Para ser um bom objeto de aprendizado, ter uma granularidade bem definida, é importante que haja apenas um objetivo primário. Objetivos secundários podem aparecer ou serem definidos. Eles são metas alcançáveis por indivíduos fora do público-alvo definido, por subgrupos do público-alvo ou públicos que utilizem o aplicativo em um escala de tempo diferente daquela projetada. A importância dos objetivos secundários é fornecer ao aluno uma experiência única no uso do aplicativo.

É importante que o aplicativo traga várias camadas de interatividade em seu uso para que facilite o aprendizado. Portanto, é fundamental que sua construção inicie-se na definição dos recursos de interatividade que serão implementados. Controle da velocidade da animação ou apresentação de texto, controle de parâmetros de simulação, controle de opções de visualização, controle de situações de simulação são alguns dos exemplos possíveis de interatividade que foram explorados como *scaffoldings*. É desejável que cada recurso (ou conjunto de recursos) de interatividade implementado seja associado a um dos objetivos previamente definidos, nesse momento é que os *scaffoldings* previamente identificados são implementados no aplicativo.

A partir dessa etapa, o aplicativo começa a tomar forma propriamente dita. Definidos os recursos de interatividade, elementos multimídias (textos, imagens, animações, filmes e sons) devem ser associados a eles. Nessa associação a contextualização é fornecida ao aplicativo de forma a aproximá-lo do mundo real. Em seguida, tais elementos devem ser esboçados, a

programação necessária para interatividade precisa ser desenvolvida e finalmente associada aos elementos previamente esboçados. Em geral, os elementos multimídias são mantidos como esboços até o término e teste da programação. Essa fase de programação acaba restringindo certas características, portanto é importante manter a flexibilidade.

Terminada a fase de programação, os elementos multimídias finalmente podem ser refinados, com a ajuda de um designer. Este profissional também dá ao aplicativo um layout final. As características importantes desse layout final são a limpeza, clareza e fácil identificação dos recursos disponíveis. Depois de testado, o aplicativo é considerado pronto para ser aplicado a grupos de alunos e sua produção documentada.

Capítulo 5 - Estudo de Campo

Neste capítulo é apresentada a análise dos dados coletados através de entrevistas semi-estruturadas, explicitando as necessidades do curso de formação de controlador de tráfego aéreo sob a perspectiva dos instrutores. A análise foi realizada usando a metodologia de levantamento de requisitos proposta nesta pesquisa. O resultado da análise norteia a pesquisa dessa dissertação servindo como base para a aplicação do modelo de projeto instrucional proposto, assim como o desenvolvimento de um Objeto de Aprendizagem.

Basicamente, o estudo é realizado considerando os fatores que configuram os limites que impactam (e geram desvios ativos e latentes) em um sistema complexo. Para entender o trabalho, o sucesso e o insucesso (o erro) em situações reais, é necessário estudar os grupos de indivíduos inseridos no sistema que provém recursos e obrigações, ao invés de estudar a cognição do indivíduo isoladamente (Woods *et al.*, 1994). O estudo das situações operacionais reais permite identificar os elementos invariantes mobilizados pelos controladores no desempenho de suas funções e assim utilizá-los para a elaboração dos *scaffoldings*.

5.1. Seleção da Amostra

Nesse estudo, foi entrevistado um total de 03 instrutores, todos participantes do curso de formação de controlador de tráfego aéreo com sede em São José dos Campos, no Instituto de Controle do Espaço Aéreo.

A amostra representa o total da população de instrutores do curso em 2009, data desta pesquisa. Para manter a identidade dos entrevistados em sigilo, os participantes foram enumerados e os identificados pelas iniciais “I” (instrutores).

5.2. Instrumentos

Foram realizadas entrevistas semi-estruturadas, gravadas e posteriormente transcritas (em anexo), onde os participantes tiveram liberdade para expressar como vêem a disciplina Fundamentos da Meteorologia, observando os seguintes aspectos:

1. Importância da disciplina no curso e na atividade profissional;
2. Recursos didáticos utilizados;
3. Itens onde há maior incidência de dificuldades e suas razões;
4. Tratamento dado às dificuldades;
5. Sistema de fixação do conteúdo e avaliação;
6. O que poderia ser feito para prover melhorias no processo de ensino e aprendizagem.

Esta disciplina foi apontada pela coordenação do curso por requerer conhecimento prévio de temas da Física e Matemática no nível de Ensino Médio e porque apesar da alta tecnologia

presente nas aeronaves atuais não se pode ignorar os potenciais riscos oriundos das condições meteorológicas para a aviação.

Neste estudo foi considerada especificamente a Unidade 2.2 - Fundamentos da Meteorologia, que apresenta conteúdos acerca dos principais fenômenos meteorológicos e o modo como influenciam na segurança, regularidade e eficiência da navegação aérea. Nela são explorados os seguintes assuntos:

1. Calor e Temperatura;
2. Umidade na Atmosfera;
3. Nuvens;
4. Pressão Atmosférica;
5. Vento;
6. Atmosfera Padrão;
7. Altimetria; e
8. Altitude Densidade.

5.3. Procedimentos

Os participantes foram orientados quanto ao objetivo do estudo, o destino das informações fornecidas e o compromisso de que suas identidades seriam mantidas em sigilo, sendo as entrevistas agendadas e realizadas individualmente em local com privacidade.

5.4. Análise das Entrevistas

Esse estudo de campo detalhado foi realizado através da coleta de dados e aplicação do método GOMS segundo Lapolli e Cruz (2010 d) para a análise das falas dos instrutores e posterior levantamento dos requisitos necessários ao desenvolvimento do OA.

Metas (Goals)

Para alcançar os objetivos pedagógicos, cada elemento da interface, deve ser pensado e desenvolvido a partir da identificação e definição de metas sobre cada ação desempenhada no cenário operacional. Assim foram identificados alguns exemplos apresentados a seguir:

“(...) os tópicos que eles têm mais dificuldades são nuvens e pressão atmosférica.” (I1).

“Eles precisam conhecer o conceito e ter em mente pra quando chegar mais à frente como em mensagens meteorológicas e cartas de previsão ele bata o olho e já conhece.” (I1).

“Altimetria é uma disciplina que precisa de uma maior atenção onde eu creio que os alunos tinham maior dificuldade em aprender, mas nós temos desenhos, recursos com figuras. Isso é dado por nós e na parte de controle de tráfego porque é uma parte que vai ser muito utilizada na torre de controle.” (I2).

“Embora seja o meteorologista que tenham esta noção, que tá treinado pra isto, é interessante que eles tenham esta noção porque quando aparece pra eles uma mensagem meteorológica, é um código meteorológico. Na verdade são vários números. É interessante que eles saibam que aquilo é uma mensagem, identifiquem o ponto crítico atmosférico que consta naquela mensagem e saibam dialogar ou pelo menos entender as explicações de um meteorologista quando houver necessidade.” (I3).

Operadores (Operators)

Para alcançar as metas estabelecidas, cada elemento da interface deve ser pensado e desenvolvido a partir da identificação e definição de elementos e conceitos mobilizados pelos instrutores ou profissionais, durante a aula ou em ação desempenhada no cenário operacional. Assim foram identificados alguns exemplos apresentados a seguir:

“A maior dificuldade é em nuvens porque quando você começa a falar dos gêneros e olha para o céu, eles dizem: é tudo igual.” (I1).

“A temperatura, umidade e vento vai estar intimamente ligado com a pressão.” (I1).

“Tem que tá ligado porque ele tem uma atmosfera padrão, criada pelos americanos na década de 50 pra chegar a um denominador comum, ta todo mundo no ar e que eu brinco com os alunos. Tá todo mundo voando errado pra todo mundo chegar certo.” (I1).

“Geralmente eles tem uma dúvida e que a temperatura tem uma variação horizontal e na vertical também.” (I2).

“Altitude Densidade também é uma disciplina que precisa de um pouco de atenção. Porque eles vão ter que fazer uma relação entre atmosfera padrão e atmosfera real, assim como altimetria.” (I2).

“Uma coisa interessante também é que a gente fala sobre umidade relativa do ar e contenção de vapor de água na atmosfera. Então muita gente não tem esta noção de relação entre um e outra até mesmo, por incrível que pareça, por falta de base matemática.” (I3).

Métodos (Methods)

Para estabelecer uma relação entre elementos e conceitos mobilizados como operadores, as a interação dos elementos da interface deve ser pensada e desenvolvida a partir da identificação e definição das estratégias adotadas pelos instrutores ou profissionais durante a aula ou em ação desempenhada no cenário operacional. Assim foram identificados alguns exemplos apresentados a seguir:

“Levo, lá mostro os equipamentos, os termômetros, mostro a área de trabalho fora da estação né?! Porque você falar para um aluno de tráfego aéreo sobre temperatura, calor (...) então antes de começar a aula eu pego o pessoal levo para a área externa, peço pra eles olharem o céu, (...) e falo os conceitos básicos de calor tempestade, falo da umidade, aí quando eu entro na unidade aqui eu já faço menção de nuvem e vou discorrendo.” (I1).

“Uso muito desenho, muitas fotografias do Brasil principalmente de áreas como do Sul, Amazonas para poder relacionar com eles e eles irem assimilando porque a carga dessas informações pra eles.” (I1).

“Quando chega em nuvens estes conceitos de temperatura, do decréscimo de temperatura com elevação de altitude, umidade que são coisas intimamente ligadas que vão gerar as nuvens. Se faltar uma das condições o ciclo não chega a gerar a nuvem e vou explicando os gêneros, as espécies.” (I1).

“Então a gente usa o quadro negro e faz transformação de temperatura de Celsius para Fahrenheit mostrando a regra básica. A gente mostra pra eles poderem assimilar pra não ficar aquilo mecânico.” (I1).

“Informações meteorológicas a gente tem a REDEMET que é um órgão oficial que dispõe de todas as informações meteorológicas (a todo minuto eles têm informações para que o vôo seja tranquilo que é nosso objetivo)”. (I1)

“A gente consegue chegar numa atmosfera real usando calor, umidade, nuvens para chegar em altimetria e eles terem (faz um desenho, pergunta de que localidade é) noção do que uma informação errada pode ocasionar (usa bastante imagem, bastante detalhes para eles). (I1).

“Muitas vezes nós trazemos para sala de aula algo que eles possam enxergar. Por exemplo, por calor trazemos termômetro (de solo, de psicômetro). Então quando a gente vai dar este tipo de disciplina procura levar alguma coisa sólida, concreta para que eles possam ter uma visualização. Ou então, também, passar algum vídeo a respeito de calor.” (I2).

“Tudo bem que eles vão utilizar isso na prática, mas o que a gente pode estar mostrando de mais concreto são os vídeos que passamos pra isso (tempestade, tornado, furação).” (I2).

“Por exemplo, Simulador. Ali ele vai fazer experiências a respeito de controle do simulador. Já pra estas disciplinas mais teóricas a gente fica (...)” (I2).

“Quando a gente fala sobre as nuvens é uma coisa que chama bastante atenção deles porque a gente joga dois tipos de slides de uma mesma nuvem porque às vezes ela aparece assim de uma forma diferente da outra mas a gente como meteorologista sabe identificar ela. Quando a gente acaba de falar isso aí, no próximo intervalo a gente sai da sala de aula e eles ficam “cadê a nuvem?”. Eu acho interessante este comportamento com relação ao que a gente mostra (a parte científica da coisa) eles procuram relacionar com a prática do dia-a-dia.” (I3).

“Essa parte de pressão atmosférica a gente mostra o barômetro, tem até um slide legalzinho que mostra a variação da coluna de ar atuando num barômetro e a variação dos ponteiros de marcação da pressão.” (I3).

Regras de Seleção (Selection Rules)

Para estabelecer uma hierarquia entre operadores e métodos é necessário identificar e definir regras de seleção adotadas pelos instrutores ou profissionais durante a aula ou em ação desempenhada no cenário operacional e que irão orientar a sequência lógica de interação a ser

estabelecida em uma interface. Assim foram identificados alguns exemplos apresentados a seguir:

“(...) eu ensino aos alunos de tráfego aéreo identificar as nuvens.” (I1).

“A gente com a experiência do dia-a-dia de ser observador de que a nuvem cumulus quando vai se aproximando do aeródromo a pressão cai, a intensidade do vento começa a variar, já não tem aquela velocidade constante, tem picos que é chamado rajadas, que interfere na pressão atmosférica.” (I1).

“Ele não atentou que já mudou que deixamos de falar de uma coisa e já estamos falando de outra. Essas são as dificuldades que eu vejo. Geralmente alguma coisa mais complexa, mas difícil de identificar como, por exemplo, a variação da densidade com relação a temperatura.” (I2).

“A gente procura, principalmente pra tráfego aéreo, dar a causa que aquele fenômeno pode dar na aviação, no procedimento, principalmente de pouso e decolagem também.” (I3).

“O principal para o tráfego aéreo é base de nuvens e visibilidade horizontal. É o que restringe as operações de pouso e decolagem. São esses parâmetros que vão fazer com que o trabalho deles flua bem ou flua mal.” (I3).

“Essa é um aparte que eles têm bastante dificuldade porque é uma coisa que eles não vêem no dia-a-dia, que é diferença de pressão. Então nós mostramos que existem parâmetros referenciais para vôo.” (I3).

5.5. Considerações Finais do Capítulo

Estudos sobre treinamento e processos de aprendizagem têm mostrado, mais especificamente, que o processo de aprendizagem humana se consolida quando salta da teoria para prática principalmente no aprendizado em sistemas complexos onde a explicação sobre um fato ou conceito reside e emerge da relação entre elementos variantes e invariantes que compõem o sistema. Desse modo, ao considerar as tarefas do futuro controlador de tráfego aéreo a exigência de combinar teoria e prática será cada vez mais importante.

Neste sentido, a análise dos dados aponta para algumas áreas do curso de formação de controlador de tráfego aéreo onde há uma necessidade de melhora a médio ou longo prazo. Assim, para fins desta pesquisa de dissertação foram consideradas as seguintes áreas:

- Atualização do atual modelo pedagógico, no que diz respeito ao modo de elaboração do Plano de Unidades Didáticas e recursos educacionais;
- Uso de mecanismos e/ou recursos educacionais que permitam identificar os conceitos relevantes para a segunda etapa do curso (simulação) e posterior atividade profissional;
- Utilização de recursos que permitam reduzir ou eliminar lacunas conceituais oriundas de cursos anteriores;
- Incluir atividades de apoio a compreensão e fixação do conteúdo, bem como instrumento de preparação para a avaliação do aluno;
- Fazer uso da contextualização a fim de aproximar a teoria da prática, ao possibilitar a percepção da real utilização do conhecimento adquirido ao longo do curso.

Para auxiliar o processo de melhoria no curso de formação de controlador de tráfego aéreo, as áreas citadas anteriormente poderiam beneficiar-se das atuais tecnologias e estratégias educacionais.

Capítulo 6 - Metodologia de Desenvolvimento de OAs Baseados em Simulações

Neste capítulo é apresentada uma proposta de metodologia de desenvolvimento de Objetos de Aprendizagem baseados em simulações a ser aplicada no levantamento de contextos de aprendizagem para apoiar a compreensão do conteúdo teórico, com vistas a reduzir ou eliminar as lacunas conceituais para o seu correto entendimento. Além do método de levantamento de requisitos apresentado nas considerações finais do capítulo 2 e cujas aplicações foi descrita no capítulo 5, o restante da proposta desta dissertação apresentada é descrita neste capítulo. A proposta compreende um modelo de projeto instrucional; um modelo de projeto de interface e uma metodologia de desenvolvimento integrando métodos ágeis de engenharia de software e conceitos de interface humano-computador.

No capítulo também é apresentado o objeto de aprendizagem desenvolvido com base na metodologia proposta, evidenciando a especificação e as etapas de sua produção.

A produção de um OA é bastante complexa (Longmire, 2001), pois envolve a participação de uma equipe multidisciplinar, composta por pedagogos, desenvolvedores, designers gráficos e especialistas de área, também denominados de conteudistas. Esses profissionais devem interagir de modo a se atingir os objetivos tanto tecnológicos quanto pedagógicos desses produtos. Desse modo, a precisão na forma de comunicação das informações relacionadas aos OA deve ser garantida, a fim de permitir a sua construção com qualidade e minimizando o tempo de produção dos mesmos.

Essa comunicação entre profissionais de diversas áreas é um dos maiores desafios dos modelos de processo que tratam desse tipo de produto. Os modelos de processos disponíveis atualmente para a produção de OA, além de não contemplar de forma efetiva as peculiaridades das relações multidisciplinares, ainda baseiam os mecanismos de comunicação em documentos escritos em linguagem natural. Acontece que profissionais de diferentes áreas podem compreender esses documentos de formas distintas.

No desenvolvimento de um objeto de aprendizagem para processo de ensino e aprendizagem da disciplina Fundamentos da Meteorologia, percebeu-se a necessidade de inovação no modelo de projeto instrucional e de desenvolvimento. Motivados também pela expansão e uso das técnicas de desenvolvimento ágil de software, verificou-se que a aplicação de suas práticas poderia incorporar melhorias no projeto instrucional e, conseqüentemente, na produção de recursos didáticos.

Diante da multidisciplinaridade das equipes que produzem OA, e das dificuldades em estabelecer uma comunicação clara entre seus membros, foi adotada uma nova estratégia de desenvolvimento de OA baseados na integração entre as metodologias ágeis de desenvolvimento de software e os conceitos de interface homem-computador.

Como já citado anteriormente ao longo deste trabalho, o desenvolvimento dos OA envolve aspectos técnicos (engenharia de software, design, interface humano computador, usabilidade) e pedagógicos (conteúdos educativos, concepções de aprendizagem, estrutura pedagógica) e envolve profissionais das mais diversas áreas do conhecimento. Esta multidisciplinaridade das equipes de desenvolvimento acaba por dificultar a participação efetiva dos profissionais com conhecimentos não técnicos, em termos de produção de software, a se engajarem de fato no processo de desenvolvimento dos OA. Essa dificuldade é acentuada no desenvolvimento de OAs dirigidos ao treinamento de operadores de sistemas complexos devido à necessidade dos desenvolvedores incorporarem ao software características específicas do ambiente operacional

desses alunos. Na representação desse ambiente constituído por linguagens técnicas, a participação das partes envolvidas no processo é fundamental para a certeza da eficácia do aplicativo.

A partir do exposto e das proposições apresentadas nas considerações finais de cada capítulo deste trabalho, são propostos mecanismos para minimizar esse problema e permitir que as abstrações necessárias para representar os conhecimentos de cada profissional envolvido na criação de OA sejam representadas através da utilização de cenários e protótipos até a versão final do OA. Esta proposta é baseada na integração de metodologias ágeis de desenvolvimento de software e conceitos de interface humano-computador, resultando no modelo de processo e um conjunto de técnicas e ferramentas de desenvolvimento de OA que possibilitam a representação e integração dos conhecimentos de todos os envolvidos na elaboração desses produtos, e que, além disso, permita que essas representações possam ser tratadas em todo o processo de desenvolvimento, possibilitando o envolvimento necessário dos profissionais, para se alcançar a qualidade requerida por esses produtos em todas as suas formas.

6.1. Metodologia Integrada de Desenvolvimento Ágil de ES e IHC

Fase 1 - Definição dos Requisitos Iniciais

O modelo integrado proposto neste trabalho começa com os requisitos iniciais (Figura 5), levantados a partir da aplicação do método GOMS proposto nas considerações finais do Capítulo 2 deste trabalho e a exemplo, aplicado no estudo de campo apresentado no Capítulo 5. O tempo de desenvolvimento da metodologia ágil pode ser preservado se os métodos empregados forem classificados como ágeis e interdisciplinares. No que diz respeito ao design da interface do usuário, XP e MA não são suficientes e tem que ser aplicado junto ao desenvolvimento prático do design da interface (DCT, DCA).

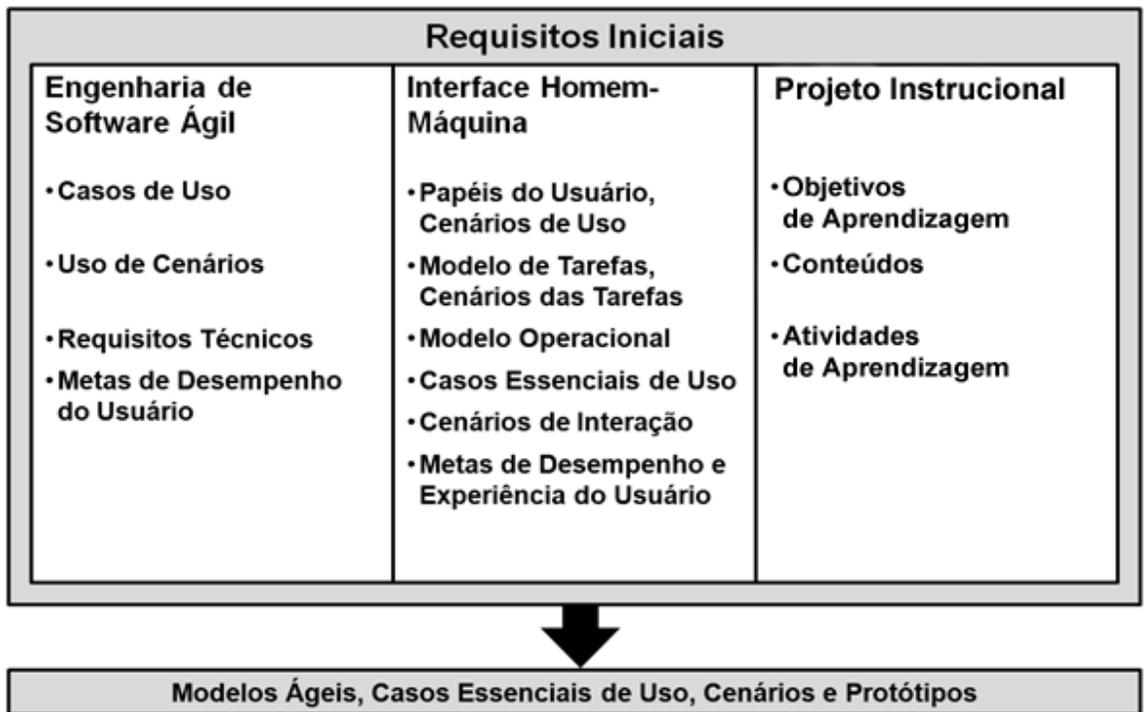


Figura 5 - Requisitos iniciais para o desenvolvimento do software

Nessa fase, a coleta e análise dos dados apurados com os *experts* servem para levar a visão deles ao projeto e assim identificar os principais elementos mobilizados por eles na realização das tarefas operacionais e que podem facilitar a compreensão e trabalhar o processo de aprendizagem dentro do aplicativo.

O modelo baseado em ER proposto nas considerações finais do Capítulo 2 deste trabalho centra-se no levantamento de informações essenciais e em adequar-se a uma metodologia de desenvolvimento ágil devido ao uso de cartões indicativos (Quadro 10). Nesses cartões, as funções dos usuários são priorizadas (Funções Focais do Usuário) a partir da fase de levantamento de requisitos com aplicação do método GOMS e classificadas em relação ao seu impacto sobre o sucesso do processo de aprendizagem sempre observando a estrutura hierárquica em que estão organizadas, para assim oferecer e retirar gradativamente os *scaffoldings*. Essa proposta foi desenvolvida a partir dos pressupostos apresentados nos Capítulos 2 e 4, onde é proposto que através da identificação dessas funções é possível estabelecer uma hierarquia de linguagem e interação criando uma estrutura de apoio ao aprendizado através de atividades e tarefas, onde a relação entre os níveis será desencadeada através de mecanismos de interação homem-máquina.

Finalmente, casos essenciais de uso descrevem as tarefas do usuário e permitem a construção de modelos de tarefas e mapas de tarefas, onde essa forma de intervenção do aluno no ambiente será determinada. É possível assim especificar um modelo instrucional com o processo de interação sendo dirigido ao comportamento. Sempre que necessários cenários são estabelecidos para a compreensão, comunicação e contextualização da situação problema. Aspectos importantes do local de trabalho do usuário e contexto são documentados no modelo operacional. Para um entendimento compartilhado de desenvolvedores e de comunicação com os interessados, todos os modelos são traduzidos em cenários, que podem incidir sobre os diferentes aspectos do design da interface (usuários, tarefas e interações).

Métodos	Descrição
Card Sorting	Método rápido para reunir os requisitos em cartões de índice de acordo com a hierarquia de prioridades (DCT)
Brainstorming	Similares a "modelo com os outros" na MA
Focus Groups	Relacionado com o que XP chama de o Jogo do Planejamento Ágil, para que o número de participantes seja mantido baixo e a documentação reduzida ao essencial
Tarefa / Cenários de uso	Story Cards, Task Cards, User Stories

Quadro 10 - Métodos de ER (Engenharia de Requisitos) classificados como ágeis.

Metodologias ágeis não consideram a interface do usuário em detalhes, por isso não reconhecem extensas guias de estilo, utilizadas nas práticas de IHC. É sugerido, portanto, leves guias de estilo que são mais curtas, mais relevantes e contém padrões de interface do usuário.

Padrões de IHC podem ser aplicados em correlação com os princípios de MA aplicando estruturas padrão e usar os recursos existentes. Eles facilitam o processo de design, fornecendo elementos de interfaces já reconhecidas pelos usuários e o aproveitamento de experiências prévias através do uso de metáforas representativas do ambiente operacional que são utilizadas como *scaffoldings*.

Uma das contribuições importantes de IHC é apoiar o desenvolvimento de cenários em experiências prévias, utilizando elementos do cotidiano operacional com o objetivo de criar uma conexão entre esse conhecimento já adquirido pelo aluno e o novo conhecimento a ele apresentado.

Durante a fase de levantamento inicial dos requisitos, as equipes de IHC e de ES devem apoiar a finalização da análise dos requisitos com os primeiros debates sobre cenários e alternativas de design.

Ao contrário de outros ciclos de desenvolvimento existentes na literatura sobre IHC, nesse modelo integrado são previstas as representações gráficas, mesmo antes de terminar a análise dos requisitos. Essa quebra com práticas comuns de IHC permite que o designer de interface decida mais cedo sobre o grau de envolvimento do usuário e a necessidade de soluções mais inovadoras.

O designer de interface pode ter uma influência considerável no equilíbrio das metas de desempenho do usuário, nas metas de experiência do usuário e na manutenção das demandas de qualidade e pode guiar o levantamento inicial dos requisitos nas adequações.

O resultado do levantamento inicial dos requisitos são modelos ágeis que descrevem as necessidades do usuário e os objetivos da tarefa. Através da aplicação de cenários e métodos de prototipagens iniciais, primeiras visões de design são criadas e refinadas durante a fase seguinte.

Fase 2 - Fase Conceitual Inicial

A segunda fase do processo de desenvolvimento é a fase conceitual inicial (Figura 6). No CIP (Ciclo Inicial de Produção) é previsto uma separação entre o protótipo da interface do usuário e o protótipo da arquitetura do software, sempre que possível para acelerar o processo de desenvolvimento.

A aplicação consciente de padrões de software facilita este procedimento. O desenvolvimento da interface e a arquitetura do sistema podem ser desenvolvidos em paralelo, desde que de forma minimalistas com especificações padrão geradas para em seguida identificar outros componentes de interface necessários a serem desenvolvidos.

Pontos de dependências entre interface e arquitetura do sistema podem ser encontrados com a ajuda de casos e cenários de tarefas estabelecidas na fase levantamento de requisitos. É muito provável que interfaces de usuário altamente interativas tenham maior impacto sobre a concepção da arquitetura do sistema.

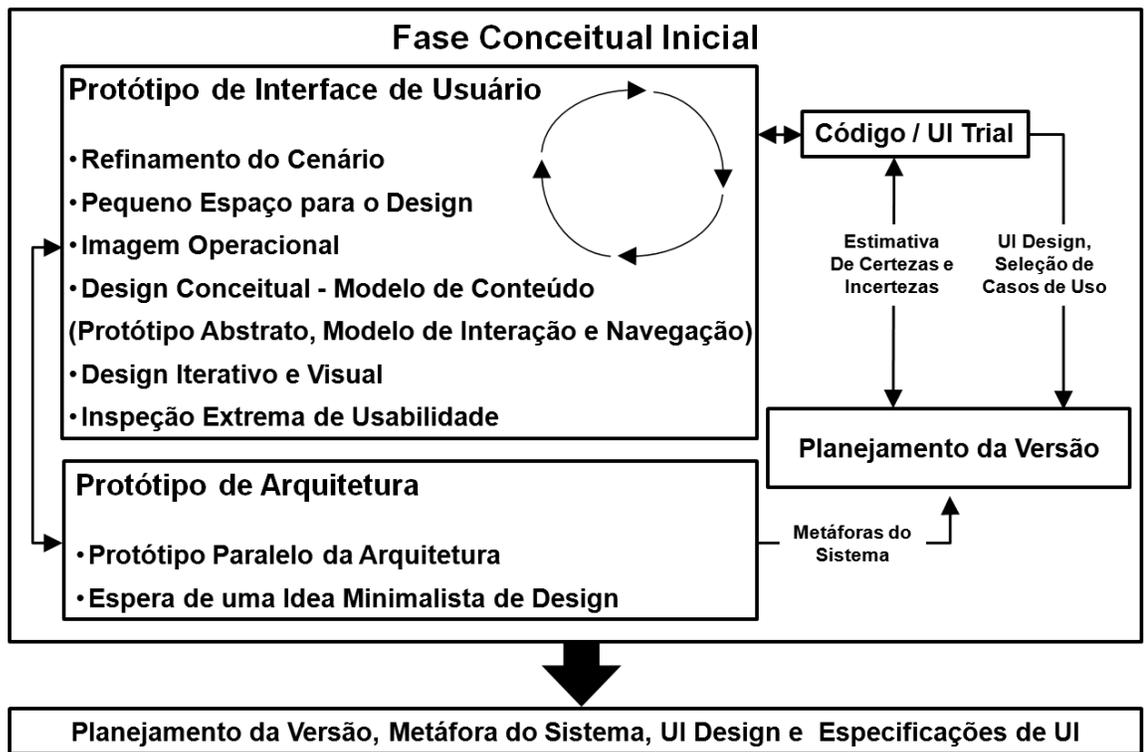


Figura 6 - Ciclo Inicial de Produção do software

Como foi observado, protótipos são práticas comuns em IHC e ES. O objetivo geral do Ciclo Inicial de Produção (CIP) é a geração de protótipos mais detalhados e interativos para reduzir as opções de design para uma única solução através do debate com as partes envolvidas no projeto e através do refinamento do cenário. Para esta tarefa, o designer deve ficar entre o abstrato e níveis detalhados de protótipos, sempre considerando se os prazos são adequados para um ambiente de projeto ágil (Figura 6). Levando em conta as diretrizes dos métodos ágeis, os protótipos devem ser fáceis de trabalhar e, acima de tudo, de rápida produção, de fácil manutenção já com o design de interação aplicado.

Quanto mais interativas e complexas as representações externas, melhor para o designer conduzir um debate com os desenvolvedores sobre soluções de interação e idéias.

Protótipos que são visualmente mais detalhados ajudam a superar as limitações da capacidade cognitiva para processar, desenvolver e manter idéias complexas resultando em uma detalhada imagem operacional, o design inicial da interface.

Apenas testes detalhados permitem uma avaliação real do comportamento da interface, portanto, são necessárias para a verificação da integração da interface do usuário com os

requisitos. Enquanto o protótipo pode ser modificado usando técnicas simples de manipulação direta, os usuários podem ser ativamente envolvidos em um processo participativo de desenvolvimento. É recomendável, portanto a criação de protótipos de ferramentas que possam ser aplicados em ambientes ágeis de acordo com as exigências descritas. Eles são fáceis de usar por todas as partes do projeto, devido à pouca necessidade de codificação, permitem a reutilização de componentes através da aplicação de padrões ou modelos e produzem rápidos testes interativos que podem ser melhoradas para pequenas versões.

Protótipos interativos podem também funcionar como soluções utilizadas para avaliar e comprovar a funcionalidade e interoperabilidade dos conceitos de interface e da arquitetura do sistema. O mais importante é que eles podem ser aplicados como especificações visuais e interativas e serem utilizados subsequentemente na fase de construção do software.

Especificações visuais como protótipos de alta fidelidade são inequívocos e podem garantir os objetivos finais do sistema correspondendo às expectativas dos usuários sobre o design da interface e o comportamento do software.

Voltando para a fase de construção (figura 6), o desenvolvedor pode utilizar a especificação visual sempre que documentos convencionais de especificação não descrevem requisitos coerentemente ou de forma clara. A prototipagem, portanto, minimiza o risco de tomar decisões erradas de projeto durante a programação e abre caminho para uma solução de design bem sucedida.

Através da aplicação bem-equilibrada e ponderada de métodos selecionados de modelagem de ER como modelos abstratos, detalhando protótipos e avaliações ágeis de usabilidade, o método integrado evita um projeto por tentativa e erro e faz o projeto avançar de forma rastreável e racional. O processo de identificação da solução de projeto mais promissora é guiado por avaliações de interface do usuário, que podem ser mantidos em baixa complexidade de desenvolvimento.

Fase 3 - Construção e Teste do Protótipo

Ao entrar na fase de Construção e Teste do Protótipo (CTP), a codificação é iniciada (Figura 7). Nesta fase, o ciclo de desenvolvimento do método integrado se assemelha a forma incremental e iterativa do XP. CTP, portanto, começa com o planejamento da interação e na

criação de unidades de testes de aceitação, que são posteriormente utilizados para avaliar as partes da arquitetura do sistema e da interface do usuário. As funcionalidades mais específicas previamente definidas ou metas de qualidade são consideradas, porém elas só devem ser desenvolvidas se a interação projetada for identificada como necessária ao processo de aprendizagem. É recomendada, portanto, a integração do pessoal de IHC no desenvolvimento da programação em par.

Tal como acontece com a construção de protótipos, o código final do design da interface e a arquitetura do sistema novamente ocorre em paralelo. Componentes da interface do usuário que têm grande impacto podem ser desenvolvidos mais rápido inicialmente e depois refinados durante as seguintes interações.

Como no XP, a fase de construção termina com a implantação de uma pequena versão. Durante as primeiras interações, os lançamentos de pequenas versões provavelmente terão o caráter de protótipos horizontais, com uma profundidade cada vez maior de forma incremental das funcionalidades.

Antes do início da próxima fase, cada pequena versão pode ser novamente avaliada usando métodos baratos e rápidos. Se questões de usabilidade ou novas funcionalidades são identificadas em uma das fases do projeto, podem também ser documentadas em cartões indicativos (cartões de “defeito”) onde cada defeito é atribuído como correspondente ao seu caso de tarefa ou cenário de uso. Se ocorrerem problemas de usabilidade ou design, as equipes de IHC e ES juntamente com as partes envolvidas no desenvolvimento, podem decidir sobre as medidas necessárias para readequar o projeto.

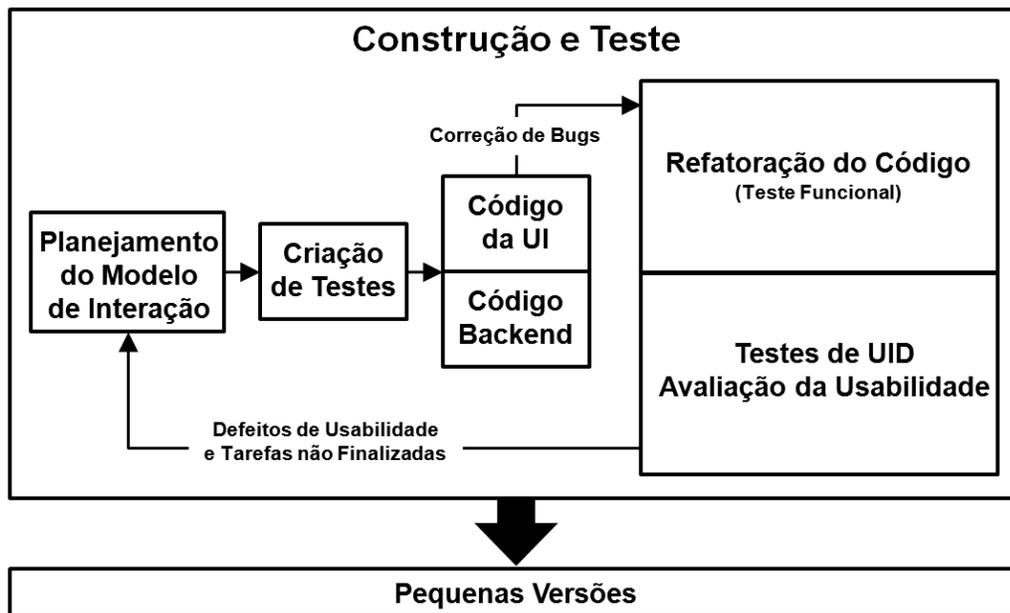


Figura 7 - Ciclo de construção e teste de pequenas versões

Fase 4 - Implementação

A última etapa do ciclo de desenvolvimento do modelo integrado é a implementação e a fase de produção (Figura 8). Enquanto os usuários estão trabalhando com o sistema, novas funcionalidades podem ser solicitadas, ou questões de usabilidade e de design que foram pouco detalhadas durante as iterações podem ser levantadas. O ciclo de desenvolvimento, portanto, permite um retorno a fases anteriores para atender a essas novas exigências.

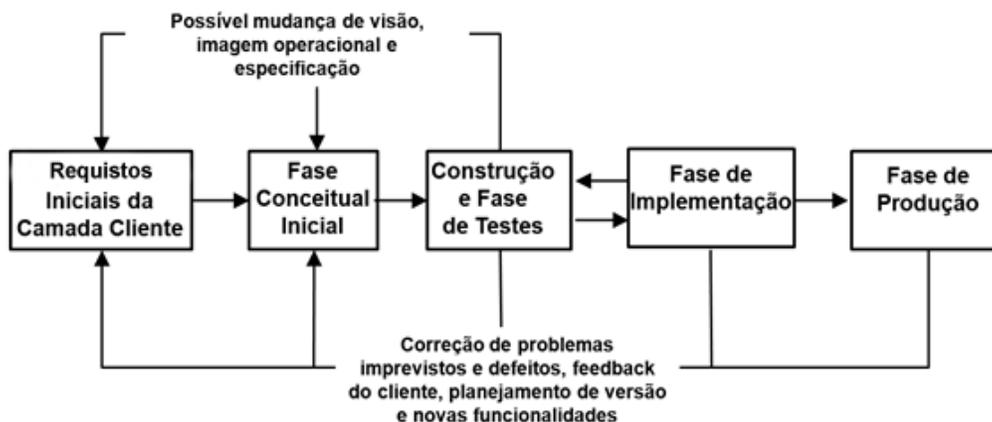


Figura 8 - Ciclo de desenvolvimento contínuo

6.2. Modelo de Projeto Instrucional

A fim de identificar os conhecimentos que deveriam ser abordados neste trabalho, foram realizadas entrevistas semi-estruturadas no Instituto de Controle do Espaço Aéreo (ICEA).

Através das entrevistas foi possível apurar as dificuldades em se construir cognitivamente um modelo do momento em que os conceitos mais abstratos de Meteorologia influenciam significativamente nos procedimentos operacionais.

A elaboração do projeto instrucional da disciplina Fundamentos de Meteorologia foi desenvolvido através do modelo integrado apresentado no Capítulo 4 deste trabalho. O modelo tem como base do desenvolvimento ágil a adaptação do *Behaviour-Driven Development* (BDD) como uma das atividades de um projeto instrucional e de desenvolvimento, possibilitando a identificação dos cenários que permitam ao aluno interiorizar o conceito de tal forma que seja possível a sua abstração e aplicação em diferentes situações, inclusive as não previstas durante o seu treinamento.

Cohn (2004) afirma que a história escrita por um usuário, descreve o que é valioso para os interessados de um sistema ou software e traz em si três características:

1. Descrição escrita da história, usada para planejamento e como lembrete;
2. Conversações sobre a história que servem para aprofundar os detalhes;
3. Testes e documentos que transmitem informações que podem ser utilizados para determinar quando uma história está completa.

Usando o acrônimo em inglês INVEST, Cohn (2004) diz que uma boa história do usuário é: **I**ndependente; **N**egociável; **V**aliosa ao comprado; **E**stimável; **S**mall (pequena); **T**estável.

Na escrita da história do usuário foi utilizado o seguinte modelo:

Como um [pessoa ou papel desempenhado]

Eu quero [funcionalidade]

Para que [benefício ou valor dessa funcionalidade ao negócio]

<p>como um controlador de tráfego aéreo eu quero interpretar ajuste padrão do altímetro para orientar corretamente os pilotos</p>
--

Figura 9 - Exemplo de *Story Card* usado no desenvolvimento do OA (Lapolli, Cruz *et al.* 2009 b)

Uma vantagem significativa de *story cards* (Figura 9) é que são frequentemente escritas utilizando artefatos muito simples, como por exemplo, fichas.

A identificação de novas histórias de usuário ou “particionamento” de histórias já existentes ou, ainda, a retirada de histórias que não são mais consideradas parte do escopo é uma atividade de processo considerada relativamente fácil. Essas histórias podem ser usadas para descrever uma grande variedade de requisitos. As histórias do usuário não são simplesmente uma versão reduzida de um caso de uso. Embora sejam semelhantes, o que os difere é que nas histórias do usuário, são definidos também os critérios para a sua aceitação (*Acceptance Criteria*). Uma história do usuário só estará pronta quando todos os seus critérios de aceitação forem atendidos.

David Churchville (2008) apresenta alguns princípios para escrever boas histórias do usuário. Elas devem: i) estar focadas naquele que necessita da solução; ii) ser perfeitamente explicáveis em 30 segundos; iii) “caber” no trabalho a ser executado em uma semana pela equipe de desenvolvimento e iv) ser facilmente testáveis.

O comportamento de um sistema constitui o seu critério de aceitação. Isto é, se o sistema cumpre todos os critérios de aceitação assume-se que ele se comporta corretamente. Caso contrário, não. Assim, North (2007) aponta que um padrão para os critérios de aceitação da história do usuário é bastante livre, a fim de não ser artificial para os analistas, mas suficientemente estruturado de modo a permitir a divisão da história nos fragmentos que a compõem e automatizá-los. Em termos de cenários, os critérios de aceitação tomaram a seguinte forma:

Dado algum contexto inicial (os dados),

Quando ocorre um evento,

então assegure alguns resultados.

Em **dado** é definido tudo o que o usuário precisa antes para **quando** o evento ocorrer **então** verificar o resultado. Portanto, uma funcionalidade pode admitir N-cenários. (Figura 10)

Funcionalidades: Controlar o Tráfego Aéreo

Como um controlador de tráfego aéreo
Eu quero obter informações meteorológicas e climáticas
Para utilizar estas informações em outras atividades do Sistema

Cenário 1: Ajustar Altimetro

Dado que estou em vôo nivelado
E vou iniciar o procedimento de sua aproximação para pouso
E defino "altitude"
E defino "pressão atmosférica"
E defino "temperatura"
Quando "reunir" as informações altitude +pressão atmosférica+temperatura
Então preciso ver "o valor a ser ajustado no altímetro"

Cenário 2: Identificar Altitude

Dado que temos uma atmosfera padrão
E defino "pressão padrão"
E defino "nível médio do mar" (QFF)
Quando "reunir" as informações pressão padrão + QFF
Então preciso ver "a distância vertical que separa uma

Figura 10 - Exemplo de história de usuário (Lapolli, Cruz *et al.* 2009 b)

O uso deste padrão fornece uma linguagem ubíqua para a área de negócios e a de tecnologia, possibilitando a escrita de melhores softwares em que o foco está na essência, no que realmente importa.

O modelo proposto foi idealizado levando em consideração os pressupostos de boas práticas das metodologias ágeis de desenvolvimento, mais especificamente o desenvolvimento dirigido pelo comportamento esperado do sistema, que visa aplicar testes unitários, funcionais e de integração para descrever como uma determinada funcionalidade deveria trabalhar. Sendo assim, o modelo foi baseado no padrão de teste de aceitação definido por North (2007) visando atender algumas expectativas e necessidades apontadas por instrutores e alunos durante as entrevistas semi-estruturadas. Assim é definido o Modelo de Projeto Instrucional Dirigido pelo Comportamento, composto por cinco etapas: (1) Programa Oficial da Disciplina; (2) Proposta de Atividades; (3) Plano de Atividades; (4) Formas de Mediação; (5) Análise do Processo e do Produto. (Figura 11)



Figura 11 - Modelo de projeto instrucional dirigido pelo comportamento (Lapolli e Cruz *et al.* 2009 b)

1ª Etapa: Programa Oficial da Disciplina

Nesta etapa a equipe responsável pela área de Treinamento e Desenvolvimento deve identificar os conteúdos (conhecimentos/habilidades) que permitirão alcançar os objetivos (resultados) esperados. Esta combinação é chamada de Programa Oficial da Disciplina.

Cursos que visam preparar profissionais para atuar em Sistemas Complexos contêm no seu programa oficial, diversas disciplinas distribuídas por vários grupos disciplinares. Cada disciplina ocupa um lugar próprio na formação básica destes profissionais, porém o programa oficial deve favorecer a aquisição simultânea de saberes oriundos de diferentes áreas dado sua característica multidisciplinar e interdisciplinar.

O produto desta etapa é, portanto, um Programa não linear (em alguns momentos) que possibilite diferentes percursos de aprendizagem, com vistas a potencializar os resultados a partir do perfil de cada grupo de profissionais.

2ª Etapa: Proposta de Atividades

Da proposta de atividades de treinamento e desenvolvimento dos profissionais que irão atuar em Sistemas Complexos importa salientar que ela precisa atender aos conhecimentos, habilidades e comportamentos desejados para a realização de procedimentos operacionais previstos em sua rotina de trabalho.

Durante esta etapa são definidos os objetivos de aprendizagem específicos para cada assunto; os conteúdos que serão desenvolvidos; e as atividades de aprendizagem. Cabe salientar que as atividades de aprendizagem devem estar fundamentadas nos conceitos adequados tanto

para o ensino do conteúdo quanto para as habilidades e comportamentos a serem desenvolvidos. (Figura 12)

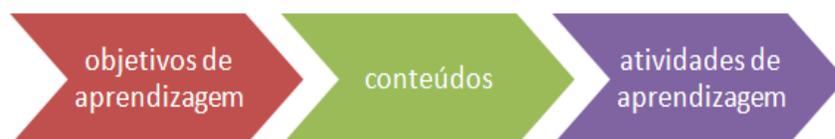


Figura 12 - Produtos da etapa proposta de atividades

Uma vez elaborada a proposta de atividades é possível iniciar a etapa de identificação dos cenários que serão utilizados para contextualizar os conhecimentos necessários para a realização das tarefas, apresentada a seguir.

3ª Etapa: Definição dos Cenários

Segundo definição do dicionário Aurélio, cenário é o lugar onde decorre uma ação ou parte da ação de uma peça, romance, filme etc. Sob o ponto de vista educacional, cenários são representações realistas (simulações) com a finalidade de prover a experimentação e exploração de fenômenos (hands-on). Nesta etapa são recomendadas algumas atividades da análise do trabalho cognitivo como, por exemplo, entrevistas com *expert* e observação *in loco*.

Existem diferentes métodos e ferramentas para realizar a análise do trabalho cognitivo. Para que atenda a expectativa de construção de cenários educacionais, os métodos e ferramentas utilizados na identificação e definição de cenários devem incluir o mapeamento das tarefas, identificando e priorizando os pontos de decisão crítica. A Figura 13 mostra os produtos desta etapa. Eles são bastante úteis para explicitar como um conceito deve ser apresentado, de maneira tal que facilite a compreensão de sua aplicação prática.



Figura 13 - Produtos da etapa definição dos cenários

Estes produtos são de extrema importância para a escolha das formas de mediação e o ponto de partida para suas respectivas construções.

4ª Etapa: Formas de Mediação

Dentre as ações delineadas no modelo de projeto instrucional dirigido pelo comportamento, conta a escolha e construção das formas de mediação que serão utilizadas para apoiar e/ou facilitar a aprendizagem dos que almejam uma formação profissional com mais eficiência, eficácia e agilidade. Assim, nesta etapa, a equipe de Treinamento e Desenvolvimento especifica e cria os recursos educacionais adequados para facilitar o ensino e aprendizagem de profissionais para atuar em Sistemas Complexos. De modo semelhante à etapa Proposta de Atividades, uma mesma *storyboard* pode dar origem a diferentes recursos educacionais. Além disso, o modelo possibilita combinar diferentes teorias de aprendizagem em uma mesma forma de mediação.

5ª Etapa: Análise do Processo e do Produto

Embora o termo análise seja freqüentemente utilizado para designar a fase inicial de um processo, no modelo de projeto instrucional dirigido pelo comportamento ele representa nossa constante busca pelos interesses e aspectos relevantes a serem considerados na formulação e construção de um curso de formação profissional. Nesta etapa, quanto mais precisas forem as informações coletadas sobre o funcionamento do Sistema e o comportamento esperado para os profissionais que nele atuam, maiores são as possibilidades de planejar de forma mais plena e consciente a disponibilização da disciplina, o que ela pode oferecer e as estratégias mais eficazes para facilitar a aprendizagem. Uma das vantagens do modelo proposto é permitir a análise recursiva do processo e do produto permitindo incorporar melhorias ao longo do processo, o que viabiliza um processo de educação flexível, através de diferentes recursos que despertem no aluno o interesse pelo seu auto desenvolvimento.

A fim de identificar os requisitos e funcionalidades que deveriam ser abordados nesse OA, através das entrevistas semi-estruturadas realizadas com instrutores e alunos no Instituto de Controle do Espaço Aéreo (ICEA), em setembro de 2007, foram apuradas as dificuldades em se construir cognitivamente um modelo do momento em que os conceitos abstratos de Meteorologia influenciam significativamente nos procedimentos operacionais.

Na elaboração da proposta de atividades contamos com a colaboração de um controlador de tráfego aéreo com uma gama de conhecimentos, habilidades, convicções e conceitos adquiridos ao longo dos 20 anos de atividade profissional. Sua contribuição faz com que o *workflow* (Figura

14) de produção de um objeto de aprendizagem dirigido a treinamentos para atuar em Sistemas Complexos pareça diferente de um OA ‘simples’, ao inserir a participação de *expertises* nos processos de identificação de cenários e produção de recursos educacionais.

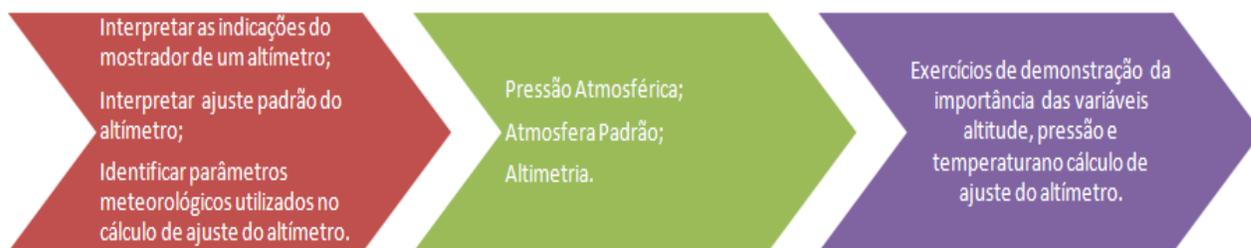


Figura 14 - Proposta de atividade do aplicativo

Cabe ressaltar que um mesmo conteúdo pode admitir mais que uma atividade de aprendizagem assim como uma mesma atividade de aprendizagem ser utilizada para alcançar diferentes objetivos previstos pelos grupos disciplinares.

A identificação do cenário (contexto) foi obtida por meio de entrevistas semi-estruturadas apresentadas no capítulo 5 deste trabalho, em que foram identificados os momentos em que os conceitos relacionados à pressão atmosférica, atmosfera padrão e altimetria influenciam significativamente nos procedimentos operacionais. Desse modo, a idéia principal do objeto de aprendizagem é possibilitar a revisão dos conceitos relacionados à pressão atmosférica, atmosfera padrão e altimetria por meio de exercícios práticos que estimulem sua compreensão e fixação.

6.3. Modelo de Projeto de Interface

A partir da elaboração do modelo de projeto instrucional configurado através de cenários e baseado no comportamento, esses requisitos são materializados através do modelo de projeto de interface aplicado ao desenvolvimento do OA.

O papel da interface é, portanto, o de mediador e orientador entre o aluno e o conhecimento, facilitando a aprendizagem dos conceitos veiculados na interface não se tratando apenas de aprender a fazer algo, mas de aprender o conceito por trás ou que o levou a uma ação dentro do sistema. Por esse motivo, seu processo de design de interface (design gráfico e de interação) distingue-se em função dos usuários finais ou alunos para os quais foram projetados. Sob o ponto de vista de IHC e ES, os operadores de sistemas complexos, são pessoas que

conhecem o domínio da tarefa em execução e usam o computador apenas para ajudá-los a completá-la de maneira fácil e eficiente, enquanto os alunos são pessoas que estão sendo treinadas para adquirir habilidades em domínios novos e desconhecidos. A aprendizagem de conceitos através do uso de uma interface educativa está relacionada à construção e simulação de objetos, que são representantes de conceitos abstratos. Tal representação é construída com o propósito de proporcionar ao aluno investigar, levantar hipóteses e testá-las, para que, dessa forma, ele seja conduzido a revisar suas idéias iniciais e construir o seu conhecimento com autonomia. Numa interface educativa, portanto, o erro é uma oportunidade para a reconstrução do conhecimento.

Os conceitos, em uma interface educativa, são representados metaforicamente e submetidos às ações dos usuários por meio de do design de interação, que influencia no potencial da interface para a aprendizagem de conceitos. Em geral, em tais interfaces, o aluno se vê diante de situações, que podem envolver a resolução de problemas, influenciada pelo estilo de interação adotado, o qual deve de algum modo, requerer o uso de recursos mentais. Desta forma, o estilo de interação do usuário com a interface, deve ser pensado e elaborado com o objetivo de criar situações promotoras de sentido, fazendo o aluno refletir sobre os conceitos veiculados e sobre a tarefa em execução.

6.3.1. Estilos de Interação

O Estilo de Interação é um termo genérico que inclui todas as formas como os usuários se comunicam ou interagem com sistemas computacionais. Sendo assim, a interação é um processo que engloba as ações do usuário sobre a interface.

O design de interação planejado para apoiar a aprendizagem deve ser elaborado aplicando restrições às ações dos usuários e permitir a correção dos erros através do input de ações corretivas por parte do usuário na realização de uma atividade ou tarefa. Por meio de tais restrições e correções de erro, espera-se que o usuário reflita sobre a tarefa em execução.

Todavia, são as ações realizadas pelo usuário na interface, que favorecem o acompanhamento da sua evolução nos conceitos veiculados pela mesma. Por esse motivo, nesta pesquisa foram avaliados os resultados encontrados na literatura sobre o estilo de interação MD - Manipulação Direta - no tocante ao esforço cognitivo conseqüente do seu uso.

Manipulação Direta

O termo Manipulação Direta (Norman, 1985), é utilizado para referenciar sistemas que representam na interface os objetos utilizados no momento. As ações na tela são realizadas através de botões ou de manipulação do mouse, utilizando operações reversíveis, incrementais e rápidas cujo impacto no objeto utilizado seja imediatamente visível. Os benefícios da manipulação direta residem nas seguintes características: os usuários novatos podem aprender funcionalidades básicas rapidamente, geralmente mediante uma demonstração; os usuários experientes se tornam mais eficientes para uma grande quantidade de tarefas; as mensagens de erro são utilizadas raramente; os usuários podem ver imediatamente se suas ações os conduziram para seus objetivos; os usuários reduzem o nível de ansiedade porque o sistema é compreensível e porque as ações são facilmente reversíveis.

Embora a manipulação direta apresente-se como um estilo de interação aceito pelo usuário, resultam do fato de solicitar pouco esforço cognitivo, e, dessa forma, exige o mínimo de aprendizagem necessária. Complementando, Sedighian (1997) indica que uma alternativa é mudar o foco, ou seja, colocar o usuário para manipular diretamente os conceitos do domínio, possibilitando a interação e a formulação de pensamentos voltados mais para os conceitos que estão sendo aprendidos do que para os seus objetos representativos. Dessa forma, o mesmo autor diz que o foco deve ser deslocado da manipulação direta de objetos para a manipulação direta de conceitos.

Manipulação Direta de Objetos - MDO: reduz a distância semântica entre os objetos da interface e o modelo mental do usuário, ou seja, saber como ele pensa e lhe oferecer algo parecido. Isso exige pouco esforço cognitivo e possibilita uma adaptação mais rápida à interface.

Manipulação Direta de Conceitos - MDC: Nessa interação, o usuário é convidado a adaptar-se gradualmente à semântica dos conceitos, ou seja, fazer o indivíduo mudar, aos poucos, seu modelo mental e, conseqüentemente, aprender. Em outras palavras, o usuário é conduzido a pensar sobre os conceitos que estão, metaforicamente, representados pelos artefatos da interface.

Manipulação Direta Reflexiva de Conceitos - MRDC: é a MDC - *Direct Concept Manipulation* - estendida com *scaffolding* ou suporte interacional. Nessa forma de manipulação, Sedighian 2001, Reiser 2002, Quintana 2002 e outros constataram que nesse modelo, alunos são submetidos a um esforço cognitivo maior do que quando utilizados o MDO e MDC. Segundo os

mesmos autores, esse estilo de interação é proveniente da extensão da MDC por meio do uso de um *feedback*, o *scaffolding*, que é oferecido durante a interação.

6.3.2. Tipos de Scaffolding

A partir das considerações finais do Capítulo 3 sobre as principais linhas teóricas da pedagogia foi possível identificar formas de mediação que configuram o processo de aprendizagem e que juntamente a análise comparativa entre IHC e ES apresentada no Capítulo 4 possibilitou desenvolver funcionalidades para atuarem como recursos de mediação pedagógica e assim elaborar dentro da proposta deste trabalho, o modelo de projeto de interface baseado em *scaffoldings* apresentado a seguir:

Compartilhamento dos objetivos específicos: Um ponto central do *scaffolding* bem sucedido é a noção de uma compreensão compartilhada do objetivo da atividade. Embora alguns elementos da atividade estejam além da compreensão do aluno, a intersubjetividade ou compreensão compartilhada da tarefa ajuda a fornecer a motivação para concluí-la. Dessa forma, é possível perceber que o compartilhamento de objetivos específicos possibilita a manutenção do interesse do aluno.

Engajamento / Manutenção do interesse do aluno: Consiste em atrair a atenção do aluno para a realização da tarefa. Uma forma eficaz de aumentar o envolvimento dos alunos em uma tarefa é tornando a mesma atraente, de forma a aumentar a motivação dos alunos.

Redução do grau de liberdade: Tem como objetivo permitir ao aluno desenvolver suas habilidades nos componentes da tarefa que ele pode controlar. A redução do grau de liberdade é obtida através da redução dos passos necessários para a realização da tarefa, restringindo as ações e o fornecimento de *feedback* por parte do tutor.

Manutenção do sentido / direção: Tem como objetivo impedir que o aluno se desvie do objetivo ou do caminho necessário para o cumprimento da tarefa. Durante o processo de realização da atividade, é função do tutor manter a motivação do aluno voltada para aquilo que ele está realizando, para que o mesmo possa aprender os conceitos envolvidos na resolução da tarefa.

Simplificação da tarefa / Destaque de aspectos críticos: Consiste no destaque a ser dado pelo professor, durante o ensino de um conteúdo, às partes que ele julga relevantes para o processo de aprendizagem bem como para a resolução de problemas. O uso dessas partes relevantes pelo aluno pode ser posteriormente utilizado pelo professor como critério para medir o progresso do aluno na aprendizagem e no domínio do conhecimento e habilidade necessários para o cumprimento da tarefa.

Controle da frustração: A realização de uma tarefa pode tornar-se uma experiência estressante quando o mesmo possui dúvidas sobre como resolvê-la. É função do professor, procurar diminuir a frustração do aluno, aumentando sua motivação para o aprendizado.

Demonstração / Fornecimento de trajetos ideais da solução: Consiste em desenvolver modelos de soluções para a tarefa ou para partes dela, de forma a facilitar a percepção, por parte do aluno, dos conceitos e procedimentos relevantes na realização da atividade.

Foco na resolução da tarefa completa: Embora a tarefa possa ser dividida em sub tarefas, a literatura sobre *scaffolding* conclui que é melhor focar no problema todo do que em suas subcategorias. Esse aspecto do *scaffolding* significa que os alunos manterão sua atenção em adquirir habilidades para a tarefa completa, que constitui o objetivo para o qual estão sendo treinados.

A disponibilidade imediata da ajuda: A idéia desse elemento é fornecer aos alunos uma ajuda imediata. Se não lhes for fornecido esse auxílio imediato, eles podem tornar-se frustrados com a tarefa e, assim, não conseguir seu objetivo.

Ajudar quando houver necessidade: similar à última idéia, esse elemento do *scaffolding* ideal incentiva o instrutor a ajudar, imediatamente, o aluno, tomando por base se essa foi ou não necessária. Caso a ajuda seja fornecida em outro momento, haverá uma interrupção do pensamento do aluno.

Fornecimento de uma ajuda efetiva: A ajuda efetiva envolve disponibilidade imediata, porém apenas quando for isso for necessário.

Conhecimento do modelo do tutor: Um modelo do tutor dá aos alunos informações acerca das habilidades que precisam adquirir. Tendo uma representação ideal da habilidade que estão tentando adquirir, os alunos terão uma idéia do objetivo para os quais estão sendo treinados.

Divisão de uma tarefa em suas sub tarefas: O propósito da divisão de tarefas em sub tarefas é o de facilitar o entendimento da tarefa maior por meio da sua subdivisão em tarefas menores. Assim é o possível evidenciar ao aluno que a execução de uma tarefa é realizada por um conjunto de etapas.

Estruturação e Problematização: Segundo Reiser (2002), o *scaffolding* torna-se necessário frente a uma tarefa, que possui uma maior complexidade, de natureza aberta ou não estruturada. A problematização surge como uma forma de impedir que os passos da tarefa sejam negligenciados, e a estruturação, é utilizada como uma forma de organizar o pensamento do aluno, fazendo-o refletir. Por exemplo, se os alunos forem forçados a usar um menu para categorizar os dados coletados, eles terão que considerar o significado dos mesmos, sendo obrigados a refletir sobre tais categorias.

Sob a ótica das teorias de Ausubel, Flavell, Seminério e Bruner, as metáforas, também podem ser consideradas como *scaffoldings*, pois funcionam como elementos de sustentação para o aprendizado ou subsunçores, possibilitando que conceitos previamente aprendidos tornem-se significativos e sejam aplicáveis a novas situações. Assim as metáforas em interfaces ampliaram o foco de sua finalidade e seu objetivo inicial de facilitadora da navegação pelo ambiente virtual, para o de facilitadora do processo de aprendizagem.

A utilização de metáforas na interface tem como objetivo prover o usuário com um modelo convencional e já conhecido do sistema com o qual irá interagir. Erickson (1990) propõe que a utilização de metáforas em interfaces deve seguir um processo baseado nas seguintes etapas:

- 1) Estudar a funcionalidade e representatividade do sistema a ser criado;
- 2) Apoiar a definição e utilização de metáforas a partir da identificação das partes mais difíceis de representação do conteúdo e compreensão do aluno.
- 3) Utilizar metáforas já implícitas na descrição da atividade ou do problema procurando eventos reais ou elementos representativos de algumas das características que os alunos tenham dificuldade de entender. Portanto a metáfora escolhida deve ser representativa de funcionalidades do sistema, ser baseada nos aspectos da estrutura da tarefa, sua aplicabilidade e contexto de utilização, poder de representação conceitual, adequada ao público alvo e ter a possibilidade de extensão.

Em relação à estrutura, o objetivo é verificar quanto da estrutura de apoio a aprendizagem, a metáfora ajuda ao usuário pensar no sistema. Em relação à aplicabilidade deve ser verificado se a metáfora é relevante ao problema; metáforas que podem conduzir o usuário na direção errada ou levantar falsas expectativas devem ser evitadas.

Quanto à representação, metáforas ideais têm representações visuais distintas e palavras específicas associadas. Em relação à adequação ao usuário, deve-se verificar se o público-alvo entende a metáfora; ela seria inútil em caso negativo. Portanto, ainda com base nas considerações finais do Capítulo 3 sobre as principais linhas teóricas da pedagogia juntamente a estudos de casos realizados e coletados da literatura principalmente em IHC, o modelo de projeto de interface deve seguir três diferentes etapas para o design baseado em metáforas:

Fase 1 - Identificação e elaboração de metáforas:

- Analisar como os usuários entendem e interagem com sistemas computacionais;
- Construir metáforas a partir de referências já existentes;
- Utilizar artefatos predecessores como metáforas;
- Identificar metáforas implícitas na atividade ou representativas de um problema descrito;
- Identificar eventos reais que representem e forneçam aspectos chave para a compreensão.

Fase 2 - Seleção de metáforas candidatas ao design:

- Selecionar metáforas de estrutura rica e aplicável;
- Selecionar metáforas adequadas ao perfil do usuário, com significado literal de fácil compreensão;
- Selecionar metáforas conceitualmente distantes entre a fonte e o significado metafórico do conceito representado;
- Selecionar metáforas que possuam conceitos que sirvam de “ponte” entre o significado literal e o conceito metafóricamente;
- Não necessariamente incorporar a metáfora ao design final.

Fase 3 - Desenvolvimento e aplicação no sistema propriamente:

- Definir o conceito principal;
- Buscar novos significados para cada conceito;
- Reestruturar a percepção da realidade;
- Buscar suposições para identificar conceitos que estão explícitos e implícitos na metáfora;
- Estudar a estória da metáfora, buscando identificar como a conceituação foi constituída e como ela pode ser reaproveitada;
- Identificar as partes não usadas da metáfora;
- Gerar situações de conflitos.

A criação de um software educativo de boa qualidade requer a observação de detalhes em cada etapa de desenvolvimento. Segundo Gomes (2005), trata-se da criação de um sistema dinâmico com especificações e requisitos específicos quanto à interação com o usuário e seu impacto sobre a aprendizagem. Tendo em vista o exposto, apenas a análise da tarefa é insuficiente, pois nesse contexto não apenas a tarefa é relevante, mas também os aspectos cognitivos dos usuários.

A utilização de *scaffolding* em um software educacional requer a compreensão do papel a ser desempenhado pelo software enquanto parceiro do aluno no processo de aprendizado. Neste sentido, as categorias de *scaffolding* vistas na seção anterior destacam questões importantes a serem atendidas no projeto desse recurso, o qual tem como objetivo final oferecer ajuda adequada às necessidades dos usuários.

Uma forma de adequar o *scaffold* utilizado em um software educacional às necessidades dos alunos é investigar o processo de execução da tarefa realizado pelos alunos. A partir da observação dos passos, procedimentos, observações e dificuldades apresentados pelos alunos durante a execução da tarefa é possível formular uma ajuda adequada ao aprendizado do conteúdo em questão. Outras fontes de inspiração a serem consideradas no projeto do *scaffolding* podem ser encontradas no próprio conteúdo a ser veiculado na interface, através da identificação

dos conceitos e invariantes necessários à aprendizagem do assunto bem como na análise de programas concorrentes.

A partir das diretrizes e recursos apresentados como proposta neste capítulo é possível apontar os seguintes passos para a confecção de uma interface educacional baseada no uso de *scaffolding* e estruturar o modelo apresentado a seguir (Figura 15):

Estruturar Tarefas e Funcionalidades:

- Definir a atividade de aprendizagem
- Definir o cenário operacional e os conceitos a serem veiculados na interface;
- Identificar os conceitos necessários ao aprendizado deste cenário;
- Identificar as situações que tornam esses conceitos significativos e oferecer essas situações ao usuário
- Definir as diferentes representações que serão utilizadas para lidar com essas situações e permitir ao usuário fazer uso delas durante o uso da interface;
- Definir a conexão entre algumas das representações utilizadas e de que forma o usuário poderá utilizar estas conexões para passar informações de uma representação para outra;
- Identificar os invariantes que devem ser capturados pelos usuários ao lidar com essas situações;
- Projetar o *scaffolding* (ajuda para o usuário) a ser utilizado na interface de forma a facilitar a percepção desses invariantes bem como auxiliar o usuário no processo de resolução de problemas com a interface;
- Restringir uma tarefa complexa por definição de “limites de coerência” para os alunos
- Viabilizar a visualização tarefas complexas usando a decomposição ordenada e desordenada das tarefas
- Restringir o espaço de atividades usando modos funcionais;
- Desenvolver um protótipo para realização de testes com usuários;

- Avaliar os invariantes mobilizados pelo usuário ao utilizar o protótipo bem como a eficácia do *scaffolding* em auxiliar o aluno durante o uso da interface educativa;
- Caso necessário, realizar modificações no protótipo e testar novamente com o usuário. Repetir o ciclo (modificação do protótipo/teste/avaliação) até que o software seja considerado adequado para a aprendizagem dos conceitos veiculados pelo mesmo.

Fornecer acesso ao conhecimento do especialista:

- Fornecer conteúdo de orientação para ajudar os aluno;
- Fornecer informações para o uso da ferramenta;
- Fornecer informações metacognitivas.

Automatizar de forma implícita as tarefas de rotina

- Facilitar a organização dos elementos de interação;
- Facilitar a circulação entre as ferramentas e as atividades;
- Automatizar partes de tarefas para reduzir demandas cognitivas acima do esperado.

Usar representações que podem ser identificadas pelos alunos

- Fornecer representações que podem ser identificadas pelos alunos e revelar conceitos subjacentes;
- Dar aos alunos a capacidade de ligar diferentes pontos de vista sobre o mesmo conteúdo;
- Dar aos alunos "pontos de vista mutáveis" que lhes permitam manipular diretamente (ou alterar) as representações;

Usar representações e linguagem como ponte para a aprendizagem

- Fornecer organizadores visuais conceituais para dar acesso a uma funcionalidade;
- Usar terminologia que forneça aos alunos conceitos intuitivos para descrever conceitos complexos.

Facilitar a articulação de descrições e explicações

- Destacar as características epistêmicas das explicações para apoiar o desenvolvimento de explicações científicas;
- Destacar as características epistêmicas das descrições.

Facilitar a articulação das ações e do progresso da tarefa

- Dar aos alunos um espaço para facilitar o planejamento de aspectos do resultado de sua ação;
- Fornecer informações e orientações para facilitar o monitoramento.

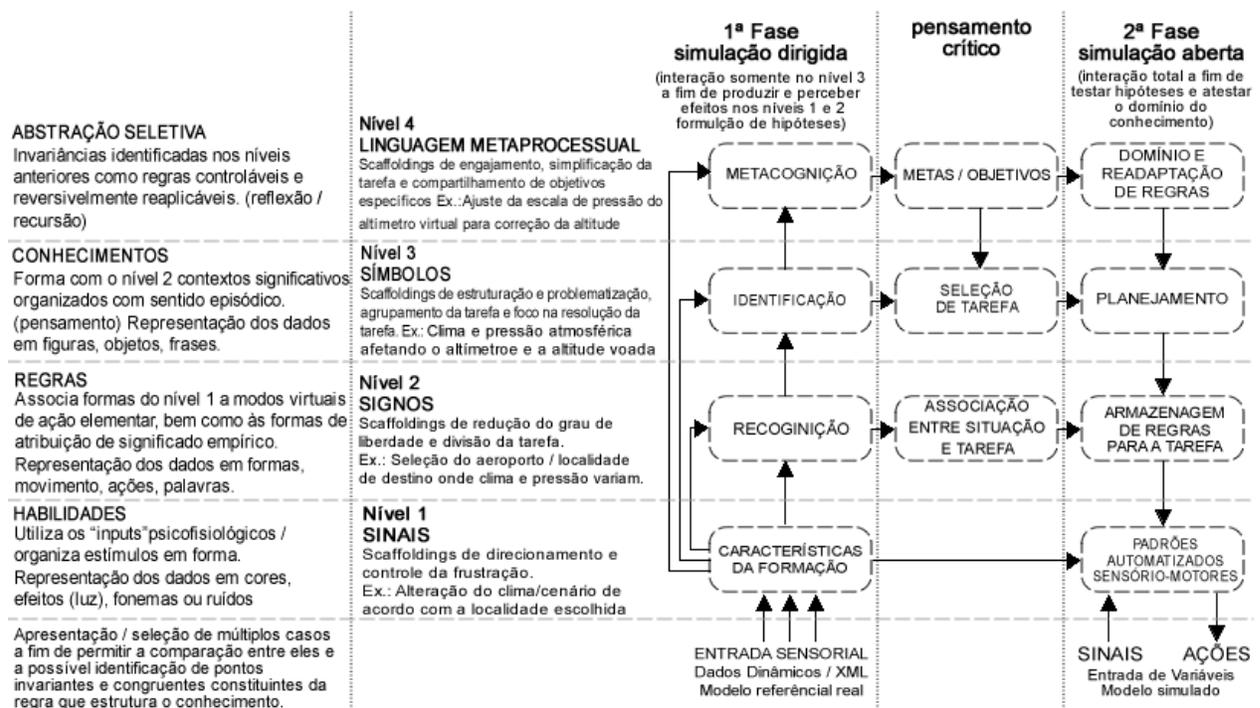


Figura 15 - Modelo de objeto de aprendizagem apoiado em scaffoldings (Lapoli 2009, 2010 e)

É possível afirmar, que a estruturação de um recurso educacional baseado no modelo integrado de desenvolvimento de IHC e ES, estimula o aluno a analisar problemas, desenvolvendo a capacidade de investigação, formando uma base de conhecimento através de um processo progressivo.

A estruturação do conhecimento sobre altimetria (Figura 16) parte primeiramente da percepção da temperatura, que se traduz na forma da condição climática a partir da escolha do

aeroporto contextualizada no cenário do OA. Nesta fase de aprendizagem o aluno é estimulado a descobrir qual a relação entre temperatura local e a altitude da aeronave.

O conhecimento sobre pressão atmosférica está relacionado à temperatura através da escala barométrica sendo a escala mais baixa referente à temperatura mais baixa (e vice-versa). Através desta escala, o aluno é orientado a fazer o ajuste do altímetro configurando a aprendizagem superordenada. Esse ajuste também é feito a partir dos dados numéricos capturados em tempo real diretamente da Rede de Meteorologia do Comando da Aeronáutica (REDEMET). Neste momento, caso ocorra a variabilidade da temperatura e da pressão atmosférica de acordo com a localidade escolhida há a necessidade de ajuste do altímetro e o aluno pode perceber a correlação entre essas variáveis.

Assim como as condições climáticas apóiam a aprendizagem combinatória de temperatura, entende-se que a construção da aprendizagem subordinada referente à pressão atmosférica deveria estar apoiada nos elementos gráficos do altímetro e da aeronave embora não estejam presentes no procedimento operacional dos controladores por considerá-los fundamentais para compreender a realidade que irão encontrar em sua prática profissional.

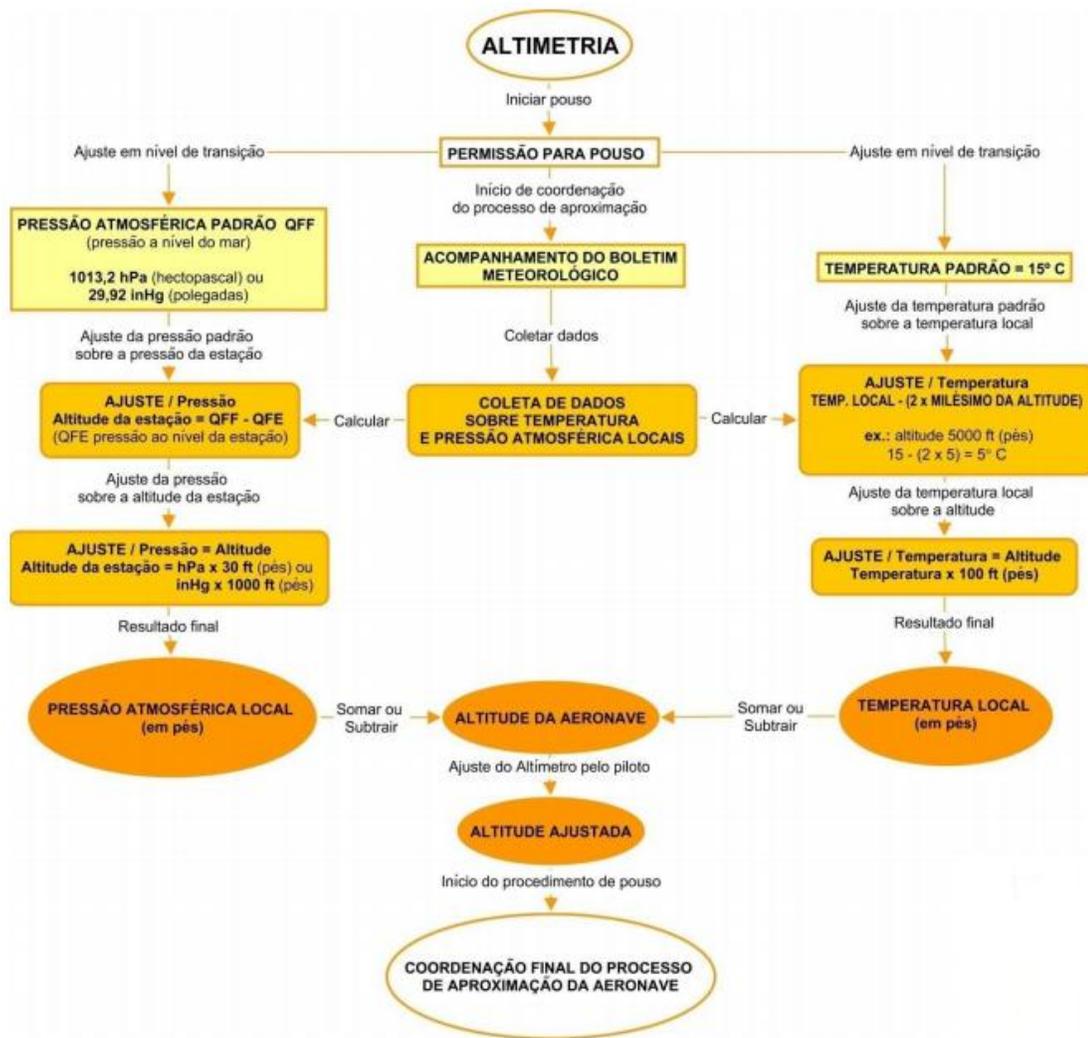


Figura 16 - Estruturação dos conhecimentos envolvidos na altimetria (Lapoli 2009)

Nessa perspectiva, ao final do processo, o aluno poderá identificar a correlação entre as variáveis representadas no OA pela sua aplicação e/ou experimentação de situações análogas à realidade. Assim, a aprendizagem mediada pelo protótipo elaborado a partir da estrutura apresentada (Figura 16) poderia contribuir na construção do percurso cognitivo do aluno, optando pelos segmentos condições reais ou condições simuladas.

As tecnologias aplicadas para o desenvolvimento do protótipo envolveram o aplicativo Flash junto com a linguagem de programação Action Script, integrada à busca e captura de dados em linguagem XML diretamente da Rede de Meteorologia do Comando da Aeronáutica (REDMET), referentes às condições meteorológicas do aeroporto escolhido.

A escolha do aplicativo Flash está fortemente embasada no fato dele oferecer amplos recursos de animação, disponibilizando arquivo de menor tamanho em relação a outros, e de ser de fácil aprendizagem, aspecto importante se houver necessidade de modificações futuras nas apresentações.

Tendo como base a definição de cenários da etapa anterior, foram realizados testes para uma pré-avaliação no que diz respeito à usabilidade, acessibilidade, bem como outras questões técnicas (programação e funcionamento) e pedagógicas. Conforme a estrutura foi sendo finalizada, iniciou-se o acabamento gráfico das interfaces e, após as últimas alterações, chegou-se ao protótipo do ClearaNCE (Figura 17).



Figura 17 - Tela principal do ClearaNCE

A aplicação pedagógica do ClearaNCE é trabalhar, mais especificamente a relação das variáveis altitude, pressão e temperatura utilizadas no cálculo de ajuste na altimetria. Assim, o que antes era feito apenas de modo teórico é representado no objeto de aprendizagem, em que o aluno tem a possibilidade de observar a correlação das variáveis, a influência de umas sobre as outras e a influência delas sobre o posicionamento vertical da aeronave.

O ClearaNCE permite a escolha das condições iniciais, isto é, é possível optar trabalhar com dados reais obtidos da Rede de Meteorologia do Comando da Aeronáutica (REDMET) ou com dados previamente definidos pelo instrutor (simuladas).

Dada as condições iniciais, a cada utilização do objeto de aprendizagem alguns fatores como chuva, nebulosidade e vento poderão ser alterados, enfatizando que as regras do ar e serviços de tráfego aéreo são sensíveis a diferentes variáveis.

Com os dados oriundos da REDEMET, a partir da escolha do aeroporto e os valores de pressão e temperatura local o aluno visualiza a diferença de altitude em que o piloto estaria se orientando, promovendo a percepção da importância destes conceitos na aplicação prática de sua atividade profissional. Já com a entrada de dados simulada, o aluno tem a oportunidade de colocar em prática o ajuste de altitude trabalhando na determinação das variáveis, pois para fazer com que a aeronave chegue a uma altitude exata, ele será obrigado a calcular a incidência da pressão e da temperatura em pés sobre a altitude padrão que ele colocou.

Ainda é possível ver a estrutura interna indicando as peças que compõem o altímetro através de um infográfico (Figura 18).



Figura 18 - A estrutura interna do Altímetro

O ClearaNCE leva em consideração algumas premissas que são refletidas em sua apresentação e representação da realidade:

- a) Os objetivos de aprendizagem a serem alcançados a partir dele foram fundamentais para o planejamento do conteúdo e das atividades, isto porque é necessário ter em mente que objetos de aprendizagem são um recurso a mais e não a solução de todas as dificuldades relacionadas ao processo de ensino e aprendizagem.
- b) A contextualização permite aos alunos traçar mais facilmente uma relação entre determinado conteúdo e suas aplicações práticas, levando a uma aprendizagem mais significativa.
- c) A simulação de eventos ou processos utilizados como exemplos nas aulas teóricas são bastante apropriados para o uso dessa mídia. Desse modo, o objeto de aprendizagem deve cobrir o procedimento de modo a permitir a construção de um comportamento e/ou habilidade comum a todos os profissionais.

Eventualmente, o aluno poderá ser direcionado a outros objetos de aprendizagem para suprir a carência em conceitos relacionados ao ensino médio ou fundamental, considerados pré-requisitos para melhor compreensão sobre pressão atmosférica, atmosfera padrão e altimetria. Isto porque no estudo de campo descrito no Capítulo 4, tanto os instrutores como alunos reconheceram que algumas dificuldades são oriundas de carências nas disciplinas Matemática e Física.

6.4. Implementação

Para atender as demandas variacionais de modelagem das funcionalidades do aplicativo de forma mais eficaz de acordo com o modelo de desenvolvimento proposto, foi utilizada a metodologia ágil de *Extreme Programming* (XP). Essa metodologia ágil permite desenvolver aplicativos com requisitos vagos, imprecisos ou em constante mudança, através da estratégia de constante acompanhamento e realização de vários pequenos ajustes ao longo do desenvolvimento. Esta abordagem segundo Boyle (2006), deve ser constituída de:

- Prototipagem rápida e iterativa;
- Uso de pequenas equipes ágeis;

- Usuário na equipe de projeto;
- Ênfase em produtos em vez de seguir processos definidos;
- Prazos apertados, embora controlados por métodos de gestão do tempo.

Os quatro valores fundamentais da metodologia XP são: comunicação, simplicidade, *feedback* e coragem. A partir desses valores, possui como princípios básicos: *feedback* rápido, presumir simplicidade, mudanças incrementais, flexibilidade frente a mudanças e trabalho de qualidade. Dentre as variáveis de controle em projetos (custo, tempo, qualidade e escopo), há um foco explícito em escopo. Para alcançar controle, recomenda-se a priorização de funcionalidades que representem maior valor possível para o negócio. Desta forma, caso seja necessário a diminuição de escopo, as funcionalidades menos valiosas serão adiadas ou canceladas. O desenvolvimento do OA foi apoiado, em um conjunto de boas práticas em XP de acordo com Beck (2000), adaptado e aplicado como descrito em Lapolli (2009 b, 2010 a, b, e f):

Jogo de Planejamento (*Planning Game*): Nesse modelo, há uma interação muito próxima entre cliente e desenvolvedor, cabendo aos programadores estimarem o esforço necessário para a execução de histórias de clientes e ao cliente a decidir sobre o escopo e o prazo desenvolvimento das versões para acompanhamento do andamento do projeto. No desenvolvimento dos objetos de aprendizagem, os clientes foram os instrutores, podendo ser também, especialistas, professores ou os designers instrucionais que são pessoas responsáveis por determinarem essas funcionalidades e prioridades a partir do modelo pedagógico e de sua experiência e prática em sala de aula.

Metáfora (*Metaphor*): Todo o sistema é definido por um conjunto de metáforas entre o cliente e os programadores. Esse sistema de códigos de linguagem é compartilhado pela equipe de desenvolvimento, servindo de referência na descrição sobre como o sistema funciona. O estabelecimento desse código visa facilitar a comunicação com o cliente traduzindo esse conjunto de funcionalidades em uma linguagem próxima a realidade dele. Nesse caso, o sistema de códigos foi construído tendo como referencial à realidade do aluno, identificada pelo professor ou pelo designer instrucional.

Equipe Coesa (*Whole Team*): No projeto recomenda-se a composição de equipes não muito numerosas com um representante de cada área de desenvolvimento. Fizeram parte da equipe, o

cliente, o programador, o designer e o designer instrucional que estiveram em sintonia com professor onde a coesão permitiu uma melhor integração das partes de desenvolvimento.

Projeto Simples (*Simple Design*): A ênfase está em projetar a solução mais simples possível e exequível neste momento. Complexidade desnecessária e códigos extras são removidos imediatamente, buscando implementar as funcionalidades essenciais para o cumprimento da proposta pedagógica.

Programação em Pares (*Pair Programming*): A programação é realizada em par/dupla num único computador com o aplicativo sendo revisto por duas pessoas, evitando e diminuindo assim a possibilidade de erros. Esse mesmo modelo foi adotado em outras partes de desenvolvimento do aplicativo, como no design.

Padrões de Codificação (*Coding Standards*): São estabelecidos códigos padronizados pela equipe de desenvolvimento a fim de integrar e facilitar o processo de construção do aplicativo. No caso do objeto de aprendizagem, a codificação foi pautada na linguagem técnica da área.

Posse Coletiva (*Collective Ownership*): O código-fonte é compartilhado entre os desenvolvedores podendo ser modificado ao mesmo tempo e construído de forma colaborativa. O objetivo com isto é fazer a equipe conhecer todas as partes do sistema podendo ser feita a atualização do desenvolvimento através de repositórios de versões compartilhados pela equipe.

Refatoração (*Refactoring*): Definição e redefinição da arquitetura a todo o momento em um aperfeiçoamento contínuo do projeto, após a consolidação de uma funcionalidade, com o mínimo de introdução de erros e mantendo a compatibilidade com o código já existente. No caso da programação, refatorar melhora a clareza (leitura) do código, dividindo-o em módulos mais coesos e de maior reaproveitamento, evitando a duplicação de código-fonte. Nas áreas de programação e design, a refatoração foi constante, de maneira integrada, com o aperfeiçoamento do código e do designer de forma progressiva, simultânea e complementar.

Reuniões em Pé (*Stand-up Meeting*): Realização de reuniões de curta duração entre os membros da equipe para discutir tarefas realizadas e tarefas a serem realizadas pela equipe. Esse formato é utilizado com o objetivo de conseguir maior concentração dos membros durante a reunião. Esse modelo foi adaptado em alguns momentos, com o uso de ferramentas de comunicação síncrona, no caso o Skype, devido à impossibilidade de deslocamento do cliente.

Integração Contínua (*Continuous Integration*): A integração de novas funcionalidades é realizada de maneira contínua e imediata evitando a possibilidade de conflitos e erros no código fonte, permitindo acompanhar o status real da programação pelo cliente, facilitando a identificação de ajustes a serem realizados nas funcionalidades do aplicativo. No desenvolvimento do OA, houve a integração contínua de novas funcionalidades não só na programação, como também no design, a fim atender a proposta pedagógica.

Pequenas Versões (*Small Releases*): O desenvolvimento modulado do aplicativo permitiu a liberação de pequenas versões com partes essenciais para a utilização e testagem simultaneamente a seqüência de desenvolvimento e aperfeiçoamento do aplicativo. Esse processo permitiu o acompanhamento pelo cliente, orientando os ajustes a serem realizados.

Testes de Aceitação (*Customer Tests*): Testes unitários executados antes do desenvolvimento final do código e executados de forma contínua ao longo do projeto, sendo os clientes responsáveis por especificar os testes funcionais através da relação de requisitos de funcionamento do objeto de aprendizagem.

6.5. Considerações Finais

O modelo apresentado foi baseado em conceitos de IHC e nos pressupostos das metodologias ágeis de Engenharia de Software, mais especificamente o desenvolvimento dirigido pelo comportamento esperado do sistema. Esse modelo é aplicável nas importantes fases do planejamento: projeção de finalidades (objetivos), formas de mediação (construção de recursos) e metodologia de desenvolvimento. A finalidade da aplicação dos modelos de captura da história do usuário é estimular a participação dos *experts* e futuros usuários do sistema durante o desenvolvimento do software. No âmbito educacional, o que se pretende com tal prática é aumentar o número de cenários que possam contribuir com a formação profissional, possibilitando ao aluno interiorizar conhecimento suficientemente necessário de tal forma que seja possível a sua abstração e aplicação em diferentes situações, inclusive, as não previstas durante o seu treinamento.

A utilização de cenários no desenvolvimento de software educacional é o meio de representar, analisar e planejar como um sistema computacional pode causar impacto nas

atividades e experiências do usuário. Um cenário é uma descrição em geral narrativa, baseado nas histórias dos usuários, mas também em outros formatos, sobre o que as pessoas pensam, fazem, ou tentam fazer durante o processo operacional de sistemas complexos.

O cenário identifica o usuário como tendo certas motivações para o uso do sistema, descreve as ações tomadas e razões para essas ações. Para o desenvolvimento de software educacional, ajuda a identificar elementos de apoio ou suporte a aprendizagem, os *scaffoldings* e a visualizar aspectos da atividade e experiência adquirida ou necessária do usuário. A elaboração de cenários permite assim definir funcionalidades e formas de interação que contribuam na mediação do processo de aprendizagem. Assim a metodologia proposta, utiliza a análise das tarefas cognitivas para elaborar cenários que podem ser analisados para identificar os objetos centrais do domínio do problema e articular o estado, comportamento e interação funcional dos objetos de interface na concepção de software educacional para o treinamento em sistemas complexos.

Por fim, a metodologia de desenvolvimento ágil *Extreme Programming* (XP) permite à equipe de treinamento e desenvolvimento motivar a participação das partes envolvidas e interessadas na formação de futuros profissionais, mostrando a importância das práticas de desenvolvimento ágil para a definição de objetivos e construção de recursos didáticos, podendo o mesmo ser expandido as outras etapas de modo que a preparação, a implementação, a avaliação, e a revisão da instrução sejam vistos como um processo integrado de estruturação das tarefas operacionais e cognitivas apoiada pela tecnologia.

Capítulo 7 - Avaliação da Metodologia de Desenvolvimento de OAs

Neste capítulo são descritas as avaliações da metodologia de desenvolvimento e do modelo de objeto de aprendizagem propostos. Para a avaliação da metodologia de desenvolvimento, foi realizada uma pesquisação. Na avaliação do modelo de objeto de aprendizagem, foi realizado um estudo de caso e uma pesquisa experimental. Ambas as pesquisas foram realizadas para verificar a viabilidade da solução implementada, que teve como objetivo avaliar a hipótese que orienta essa dissertação.

Nesta dissertação procurou-se mostrar que as metodologias ágeis de Engenharia de Software (ES) integradas a conceitos de Interface Computador (IHC) podem apoiar equipes de treinamento e desenvolvimento de OAs e sendo algo viável e útil, sob vários pontos de vista.

Identificar os objetivos e cenários de um treinamento para atuar em um sistema complexo, com características próprias, requer soluções específicas. Um cenário onde são levados em consideração os conceitos e habilidades necessários para determinada tarefa pode promover melhorias no desempenho da atividade profissional.

7.1. Avaliação da Metodologia de Desenvolvimento

Para testar a viabilidade do modelo de desenvolvimento integrado proposto, foi realizada uma pesquisa-ação com alunos do 3º ano do ensino médio de nível técnico em informática do Colégio Pedro II no Rio de Janeiro.

A amostra foi composta por 6 alunos, divididos em 3 duplas representativas de unidades e turmas diferentes. Todos os alunos possuíam conhecimentos básicos de programação em linguagens como Python e Java, mas não tinham conhecimento em *Action Script*, linguagem de programação utilizada no Flash e nem sobre Design de Interação. A escolha do aplicativo Flash está fortemente embasada no fato dele oferecer amplos recursos de animação, disponibilizando arquivo de menor tamanho em relação a outros, e de ser de fácil aprendizagem, aspecto importante para a necessidade de modificações futuras nos aplicativos. Outro fator para a escolha do Flash foi o fato de sua linguagem de programação, o *Action Script* também possuir a estrutura adequada para o desenvolvimento orientado a objeto. Essa característica que favorece o desenvolvimento de aplicativos modularizados e flexíveis a manutenção e reaproveitamento de seu conteúdo. Isso permite que a fase de prototipação se estenda até a validação plena das funcionalidades e posterior consolidação da versão final do aplicativo.

Os alunos receberam aulas e material de estudo em vídeo e texto sobre orientação a objetos e *Action Script*, onde foi demonstrada a sintaxe da linguagem e como programar, além disso, foi apresentado todo o funcionamento do programa Flash na versão CS4, assim como a linguagem de programação *Action Script* na versão 3. Sempre intercalando a apresentação do conteúdo com exercícios de fixação que permitissem a prática de programação.

Foram, então, demonstrados os conceitos básicos sobre Objetos de Aprendizagem, Design de Interface para que os alunos compreendessem o propósito das funcionalidades que seriam desenvolvidas nos projetos.

Também foi apresentado o modelo para esse processo de desenvolvimento com as etapas que deveriam ser seguidas, conduzindo ao cumprimento das metas de produção. Ao final do curso foi proposto um teste prático que tinha por objetivo por a prova os conceitos apresentados até então.

A partir dos principais alicerces teóricos apresentados nesta dissertação, procurou-se estruturar a elaboração de protótipos levando em consideração a integração existente entre as subunidades (temas de aula) que compõem a disciplina e os conhecimentos subordinados ao tema altimetria, objeto de estudo desta pesquisa.

Cada dupla escolheu e ficou responsável pelo desenvolvimento de um objeto de aprendizagem, a partir capítulos da apostila de treinamento dos controladores de tráfego aéreo. Os temas definidos foram:

A. A Terra no Espaço

B. Temperatura do Ar

C. Condições Meteorológicas Adversas ao Vôo.

Ao final do processo, era esperado que os alunos, sem o acompanhamento do professor orientador, desenvolvessem, sem que lhes fosse ensinado, protótipos de objetos de aprendizagem interativo, programado na linguagem *Action Script*.

As duplas foram orientadas pela proposta de metodologia integrada apresentada neste trabalho. Os trabalhos foram iniciados com o levantamento dos requisitos aplicando o método GOMS descrito nas considerações finais do Capítulo 2 sendo transformados posteriormente em *storycards* com a aplicação do BDD descrito no Capítulo 6 nas seções 6.1 e 6.2. *Os storycards* serviram como base para a identificação inicial dos conceitos iniciais e elaboração de *scaffoldings* a serem aplicados no desenvolvimento de cenários.

Assim como no ClearaNCE, primeiro protótipo desenvolvido a partir da metodologia proposta, as informações para a configuração dos cenários operacionais, seriam captados da Rede de Meteorologia da Aeronáutica (REDMET), através de dados em XML.

Para a construção dos cenários, cada funcionalidade do aplicativo era elaborada, a partir da descrição feita nos *storycards* sendo possível pensar e elaborar uma funcionalidade como parte essencial de uma atividade ou tarefa. Assim, a classificação e distribuição desses elementos componentes do cenário operacional, poderiam ser feitas seguindo uma hierarquia estabelecida na proposta como sendo encadeadora do raciocínio lógico sobre o conceito veiculado por esses artefatos. Assim, o processo de desenvolvimento foi dividido em 4 etapas referentes aos 4 níveis de *scaffoldings* apresentados no modelo de objeto de aprendizagem proposto na seção 6.3 deste trabalho.

A seqüência da metodologia integrada de desenvolvimento também foi estabelecida em 4 fases com o objetivo de deixar as funcionalidades e os *scaffoldings* mais específicos para serem finalizados em último estágio, estendendo a fase de prototipação ao máximo a fim de garantir a consolidação das funcionalidades principais.

A divisão do desenvolvimento em etapas teve o propósito de observar o desempenho das duplas ao longo do processo e identificar o momento de surgimento de possíveis dificuldades e entraves de desenvolvimento relacionados à especificidade das funcionalidades. É importante salientar que por esse motivo, não houve interferência do professor, no caso de surgimento de dúvidas de qualquer natureza.

O processo de desenvolvimento dos aplicativos seguiu para a fase de desenvolvimento de protótipos onde foram aplicadas as boas práticas de desenvolvimento ágil, componente importante do modelo integrado apresentado como parte da proposta. Nesse momento, foi possível realizar uma primeira avaliação do desempenho das duplas.

As duplas A, B e C apresentaram um desempenho satisfatório no cumprimento das etapas iniciais de desenvolvimento dentro do modelo proposto, porém, apresentaram dificuldades na definição das funcionalidades a serem desenvolvidas nos aplicativos a partir do nível 3. A dupla A, obteve um desempenho melhor em relação às outras, porém também apresentou dificuldades para a seqüência do desenvolvimento.

Foi notado que as dificuldades começaram a surgir na medida em que os níveis propostos de *scaffoldings* exigiam um conhecimento maior sobre as áreas específicas de cada capítulo.

A explicação sobre o desempenho diferenciado da dupla A é possível e foi relatado pelos próprios alunos indicando que basearam a escolha, no fato do conteúdo trabalhado fazer parte do currículo acadêmico de alunos do ensino médio, envolvendo conhecimentos básicos de física. Já as

duplas B e C justificaram a escolha pelos respectivos capítulos por acreditarem na maior correlação dos temas com a atividade profissional dos controladores de tráfego aéreo.

Portanto, para concluir a análise da viabilidade do desenvolvimento desse tipo de aplicativo, as duplas foram assistidas por um especialista em IHC com conhecimento de metodologias ágeis de ES, no caso o professor que forneceu as orientações necessárias sobre quais e como as funcionalidades deveriam ser desenvolvidas pelas duplas através de protótipo. Para reduzir também o impacto no experimento do tipo de conteúdo a ser trabalhado, optou-se por concentrar o trabalho das duplas no desenvolvimento do tema que havia sido escolhido pela dupla A “A Terra no Espaço” e que faz parte do currículo acadêmico de alunos do ensino médio, envolvendo conhecimentos básicos de física e que os alunos possuem maior familiaridade.

A partir desses ajustes no experimento, o desenvolvimento do aplicativo não só foi viabilizado como possibilitou o aperfeiçoamento da qualidade final do aplicativo (Figuras 19, 20, 21 e 22).

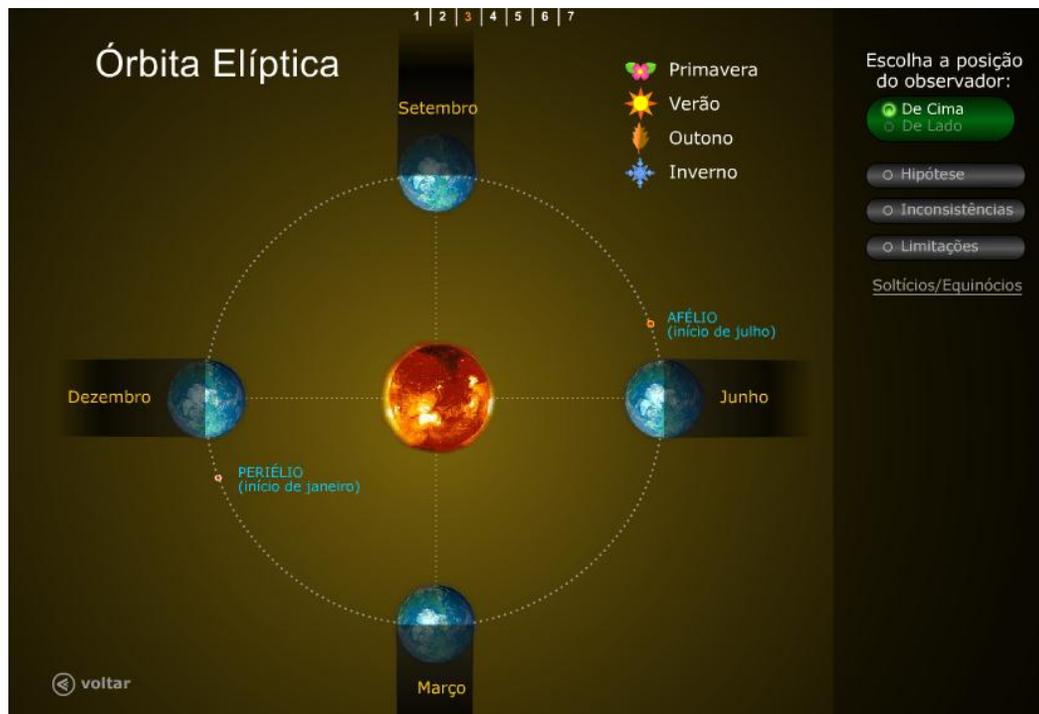


Figura 19 - Tela do aplicativo “A Terra no Espaço” com o subtema Orbita Elíptica

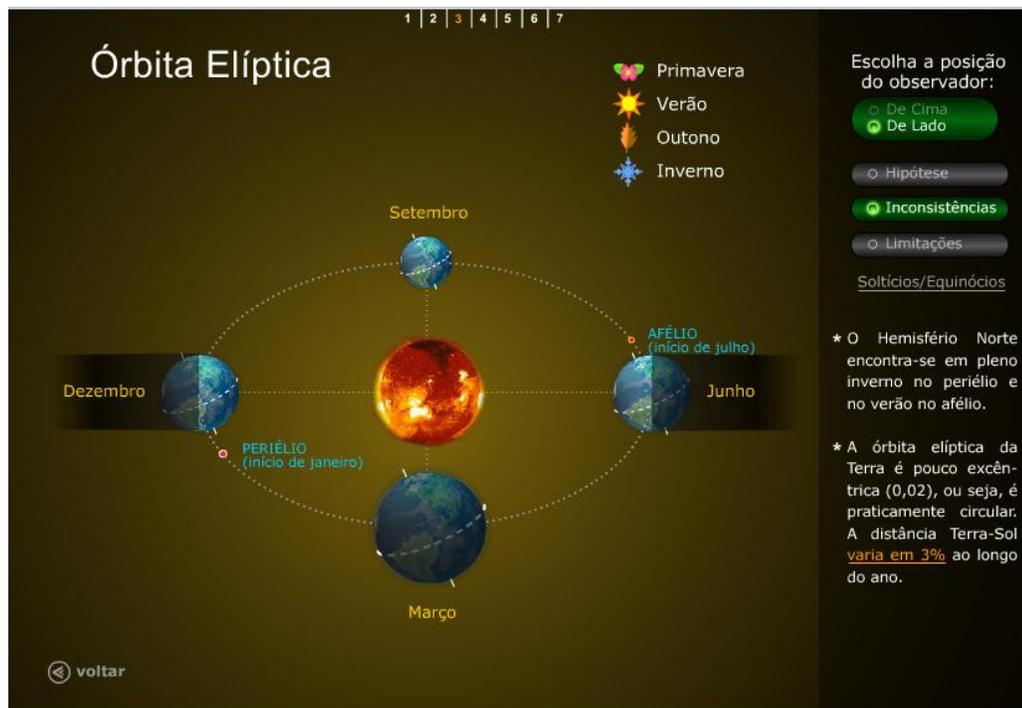


Figura 20 - Tela do aplicativo “A Terra no Espaço” com o subtema **Órbita Elíptica**



Figura 21 - Tela do aplicativo “A Terra no Espaço” com o subtema **Inclinação do Eixo de Rotação**



Figura 22 - Tela do aplicativo “A Terra no Espaço” com o subtema Inclinação do Eixo de Rotação

É possível observar, portanto que há indícios de que a falta de conhecimento sobre o conteúdo desenvolvido não foi o fator determinante para o impedimento da sequência de produção dos objetos de aprendizagem, mas que, portanto, é necessária a participação e orientação de um especialista do conteúdo abordado. A participação de um especialista em IHC permite que a interpretação dos requisitos levantados seja estruturada em protótipos interativos contemplando o design de interação do aplicativo e servindo de referência material para a equipe de desenvolvimento construir as funcionalidades.

As dúvidas surgidas até o nível de desenvolvimento que as duplas atingiram, eram principalmente relacionadas ao entendimento sobre a forma de construção e estruturação dos *scaffoldings*, portanto, há indícios de que a identificação e desenvolvimento de funcionalidades de interface com características ergonômicas e pedagógicas só pode ser viabilizada com a participação de um especialista em IHC.

7.2. Avaliação do Modelo de OA Baseado em Simulação

A avaliação do modelo de objeto de aprendizagem proposto foi realizada em duas etapas. A primeira consistiu em um estudo de caso realizado através de uma pesquisa semi-estruturada com o objetivo de avaliar a eficácia das propriedades do modelo proposto. A segunda teve o propósito de verificar através da análise fatorial, a correlação entre fatores Ergonômicos, Pedagógicos e os *scaffoldings* propostos pelo modelo.

7.2.1. Estudo de Caso

O estudo de caso foi realizado com o instrutor do curso de controle de tráfego aéreo do ICEA com mais de 20 anos de experiência e quatro controladores profissionais. A avaliação foi realizada através de entrevistas do tipo semi-estruturadas, garantindo a fluidez da análise no discurso do participante sem interrupções. O método foi definido com o objetivo de capturar as percepções sobre o protótipo, construindo assim uma avaliação prévia a partir das perguntas abaixo.

No estudo de caso, foi analisada a relação experimentação - objeto de aprendizagem para verificar se a estrutura aplicada no ClearaNCE proporcionaria uma boa receptividade e percepção da aplicação prática do conteúdo dentro de uma proposta diferenciada, verificando se a metodologia utilizada possibilitaria ou não a ação investigativa e reflexiva do aluno sobre os conceitos abordados. Nesse sentido, o estudo de caso teve como objetivos principais:

- Verificar se os conceitos foram apresentados de maneira correta e completa;
- Verificar a adequação e representação prática do conteúdo;
- Verificar se o cenário (contexto) dá uma noção concreta da importância dos conceitos teóricos;
- Aproximar a teoria de sua aplicação prática;
- Verificar os benefícios que o uso de objetos de aprendizagem poderiam trazer treinamentos na área de sistemas complexos;
- Identificar se a estruturação do conhecimento dentro das funcionalidades do aplicativo fornece como apoio a aprendizagem.

Para orientar o participante na utilização da ferramenta, foi realizada uma apresentação do objeto de aprendizagem à distância e posteriormente reuniões on-line para orientação em possíveis dúvidas relacionadas à funcionalidade do protótipo. Para que o participante tomasse conhecimento dos recursos disponibilizados no ClearaNCE, foi enviado um manual, explicando sua funcionalidade seguida de uma apresentação à distância, via Skype.

Análise das Entrevistas

O participante demonstrou boa receptividade ao protótipo e sua possível utilização em treinamentos futuros com vistas a representar a aplicação dos conceitos abordados. Contribuíram para esta receptividade os seguintes fatores: (a) qualidade na representação prática da aplicação dos conceitos; (b) aproximação da teoria e da prática; (c) percepção da importância e influência das informações na prática profissional.

A análise preliminar da entrevista nos permitiu colher impressões relacionadas a abordagem dos conceitos, definição do escopo e uso de recursos deste tipo em treinamentos para atuar em Sistemas Complexos.

Após explorar o protótipo, o participantes observaram que os conceitos básicos foram reunidos de maneira adequada com uma representação prática da aplicação e bastante sofisticada no que diz respeito a apresentação (interface). Embora considerem um grande desafio trabalhar com objetos de aprendizagem na área de treinamento para atuar em Sistemas Complexos, os participantes acreditam que sua utilização enriqueceria em muito o atual curso de formação de controlador de tráfego aéreo, pois traz uma visão bastante diferente dos conceitos necessários ao exercício da profissão.

Os participantes reforçaram que o OA conseguiu representar a aplicação dos conceitos de maneira correta e completa, facilitando o entendimento do procedimento operacional, ao dar uma noção concreta da importância do conteúdo abordado. Para eles, a apresentação de conceitos conjugados facilitam a generalização e contextualização da aplicação concreta do conceito, levando o aluno à conscientização de sua importância e a tomada de decisões mais conscientes.

Quando questionados sobre o escopo abordado no protótipo, os principais pontos levantados foram:

- Está adequado para os seus propósitos e consegue atender ao levantamento realizado no

Estudo de Campo quanto à necessidade de repensar as diferentes práticas educativas, a fim de produzir ambientes de aprendizagem que poderiam proporcionar melhorias na aquisição de conhecimentos, habilidades e atitudes, como pode ser visto no comentário a seguir:

“Objetos de aprendizagem como este [o protótipo] que apresentam os conceitos conjugados permitem generalizar e contextualizar a aplicação concreta do conceito e nos leva a conscientização de sua importância. A partir do momento em que o conceito está embasado, as decisões são mais conscientes.”

- Referente à adequação da atividade proposta (exercício) ao conteúdo, o participante declara que o protótipo cumpriu bem esta finalidade, colaborando para aproximar o aluno da realidade permitindo visualizar os conteúdos da aprendizagem ao ilustrar noções mais abstratas, conforme destacado nos comentários a seguir.

“A atividade proposta no objeto [de aprendizagem] permite que o aluno veja e entenda a aplicação prática da altimetria, item em que se pode verificar o maior grau de dificuldade na parte teórica do treinamento.”

“No ambiente de trabalho há uma série de demandas. (...) essa representação é um pouco complicada porque não temos uma ferramenta que represente isso. No momento em que o instrutor possa ter o objeto de aprendizagem para transcrever esta informação ao aluno, isso facilitaria em muito o entendimento.”

Foi ressaltado ainda que ao considerar na construção de recursos educacionais cenários que são provenientes da prática profissional, se possibilita uma real aproximação da teoria com a prática. Porém, observa-se que esperava a representação integral do procedimento em um único objeto de aprendizagem.

7.2.2. Pesquisa Experimental

Apesar do estudo de caso apresentar indícios da efetividade do modelo de objeto de aprendizagem proposto, julgou-se necessário, diante da proposta de integração entre

metodologias ágeis e conceitos de IHC, verificar a correlação de critérios ergonômicos, produtos da integração das duas disciplinas e critérios pedagógicos. Assim é possível identificar dentre os critérios de avaliação de uma interface, quais os de maior impacto no processo pedagógico e qual o ponto de equilíbrio entre uma interface mais objetiva e instrucionista e outra mais reflexiva e construtivista.

Ao tratar da avaliação de software educacional, muitos não fazem distinção entre as propriedades pedagógicas deste software e as propriedades características de um software convencional. Muitas metodologias de avaliação de software educacional foram propostas para avaliar a interface e as funcionalidades no aspecto ergonômico focando o grau de satisfação do usuário ligado diretamente à usabilidade deste software. Quando são considerados critérios pedagógicos, a avaliação é feita de forma distinta separando-os dos critérios ergonômicos.

O desenvolvimento e a avaliação, de aplicativos educacionais interativos, como multimídias e hipermídias pedagógicas, envolvem um grau de complexidade um pouco mais elevado devido à diversidade de atributos exigidos: técnicos, metodológicos, ergonômicos e pedagógicos.

Em projetos de interfaces inevitavelmente se integram, o projeto educacional e as exigências paradoxalmente contraditórias e convergentes da Ergonomia e da Pedagogia.

Muitas pesquisas tratam essas áreas de estudo em posições antagônicas, justificando que enquanto a Ergonomia tem como objetivo limitar os esforços físicos e cognitivos do usuário, tornando o uso da tecnologia mais fácil, reduzindo a complexidade de sua tarefa, a Pedagogia, busca o inverso, a tentar despertar no aluno os esforços necessários à construção do conhecimento. Porém ergonomistas e educadores buscam o mesmo objetivo e guardam semelhança com a Ergonomia tentando adaptar o trabalho informatizado ao ser humano, e a pedagogia trabalhando sobre os meios didáticos tentando obter satisfação e desempenho dos alunos no processo de aprendizagem de um conteúdo ou habilidade.

Em ambas as posições adotadas se fazem necessária à integração da Ergonomia e da Pedagogia, dadas às singularidades encontradas nessas áreas de estudo quanto a representação do conhecimento, apresentação e tratamento da informação.

Portanto, perante as novas tecnologias aplicadas a educação, o aprendiz tem uma postura mais participativa e passa a ser também, de um lado um usuário que na medida em que aprende o funcionamento de um produto, e de outro, um aluno, pois aprende o conteúdo por ele veiculado.

A distinção desses dois papéis desempenhados pelo estudante atualmente, vem se refletindo nas metodologias de avaliação de software educacional que necessita de uma diferenciação dos demais tipos de software, onde em geral, as funcionalidades, são projetadas para facilitar o cumprimento de uma tarefa ou o processo de comunicação e não para estimular o raciocínio sobre a ação executada.

Dentre as diversas metodologias de avaliação de software educacional pesquisadas, a única que mencionava a correlação entre critérios pedagógicos e ergonômicos, foi proposta pelo pesquisador do departamento de psicologia e tecnologia instrucional da universidade da Geórgia, Thomas Reeves (1994). O autor desenvolveu uma metodologia voltada especificamente para avaliar objetos de aprendizagem, em que define 14 critérios pedagógicos e 10 critérios relacionados a interface que são complementares entre eles.

Os critérios são avaliados com a marcação de um ponto sobre uma escala representada graficamente por uma seta dupla, não dimensionada numericamente, mas por conceitos antagônicos posicionados em cada extremidade da seta caracterizando o critério. Cada conceito tem uma polaridade positiva e outra negativa, determinada por seu posicionamento na extremidade da seta, sendo o negativo colocado à esquerda. As setas representativas de cada critério devem ser colocadas sobrepostas para viabilizar a análise. A análise é feita graficamente com a conclusão sendo obtida a partir da observação da disposição dos pontos marcados e com a ligação entre eles.

A metodologia elaborada por Reeves (2003) segue os seguintes critérios definidos para a abordagem de cada fator:

Critérios Pedagógicos:

1. Epistemologia - (Objetivista / Construtivista)

Epistemologia Objetivista:

- Conhecimento existe separado do saber;
- Realidade existe independente da existência da experiência;
- O conhecimento é adquirido de forma objetiva através dos sentidos;
- A aprendizagem consiste em adquirir verdades;
- O aprendizado pode ser medido precisamente com testes.

Epistemologia construtivista:

- O conhecimento não tem sentido sem a participação do homem;
- Embora a realidade exista independentemente, o que conhecemos dela, é individualmente construído;
- O conhecimento é construído subjetivamente baseado em experiências anteriores e em um processo metacognitivo ou reflexão;
- O aprendizado consiste na aquisição de estratégias que atenda a um objetivo;
- O aprendizado pode ser estimado através de observações e diálogos.

2. Filosofia Pedagógica - (Instrutivista / Construtivista)

Instrutivista:

- Enfatiza a importância de metas e objetivos independentes do aluno;
- Baseada na teoria comportamentalista;
- O aluno é visto como um agente passivo, como um recipiente vazio que será preenchido de conhecimento.

Construtivista:

- Enfatiza a primazia da intenção, experiência e estratégias metacognitivas do aluno;
- O conhecimento é construído individualmente pelo aluno;
- Garantir um ambiente de aprendizado o mais rico possível;
- Diferente da instrutivista, o aluno é visto como um indivíduo repleto de conhecimento pré-existente, atitudes e motivações.

3. Psicologia Subjacente - (Comportamental / Cognitiva)

Comportamental:

- Os fatores do aprendizado não são estados internos que podem ou não existir, mas comportamentos que podem ser diretamente observados;
- A instrução consiste na modelagem do comportamento desejável obtido através de estímulos e respostas.

Cognitiva:

- Enfatiza os estados mentais internos ao invés do comportamento psicológico;
- Reconhece que uma ampla variedade de estratégias de aprendizagem deve ser empregada considerando o tipo de conhecimento a ser construído.

4. Objetividade - (Precisamente focalizado / Não –focalizado)

Precisamente Focalizado: forma empregada em tutores e treinamentos.

Não- focalizados: forma empregada nos micromundos, simulações virtuais e ambientes de aprendizagem.

5. Sequenciamento Instrucional - (Reducionista / Construtivista)

Reducionista: o aprendizado sobre determinado conteúdo requer que todos os seus componentes sejam previamente entendidos.

Construtivista: o aluno é colocado em um contexto realístico, o qual irá requerer soluções de problemas, o apoio é introduzido de acordo com a necessidade individual do aluno.

6. Validade Experimental - (Abstrato / Concreto)

Abstrato: utiliza situações que não pertencem ao mundo real do aluno.

Concreto: se preocupa em sempre contextualizar o conteúdo apresentando situações da realidade.

7. O papel do instrutor - (Provedor de Materiais / Agente facilitador)

Professor - provedor de materiais : o programa instrutor é considerado “o dono do conhecimento”.

Agente Facilitador: o programa instrutor é visto como uma fonte de orientação e consulta.

8. Valorização do Erro - (Aprendizado sem erro / Aprendizado com a experiência)

Aprendizado sem erro: as instruções são organizadas de maneira que o aluno é induzido a responder corretamente.

Aprendendo com a experiência: apóia-se na máxima “a experiência é o melhor professor”. Provê oportunidades para que o aluno aprenda com seus próprios erros.

9. Motivação - (Extrínseca / Intrínseca)

Extrínseca: a motivação vem de fora do ambiente de aprendizado.

Intrínseca: integrada ao ambiente de aprendizado.

10. Estruturação - (Alta / Baixa)

Alta: os caminhos são previamente determinados.

Baixa: uma série de opções é oferecida de modo que o aluno possa escolher a ordem que desejar.

11. Acomodação de diferença individuais - (Não Existentes / Multi-facetadas)

Não existentes: considera os indivíduos iguais (homogêneos).

Multi-facetada: leva em consideração a diferença entre os indivíduos.

12. Controle do Aluno - (Não existente / Irrestrito)

Não Existente: todo o controle pertence ao programa.

Irrestrito: o aluno decide que seções estudar, que caminhos seguir, que material utilizar.

Obs.: Ainda é uma questão em aberto até que ponto o controle irrestrito é benéfico. Embora essa classificação esteja na posição à direita.

13. Atividade do Usuário - (Matemagênico / Generativo)

Matemagênico: ambientes de aprendizagem nos quais pretende-se capacitar o aluno a “acessar várias representações do conteúdo”.

Generativo: ambientes de aprendizagem que engajam o aluno no processo de criação, elaboração ou representação do conteúdo.

14. Aprendizado Cooperativo - (Não suportado / Integral)

Não suportado: não permite o trabalho cooperativo entre alunos (em pares ou grupos).

Integral: permite o trabalho cooperativo de modo que os objetivos sejam compartilhados beneficiando o aluno tanto instrucionalmente quanto socialmente.

A seguir, são apontados dez critérios selecionados e adaptados por Reeves (2003) a partir de critérios ergonômicos para avaliar a interface, visando correlacioná-los aos critérios pedagógicos também apontados pelo autor.

- 1. Facilidade de Utilização - (Difícil / Fácil)**
- 2. Navegação - (Difícil / Fácil)**
- 3. Carga Cognitiva - (Não gerenciável / gerenciável)**
- 4. Mapeamento - habilidade em rastrear ações do aluno - (Nenhum / Poderoso)**
- 5. Design de Tela - (Princípios violados / Princípios respeitados)**
- 6. Compatibilidade Espacial do Conhecimento - (Incompatível / Compatível)**
- 7. Apresentação da informação - (Confusa / Clara)**
- 8. Integração das Mídias - (Não coordenada / Coordenada)**
- 9. Estética - (Desagradável / Agradável)**
- 10. Funcionalidade Geral - (Não funcional / Altamente Funcional)**

A partir da definição destes critérios, o presente estudo foi definido com o objetivo de:

1. Verificar a consistência das respostas dadas aos itens das duas escalas (ergonômica e pedagógica) entre os respondentes.
2. Verificar o nível de receptividade e percepção da aplicação prática do conteúdo dentro de uma proposta diferenciada, verificando se a metodologia utilizada possibilitaria ou não a ação investigativa e reflexiva do aluno sobre os conceitos abordados.
3. Analisar a correlação entre as dimensões ergonômicas e pedagógicas e assim, conseqüentemente entre seus fatores sobre o protótipo.

Com estes três objetivos, o estudo foi realizado através da aplicação de um questionário de múltipla-escolha onde as duas dimensões e os respectivos, 14 e 10, fatores foram explicados detalhadamente para cada participante.

Uma pequena amostra de respondentes foi formada por 1 instrutor de controle de tráfego aéreo que possuía mais de 20 anos de experiência, 4 alunos do curso de formação e 4 controladores profissionais, identificados pela sigla OP, já atuantes na área com em média acima de 5 anos de experiência.

Para adequar a metodologia de Reeves (2003) à presente análise, foi necessário estabelecer uma escala numérica para a definição dos critérios apontados. No caso, foi determinada uma escala de espectro reduzido objetivando à eficiência na definição do grau de intensidade de opinião sobre o assunto pesquisado. Assim foram definidos os valores inteiros intervalares -1, 0 e 1, sendo “-1” o valor menor referente à esquerda da linha gráfica da metodologia de Reeves (2003), “0” representaria uma neutralidade, e “1” o valor à direita.

Para analisar as respostas foram aplicadas técnicas multivariadas onde, normalmente, as variáveis são determinadas pelos N itens do instrumento (no caso, questionário) dispostas em N colunas e os dados são as respostas de cada um dos M respondentes dispostos em M linhas, formando uma matriz N x M.

Consistência das Respostas

Uma primeira questão que precisa ser investigada é se os três tipos de respondentes (Instrutor, Aluno e Operadores) tem uma percepção comum sobre os 24 itens que compõem os dois critérios (Pedagógico e Ergonômico). Para tanto, faremos uma inversão da matriz de dados colocando os respondentes como “variáveis” e os 24 itens como “sujeitos”. A consistência da opinião entre os respondentes foi determinada pelo coeficiente de alfa de Cronbach para medir a confiabilidade da análise, obtendo o grau de homogeneidade dos componentes da escala, ou seja, a consistência interna dos itens. Segundo Hair (1998), essa escala varia de 0 a 1, sendo o valor mínimo referente à ausência de consistência, o valor máximo representativo da consistência plena e o valor médio de 0,70 atribuído à forte consistência de pesquisas exploratórias. O alfa de Cronbach padronizado pode ser calculado através da fórmula de Kuder-Richardson (KR_{21}).

$$KR_{21} = \frac{k\bar{r}}{1 + (k - 1)\bar{r}}$$

Onde k é o no de itens da escala, \bar{r} é a média aritmética entre os $k(k-1)/2$ coeficientes da matriz k x k de correlação entre os k itens. Assim, para análise da consistência das respostas entre os 9 respondentes, consideramos cada respondente como variável e a resposta em cada item

do questionário como “sujeito” (objeto) do estudo, formando assim uma matriz de 9 variáveis por 24 sujeitos. A tabela 1 apresenta a matriz simétrica de correlação 9 x 9 entre os respondentes. A média aritmética entre os $9(9-1)/2 = 36$ coeficientes de correlação desta matriz é $= 0,87$ que, aplicando a fórmula KR21, resulta no valor de alfa padronizado mostrado na tabela 2.

	Instrutor	Aluno1	Aluno2	Aluno3	Aluno4	OP1	OP2	OP3	OP4
Instrutor	1,000	,769	,769	,832	,832	,968	,861	,867	,910
Aluno1	,769	1,000	1,000	,918	,918	,814	,886	,855	,841
Aluno2	,769	1,000	1,000	,918	,918	,814	,886	,855	,841
Aluno3	,832	,918	,918	1,000	1,000	,872	,877	,836	,883
Aluno4	,832	,918	,918	1,000	1,000	,872	,877	,836	,883
OP1	,968	,814	,814	,872	,872	1,000	,848	,899	,938
OP2	,861	,886	,886	,877	,877	,848	1,000	,765	,822
OP3	,867	,855	,855	,836	,836	,899	,765	1,000	,965
OP4	,910	,841	,841	,883	,883	,938	,822	,965	1,000

Tabela 1 - Matriz de correlação entre os respondentes

Alpha de Cronbach Padronizado	Nº de casos	Nº de Variáveis
,984	24	9

Tabela 2 - Valor registrado do Alpha de Cronbach entre os 9 respondentes em análise.

É interessante destacar que, mesmo considerando a heterogeneidade da experiência profissional entre os respondentes (aluno, operador em serviço e instrutor), foi obtida uma excelente consistência para as respostas dadas por eles aos 24 itens do questionário.

Isto significa também que se pode tomar a média sobre todos os 9 avaliadores como um indicador de opinião confiável sobre cada item, conforme mostrado separadamente para os critérios pedagógicos (Tabela 3.a) e ergonômicos (Tabela 3.b).

Na tabela 3a, é possível observar que a média estatística dos critérios pedagógicos com o índice em 0,36, indica que houve uma tendência positiva segundo metodologia de Reeves, com o modelo de software educacional tendendo para uma linha de interação mais aberta e experimental. O desvio padrão obtido na análise confirma essa tendência e a concordância entre os respondentes ao registrar uma média relativamente baixa de 0,22.

É possível observar alguns critérios com médias muito negativas com índice -1.00, o que é justificado pela ausência de funcionalidades que contemplassem esses critérios como é o caso da Acomodação das Diferenças, a Atividade do Usuário e o Aprendizado Colaborativo.

Já outros critérios com médias pouco negativas ou com baixos índices positivos como Papel do Instrutor e Controle do Aluno, apresentam indícios de que a estrutura do aplicativo permite que o instrutor atue como apenas como um agente facilitador e que o próprio aluno adéqüe algumas características do aplicativo de acordo com o seu nível de conhecimento.

Critérios Pedagógicos	Escala		Média Estatística	Desvio Padrão
	Fechada e Diretiva (Média Estatística < 0)	Aberta e Experimental (Média Estatística > 0)		
Epistemologia	Objetivista	Construtivista	1.00	0
Filosofia Pedagógica	Instrutivista	Construtivista	1.00	0
Psicologia Subjacente	Comportamental	Cognitiva	1.00	0
Objetividade	Precisamente focalizado	Não focalizado	0.89	0.11
Sequenciamento Instrucional	Reducionista (Linear)	Construtivista (Sistêmico)	0.67	0.17
Validade Experimental	Abstrato (Situação fictícias)	Concreto (Situações reais)	1.00	0
Papel do Instrutor	Provedor de Materiais	Agente facilitador	-0.33	0.17
Valorização do Erro	Aprendizado sem erro	Aprendizado com Experiência	1.00	0
Motivação	Extrínseca	Intrínseca	1.00	0
Estruturação	Alta	Baixa	0.78	0.15
Acomodação de Dif. Individuais	Não existente	Multi-facetadas	-1.00	0
Controle do Aluno	Não existente	Irrestrito	0.00	0
Atividade do Usuário	Matemagênico (representações)	Generativo (criação, elaboração)	-1.00	0
Aprendizado Cooperativo	Não existente	Integral	-1.00	0
Média			0.36	0.22

Tabela 3a - Média estatística e desvio padrão dos critérios pedagógicos.

Na tabela 3b, é possível observar que a média estatística dos critérios ergonômicos com o índice em 0,11, indica que houve uma tendência um pouco positiva segundo a metodologia de Reeves, tendendo para uma linha de interação mais objetiva. O desvio padrão obtido na análise confirma essa tendência e a concordância entre os respondentes ao registrar uma média relativamente baixa de 0,21.

É importante observar a média de cada critério, que apesar da média geral apresentar uma tendência do aplicativo à interação objetiva, os critérios ergonômicos relativos à interação motora relativa à aprendizagem experimental, obtiveram índices negativos, ficando mais próximos da interação reflexiva. Já os critérios ergonômicos relativos às representações gráficas, apresentaram índices positivos.

Critérios Ergonômicos	Escala		Média Estatística	Desvio Padrão
	Interação reflexiva (Média Estatística < 0)	Interação objetiva (Média Estatística > 0)		
Facilidade de Utilização	Difícil	Fácil	-0.33	0.5
Navegação	Difícil	Fácil	-0.33	0.5
Carga Cognitiva	Não gerenciável- Não intuitiva	Gerenciável-Intuitiva	-0.56	0.53
Mapeamento	Nenhum	Poderoso	-1.00	0
Design de Tela	Princípios violados	Princípios respeitados	0.44	0.53
Compatibilidade Espacial do Conhecimento	Incompatível	Compatível	1.00	0
Apresentação da Informação	Confusa	Clara	0.22	0.44
Integração das Mídias	Não coordenada	Coordenada	0.22	0.44
Estética	Desagradável	Agradável	1.00	0
Funcionalidade Geral	Não funcional	Altamente funcional	0.44	0.53
Média			0.11	0.21

Tabela 3b - Média estatística e desvio padrão dos critérios ergonômicos.

Por outro lado, podemos investigar também separadamente a percepção de cada tipo de sujeitos da amostra sobre todos os itens criteriais. (Tabela 4). A análise das médias das opiniões dos respondentes tomada sobre todos os 24 itens envolvendo critérios pedagógicos e ergonômicos indica que os 3 tipos de sujeitos da amostra têm uma percepção de neutra à ligeiramente positiva, o que significa que todos reconhecem um certo grau de construtivismo nos critérios pedagógicos e certo grau de usabilidade nos critérios ergonômicos. Pode-se dizer também que esta tendência na percepção é ligeiramente maior para os mais experientes: instrutor (0,29) e operadores profissionais (0,29) comparado aos alunos (0,19). Outro estudo que seja feito com erro experimental menor, ou seja, com uma amostra maior poderá confirmar estatisticamente esta tendência.

Essa identificação do instrutor e dos profissionais com as propriedades específicas do cenário operacional em que eles atuam, podem indicar que os requisitos desenvolvidos no

aplicativo atendem às características da atividade desempenhada e aos conceitos abordados no aplicativo.

Respondente	Média	Erro Padrão
Instrutor	0,29	0,16
Aluno1	0,13	0,17
Aluno2	0,13	0,17
Aluno3	0,25	0,16
Aluno4	0,25	0,16
OP1	0,33	0,17
OP2	0,04	0,18
OP3	0,38	0,16
OP4	0,42	0,16

Tabela 4 - Distribuição das medias das respostas dos respondentes nos 24 itens

Análise da Associação Entre Fatores Pedagógicos e Ergonômicos:

Para verificar a correlação entre os fatores pedagógicos, fatores ergonômicos e assim identificar o núcleo de convergência entre esses critérios que caracterizam os *scaffoldings* foi utilizada a metodologia de análise fatorial.

Em situações como esta, onde há necessidade de analisar um grande número de fatores e variáveis, a análise fatorial se faz necessária por ser um método que permite, através de um conjunto de métodos estatísticos, explicar o comportamento de um grande número de variáveis em um número pequeno de fatores.

Essa metodologia permite encontrar a importância relativa de cada fator pesquisado através de uma combinação linear das variáveis mais correlacionadas entre si. A análise é realizada, levando em consideração a variância total nos dados para redução e sumarização do número de variáveis.

De acordo com Hair *et al.* (1998), cada variável na análise, pode ser definida e ter a variância explicada ou não por uma combinação linear de fatores comuns a partir de uma escala que varia de 0 a 1. São classificadas de comunalidade as que se aproximarem de 1, indicando que

são explicadas pelos fatores, já as que não possuem variabilidade e portanto não são explicadas, indicam valor aproximado a 0.

Assim, para análise da correlação entre os 24 critérios, sendo 14 pedagógicos e 10 ergonômicos, consideramos cada critério como variável e a resposta em cada item do questionário como “sujeito” (objeto) do estudo, formando assim uma matriz de 24 variáveis por 9 sujeitos.

Na análise fatorial, 13 variáveis foram eliminadas por não apresentarem variabilidade, havendo, portanto uma redução dos 14 critérios pedagógicos e dos 10 ergonômicos, para 7 ergonômicos e 4 pedagógicos.

A tabela 4 apresenta a matriz simétrica de correlação 11 x 11 entre os itens restantes da análise

	Obj.	Seq. Instr.	Papel instrutor	Estrut.	Fac. Util.	Navegação	Carga Cog.	Design Tela	Apres. Inf.	Int. Mídias	Func. Geral
Obj.	1,000	-,250	,500	,661	,500	,500	,316	,316	,189	,189	,316
Seq.Instr.	-,250	1,000	-,500	-,378	,500	,500	,158	,158	-,189	-,189	,158
Papel Instrutor	,500	-,500	1,000	,756	,000	,000	-,316	-,316	,378	,378	-,316
Estrut.	,661	-,378	,756	1,000	,189	,189	-,060	-,060	,286	,286	-,060
Fac. Util.	,500	,500	,000	,189	1,000	1,000	,632	,632	,378	,378	,632
Navegação	,500	,500	,000	,189	1,000	1,000	,632	,632	,378	,378	,632
Carga Cog.	,316	,158	-,316	-,060	,632	,632	1,000	1,000	,598	,598	1,000
Design Tela	,316	,158	-,316	-,060	,632	,632	1,000	1,000	,598	,598	1,000
Apres. Inf.	,189	-,189	,378	,286	,378	,378	,598	,598	1,000	1,000	,598
Int. Mídias	,189	-,189	,378	,286	,378	,378	,598	,598	1,000	1,000	,598
Func. Geral	,316	,158	-,316	-,060	,632	,632	1,000	1,000	,598	,598	1,000

Tabela 4 - Matriz de correlação entre critérios pedagógicos e ergonômicos restantes análise.

Foi feita então com o SPSS V11 uma análise fatorial, usando Método de Extração da Análise dos Componentes Principais e critério de corte Auto-Valor > 1 e um limite de 25 interações para a análise de auto-consistência convergir. A Tabela 5 mostra a matriz de componentes resultantes desta análise, rotacionada pelo método Varimax que enfatiza a estrutura fatorial para os 3 fatores que apresentam Auto-Valores > 1 e um valor explicado de 89% da variância total.

Matriz dos Componentes Rotacionados pelo Método VARIMAX			
VAR Pedagógicas	F1	F2	F3
OBJETIVI	0,144	0,466	0,719
SEQUENCI	-0,168	0,68	-0,495
PAPEL_DO	-1,80E-02	-0,208	0,934
ESTRUTUR	2,39E-02	8,20E-02	0,91
VAR Ergonômicas	F1	F2	F3
FACILIDA	0,356	0,891	0,162
NAVEGAÇÃ	0,356	0,891	0,162
CARGA_CO	0,847	0,451	-0,162
DESIGN_D	0,847	0,451	-0,162
APRESENT	0,883	-4,88E-02	0,323
INTEGRAÇ	0,883	-4,88E-02	0,323
FUNCIONA	0,847	0,451	-0,162

Tabela 5 - Matriz dos Componentes Rotacionados pelo Método Varimax

A partir do resultado da análise, foram excluídas *Ad Hoc* as cargas das variáveis nos fatores (*Factor Loading*) muito próximos de zero (em vermelho) e inferiores a 0,20 (em amarelo). Os valores considerados muito relevantes foram destacados em azul e os não tão relevantes foram destacados na cor cinza.

Dos fatores resultantes da análise, sendo F1, F2 e F3 é possível identificar uma ordem a partir da carga fatorial.

Em F1, houve uma concentração dos valores positivos mais altos junto as variáveis ergonômicas indicando, portanto uma tendência para o lado positivo classificado como construtivista segundo a escala de Reeves.

Em F3, houve uma concentração dos valores positivos mais altos junto as variáveis pedagógicas indicando novamente uma tendência para o lado positivo classificado como construtivista segundo a escala de Reeves.

Em F2, houve um equilíbrio na distribuição dos valores entre as variáveis, tanto as pedagógicas quanto as ergonômicas, indicando que essas variáveis configuram um novo fator ergo-pedagógico resultante de um ponto mediano entre as tendências construtivista e instrutivista.

Esse resultado aponta indícios de que os critérios ergonômicos mais ligados a questão visual (e que tendem ao lado positivo), são exatamente os que se contrapõem aos critérios pedagógicos característicos de uma aprendizagem mais construtivista. Já os critérios ergonômicos

mais ligados à interação são os que se alinham com os critérios pedagógicos característicos de uma aprendizagem mais construtivista. É possível assim concluir que há indícios de que o design de interação possui forte correlação com os fatores pedagógicos.

7.3. Considerações Finais do Capítulo

Na avaliação da metodologia integrada de desenvolvimento foi possível perceber a viabilidade de desenvolvimento do modelo de OA proposto, porém, evidenciou-se que a identificação e desenvolvimento de funcionalidades de interface com características ergonômicas e pedagógicas só pode ser viabilizada com a participação de um especialista em IHC atuando em conjunto com um especialista da área do conhecimento contextualizado no OA. Portanto há indícios de que o desempenho no desenvolvimento do OA verificado com alunos de ensino médio, a partir da aplicação do modelo proposto, pode ser ultrapassado se aplicado em uma equipe de especialistas. É possível concluir também que a metodologia proposta, permitiu a partir da integração entre as áreas de ES e IHC realizarem o levantamento preciso dos requisitos, identificarem cenários e *scaffoldings* e desenvolver funcionalidades com características mais pedagógicas.

No estudo de caso, foi atestada a eficiência das propriedades pedagógicas da ferramenta. De modo geral, foi julgado importante que a inclusão de recursos como o ClearaNCE em cursos de formação para atuar em sistemas complexos é de suma importância, uma vez que apóiam a compreensão de procedimentos em que há combinação de diferentes variáveis através representações que auxiliariam na eliminação ou redução de lacunas que poderiam levar à dificuldades durante realização do estágio prático.

Através deste estudo pudemos verificar a importância da participação de um especialista no processo de construção de um recurso educacional a fim de evidenciar a combinação que deve existir entre os componentes da grade curricular e a maneira como são aplicados em uma situação real. Muito embora estes resultados não sejam conclusivos, eles mostram indícios de que a utilização de objetos de aprendizagem contextualizados pode apoiar a abstração e aplicação dos conceitos abordados em uma situação real.

Das observações feitas concluímos que houve indícios de que o protótipo conseguiu representar a aplicação dos conceitos, possibilitando a percepção de sua importância, e sua utilização em um treinamento real poderia aproximar a etapa teórica da atividade prática.

Na pesquisa experimental é interessante destacar que, mesmo considerando a heterogeneidade da experiência profissional entre os respondentes (aluno, operador em serviço e instrutor), foi obtida uma excelente consistência para as respostas dadas por eles aos 24 itens do questionário. Isto significa também que se pode tomar a média sobre todos os 9 avaliadores como um indicador de opinião confiável sobre cada item.

Na análise fatorial, 13 variáveis foram eliminadas por não apresentarem variabilidade, havendo, portanto uma redução dos 14 critérios pedagógicos e dos 10 ergonômicos, restando 11 critérios sendo, 7 ergonômicos e 4 pedagógicos. As 11 variáveis restantes configuram um novo fator ergo-pedagógico resultante de um ponto mediano entre as tendências construtivista e instrutivista.

Capítulo 8 - Considerações Finais e Trabalhos Futuros

Neste capítulo é apresentado o resumo da pesquisa descrita nessa dissertação, explicitando as suas contribuições, problemas encontrados ao longo do seu desenvolvimento e sugestões para prosseguimento do trabalho.

8.1. Resumo do Trabalho

Treinamento é uma estratégia composta de atividades de qualificação com vistas a preparar o profissional para determinado cargo ou função. Porém, ainda são muitos os treinamentos que adotam a abordagem de formação tradicional, firmada na análise detalhada e estática do trabalho, tornando um desafio para as equipes de treinamento a reestruturação de seu projeto instrucional, que foi o ponto de partida para esse trabalho.

Visando atender o problema mencionado, procuramos verificar a possibilidade de aplicação das práticas ágeis de desenvolvimento de software na definição dos contextos em que o conceito teórico é aplicado e, conseqüentemente, na produção de recursos didáticos. A constatação de que tal aplicação era possível foi a motivação principal para o desenvolvimento dessa dissertação.

Além de propor um modelo que pode ser utilizado em fases distintas do projeto instrucional, foi desenvolvido um protótipo do recurso didático ClearaNCE. O modelo proposto baseia-se no desenvolvimento dirigido ao comportamento, objetivando contribuir para potencializar os cenários utilizados no processo de ensino e aprendizagem.

Quanto ao protótipo, também foi considerado para o seu desenvolvimento as boas práticas do desenvolvimento dirigido ao comportamento, que visam explicitar os conhecimentos e comportamentos necessários para alcançar os resultados esperados, contribuindo assim, para apoiar os alunos na fixação dos conceitos.

Para avaliar a viabilidade de nossa proposta, no que diz respeito ao protótipo, foi realizada uma avaliação prévia com um controlador de tráfego aéreo com mais de 20 anos de experiência profissional.

Contudo, os resultados obtidos foram importantes para mostrar uma primeira tentativa de validar a hipótese que orienta essa pesquisa. Os resultados desse estudo apresentaram indícios de que a hipótese pode ser confirmada, ou seja, a partir do momento que se constroem recursos didáticos contextualizados, a compreensão e fixação dos conceitos teóricos são potencializadas.

Esse resultado obtido no estudo de caso com o profissional de controle de tráfego aéreo, apesar de não ser conclusivo, apontou indícios da confirmação da hipótese formulada neste trabalho, o que pode ser observado também na análise fatorial. Essa análise foi realizada com o intuito de confirmar a hipótese que orienta essa pesquisa através da verificação da correlação entre fatores pedagógicos e ergonômicos.

Nesta etapa, o aplicativo foi submetido à avaliação de um instrutor, quatro alunos do curso de controlador de tráfego aéreo e quatro profissionais, tendo como parâmetro de avaliação dez fatores ergonômicos e quatorze pedagógicos.

Os resultados da análise dos dados apresentaram novamente indícios de que a hipótese pode ser confirmada, ou seja, a partir do momento que se construam recursos didáticos contextualizados a compreensão e fixação dos conceitos teóricos são potencializadas e ainda foi possível identificar quais fatores possuem maior relevância e contribuíram mais dentro do processo.

Apesar dos estudos apontarem indícios da relevância e efetividade da hipótese que orienta este trabalho, os resultados não podem ser considerados conclusivos devido à limitação do tamanho da amostra pesquisada.

Posteriormente foi realizada a avaliação do modelo que integrado de IHC e metodologias ágeis de ES para verificar a viabilidade do desenvolvimento de aplicativos educacionais apropriados ao treinamento de operadores de sistemas complexos que contemplem funcionalidades e representações da variabilidade do cenário operacional.

O estudo foi realizado com três duplas de alunos de ensino médio técnico em informática do Colégio Pedro II. Cada dupla foi responsável pelo desenvolvimento de aplicativos diferentes escolhendo o conteúdo a ser desenvolvido a partir da apostila oficial do curso de controle de tráfego aéreo do ICEA. Foi observado que apesar das duplas apresentaram desempenho similar e satisfatório, sendo que uma delas apresentou resultados melhores, que pode ser explicado pela escolha do tema fazer parte da matriz curricular de alunos do ensino médio e assim, possuem um entendimento melhor dos conceitos do capítulo abordado na apostila do ICEA.

O estudo assim apresentou indícios da viabilidade do desenvolvimento dos aplicativos através do modelo proposto, porém com ressalvas, indicando a necessária participação de especialistas como consultores junto à equipe de desenvolvimento.

Podemos destacar nos estudos realizados a utilização de metodologias ágeis adaptadas à proposta e principalmente a utilização de cenários e protótipos como linguagens e veículos de comunicação entre as partes envolvidas nos projetos. Isso permitiu que todos os envolvidos pudessem observar e compreender cada aspecto do desenvolvimento do aplicativo sob a ótica diferente de cada área. Com isso houve uma aproximação muito grande entre as áreas, o que permitiu identificar oportunidades para o desenvolvimento de funcionalidades e *scaffoldings* que trouxeram contribuições

significativas ao processo de aprendizagem com manipulação direta de conceitos através da interface dos aplicativos.

8.2. Contribuições da Dissertação

Com este trabalho, acredita-se ter apresentado diversas contribuições para a área de Treinamento e Informática na Educação. A principal contribuição desta dissertação é a proposta de um modelo baseado nas práticas ágeis de desenvolvimento de software, além de outras, que serão destacadas a seguir:

- Metodologia para o levantamento de requisitos a partir do registro e representação do conhecimento tácito de *experts* utilizado dentro do projeto instrucional.
- Apresentação de um modelo de desenvolvimento de software educacional, onde o design de interação é melhor adequado ao projeto instrucional a partir da integração de metodologias ágeis de ES e conceitos de IHC.
- Apresentação de um modelo para a contextualização da aprendizagem, capaz de indicar a aplicação do conceito teórico em diferentes situações;
- As práticas ágeis de desenvolvimento de software também podem ser adotadas em outras fases do projeto instrucional, inclusive em sua própria construção;
- Proposta de uso de OA no contexto de treinamento para atuar em Sistemas Complexos, visto que a maioria dos OA encontrados nos repositórios públicos está voltada para o ensino fundamental e médio;
- Apresentação dos resultados obtidos a partir de um estudo de caso, que contribuíram para verificar a viabilidade da solução apresentada para o problema na dissertação, pois mostraram indícios de que a aplicação de uma das práticas ágeis de desenvolvimento de software ajudou a construir um recurso didático cuja ênfase está no comportamento esperado para o exercício da prática profissional. Esses resultados podem servir de base para novos trabalhos.

8.3. Problemas Encontrados

Apesar da busca de uma base na literatura, e dos estudos realizados, ao longo do desenvolvimento dessa pesquisa nos deparamos com alguns problemas.

Primeiramente foi a dificuldade de acesso a informação e disponibilidade de um grande número de participantes para a pesquisa, com a dificuldade em agendar entrevistas com os instrutores e alunos do curso de formação de controlador de tráfego aéreo, visto que dependíamos do cronograma do curso. Somando-se a isso, ao longo do período desta pesquisa, ocorreram acidentes na aviação comercial, que culminaram na restrição do acesso a informações devido ao fato do controle do tráfego aéreo estar sob a responsabilidade da Força Aérea Brasileira. Isso impactou também no acesso a instituições da área de ensino da Força Aérea Brasileira, único órgão do país credenciado para licenciar o profissional de controle de tráfego aéreo.

Outro fator complicador, foi o pouco tempo para a implementação completa de um objeto de aprendizagem abrangendo o contexto em sua totalidade e realização de um estudo de caso mais aprofundado. Por isso, as pesquisas foram realizadas a partir de protótipo de objetos de aprendizagem, com vistas a permitir uma avaliação prévia da solução proposta.

Assim, as pesquisas, enfocam apenas os benefícios que o modelo proposto poderia trazer para os alunos em oposição aos atuais recursos, ficando distante da realidade das instituições de ensino da Força Aérea Brasileira. Sendo assim, para não exceder o tempo e escopo de uma dissertação de mestrado, desconsideramos vários pontos os quais apresentamos como sugestões de trabalhos futuros.

8.4. Trabalhos Futuros

Para trabalhos futuros, tanto em nível teórico, como de desenvolvimento de novas aplicações e realização de novos estudos de caso. A seguir são apresentadas as perspectivas para o prosseguimento dessa pesquisa.

- A finalização da construção do objeto de aprendizagem, visto que é ainda um protótipo, incluindo aspectos que não foram implementados e melhorias de alguns existentes como:

- *Feedback* informativo ao aluno a respeito de seu desempenho;

- Mapeamento das ações do aluno sobre o aplicativo;
 - Seleção dinâmica das atividades de aprendizagem de acordo com o desempenho cognitivo do aluno.
-
- Análise da metodologia de formação básica do controlador de tráfego aéreo e proposta de novas diretrizes para uma nova geração de treinamentos promovidos pelas instituições de ensino da Força Aérea Brasileira.
 - Investigação e implementação de modelos para captura da cognição que permitam compreender as atividades mentais envolvidas na realização de determinada tarefa. Desenvolvimento de um módulo para captura das ações do usuário utilizando a técnica do walkthrough para registrar os passos na execução de tarefas e atividades no aplicativo e a partir da análise desses dados, identificar a lógica estabelecida pelo aluno no processo de aprendizagem.
 - Investigação e implementação de artefatos que permitam a avaliação do comportamento necessário para as atividades práticas. Desenvolvimento de um módulo para automatizar o levantamento de requisitos através da identificação de palavras-chaves a partir do método GOMS.
 - Investigação e implementação de outros recursos e verificar a aplicação e o potencial da proposta no desenvolvimento de outros tipos de software educacionais.
 - Realização de estudos de caso nas instituições de ensino da Força Aérea Brasileira.
 - Realização de nova pesquisa experimental com um maior número de participantes.

Por fim, acredita-se que a abordagem tratada nessa dissertação referente a utilização da metodologia integrada proposta, tem um potencial não só para aplicação em programas de treinamento e desenvolvimento, mas também em outras áreas que requeiram o desenvolvimento de competências, visto que tais práticas possibilitam estabelecer excelência na geração e disseminação do conhecimento uma vez que expressam efetivo compromisso com os resultados esperados.

Bibliografia

ASTELS, D., (2003) Test-Driven Development: A Practical Guide; Prentice-Hall PTR.

AUSUBEL, D. P. (2003) Aquisição e retenção de conhecimentos: uma perspectiva cognitiva, Lisboa: Plátano, 2003.

AZEVEDO, R., & Hadwin, A.F., (2005 a). Scaffolding self-regulated learning and metacognition: Implications for the design of computer-based scaffolds. *Instructional Science (Special Issue on Scaffolding Self-Regulated Learning and Metacognition: Implications for the Design of Computer-Based Scaffolds)*.

Disponível em: http://azevedolab.autotutor.org/publications_inpress.htm , Acessado em fev. 2010

_____, Cromley, J.G., Winters, F.I., Moos, D.C., & Greene, J.A. (2005 b). Adaptive human scaffolding facilitates adolescents' self-regulated learning with hypermedia. *Instructional Science (Special Issue on Scaffolding Self-Regulated Learning and Metacognition: Implications for the Design of Computer-Based Scaffolds)*.

Disponível em: http://azevedolab.autotutor.org/publications_inpress.htm , Acessado em fev. 2010

_____, Cromley, J.G., Winters, F.I., Moos, D.C., & Greene, J.A. (2006 a). Using computers as metacognitive tools to foster students' self-regulated learning. *Technology, Instruction, Cognition, and Learning Journal*.

Disponível em: http://azevedolab.autotutor.org/publications_inpress.htm , Acessado em fev. 2010

_____, Hmelo-Silver, C.. (2006 b) Understanding Complex Systems: Some Core Challenges. THE JOURNAL OF THE LEARNING SCIENCES, Lawrence Erlbaum Associates, 2006.

Disponível em: http://azevedolab.autotutor.org/publications_inpress.htm , Acessado em fev. 2010

_____, & Moos, D.C., (2006 c). The role of goal structure in undergraduates' use of self-regulatory variables in two hypermedia learning tasks. *Journal of Educational Multimedia and Hypermedia*. Disponível em: http://azevedolab.autotutor.org/publications_inpress.htm , Acessado em fev. 2010

_____, (2007). Understanding the complex nature of self-regulated learning processes in learning with computer-based learning environments: An introduction. *Metacognition and Learning*. Disponível em: http://azevedolab.autotutor.org/publications_inpress.htm , Acessado em fev. 2010

_____, & Jacobson, M. (2008). Advances in scaffolding learning with hypertext and hypermedia: A summary and critical analysis. *Educational Technology Research & Development*. Disponível em: http://azevedolab.autotutor.org/publications_inpress.htm , Acessado em fev. 2010

BOYLE, T., Cook, J.; Windle, R.; Wharrad, H.; Jeeder, D.; Alton, R. (2006) *An Agile method for developing learning objects*. Proceedings of the 23rd annual ascilite conference: Who's learning? Whose technology? Sydney, Australia.

BECK, Kent. (2000) *Extreme Programming Explained: Embrace Change*; Addison-Wesley.

_____, (2003) *Test-Driven Development By Example*; Addison-Wesley.

_____, (2001) *Manifesto for Agile Software Development*.

Disponível em: <http://www.agilemanifesto.org> Acessado em nov. 2009

BRUNER, J. (1996) Vygotsky's zone of proximal development: The hidden agenda. 1984. In Schunk, D. *Learning Theories: an education perspective*. Columbus, OH: Merrill Prentice-Hall.

CARD, S. K.; Moran, T. P.; Newell, A. (1986) The model Human Processor: An engineering model of human performance 45-1-35. I: Handbook of Perception and Human Performance V. II Cognitive Process and Performance, New York: John Wiley.

CARROLL, J. M. and Rosson, M. B. (2002). *Usability Engineering: Scenario-Based Development of Human Computer Interaction* Morgan Kaufmann, San Francisco.

CARVALHO, Antônio Sérgio Lins de. (2009) Transferindo conhecimento tácito. Disponível em: <http://www.e-papers.com.br> Acessado em set. de 2009.

CHURCHVILLE D., (2008), *Agile Thinking: Leading Successful Software Projects and Teams*.

CRANDALL, B., Klein, G. and Hoffman, R. (2006) “Working Minds: A Practitioner’s Guide To Cognitive task Analysis”, The MIT Press, Cambridge, Massachusetts, 2006.

COHN, M., (2004) *User Stories Applied: For Agile Software Development*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

CONSTANTINE, L. L., and Lockwood, L. A. D. (1999) *Software for Use: A Practical Guide to Models and Methods of Usage- Centered Design*. Addison-Wesley, Reading, MA,.

_____, (2000) Rapid abstract prototyping. Software development. Reprinted in S. Ambler, and L. Constantine (eds.), *The unified process elaboration phase: Best practices in implementing the UP*. CMP, Lawrence, KS, 2000.

_____, (2002) Process agility and software usability: Toward lightweight usage-centered design. *Information Age*, 8.

_____, (2004) Beyond user-centered design and user experience: Designing for user performance. *Cutter IT Journal*, 17, 2.

EDMONDS, B. (1999). Syntactic Measures of Complexity. University of Manchester.

ERICKSON, T. D. (1990) Working with Interface Metaphors. Em B. K. Laurel (ed) *The Art of Human-Computer Interface Design*. Reading: Addison-Wesley Publishing Company.

FIALHO & SANTOS. (1995) Manual de análise ergonômica no trabalho. Curitiba: Gênese.

FLAVELL, J. H. (2001) A Psicologia do Desenvolvimento de Jean Piaget. São Paulo: Pioneira Thomson Learning.

GOMES, A. S.. (2005) Avaliação da aprendizagem com software educativo no projeto interativo. In: 5º Congresso Internacional de Ergonomia e Usabilidade de Interface Humano Máquina, 2005, Rio de Janeiro. Anais do 5o Congresso Internacional de Ergonomia e Usabilidade de Interface Humano Máquina. Rio de Janeiro, 2005.

_____.; Wanderley, E. G, (2003). Elicitando requisitos de software educativo. In: WIE 2003 Workshop Brasileiro de Informática Educativa, 2003, Convergências Tecnológicas, Redesenhando as Fronteiras da Ciência e da Educação: Anais. Campinas: SBC, 2003.

HAIR, J. F.; ANDERSON, R. E.; TATHAM, R. L.; BLACK, W. C. (1998) Multivariate data analysis. 4 ed. New Jersey: Prentice Hall.

HANSMAN, R. J., De Albuquerque, E. A., Trabasso, L. G. & Li, L. (2008 a). Experimental Setup for Air Traffic Control Cognitive Complexity Analysis. *X SIGE - Simpósio de Guerra Eletrônica*. São José dos Campos, Brazil.

_____, Histon, Jonathan M.(2008 b) Mitigating complexity in Air Traffic Control : the role of structure-based abstractions - Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics.

_____, Li, L.(2009) Experimental Studies of Cognitively Based Air Traffic Control Complexity Metrics for Future Operational Concepts - Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics.

HARTMANN, D. (2006). “Feature Driven Development: Still relevant?” Disponível em: http://www.infoq.com/news/Another_look_at_FDD Acessado em nov. 2009.

HOLLNAGEL, E. (1993), Human Reliability Analysis: Context and Control (London: Academic Press).

_____, __, (2000), ‘Analysis and Prediction of Failures in Complex Systems: Models and Methods’, in Elzer, P.F., *et al.* (eds.) (2000), Human Error and System Design and Management (Verlag: Springer).

_____, __, Pieri, F. and Rigaud, E. (2008). Proceedings Of The Third Resilience Engineering Symposium, Antibes-Juan-les-Pins, France.

HOLZINGER, A., and Slany, W. XP + UE @ XU Praktische Erfahrungen mit eXtreme Usability. *Informatik Spektrum*, 29, 1 (2006).

LAPOLLI, F; Motta, C.L.R. ; Gomes, J. O. ; Oliveira, Carlo Emmanoel Tolla de. Metacognição como Processo de Aprendizagem Visando a Construção de Respostas Adaptativas em Profissionais que Atuam em Sistema Complexos (2009 a). In: 2009 Brazilian Symposium on Aerospace Engineering and Applications / 3rd CTA-DLR Workshop on Data Analysis and Flight Control, São José dos Campos. Proceedings of the 2009 Brazilian Symposium on Aerospace Engineering and Applications / 3rd CTA-DLR Workshop on Data Analysis and Flight Control. São José do campos. v. 1. p. 521-530.

_____, Cruz C., Motta, C. L.R., Tolla, C, E., (2009 b) Modelo de Objetos de Aprendizagem Baseado em Metodologias Ágeis – Anais do XX Simpósio Brasileiro de Informática na Educação.

_____. Pereira, L.D.; FerrentiniFabio ; Motta, C.L.R. ; Oliveira, Carlo Emmanoel Tolla de. (2010 a) Ateliê de Objetos de Aprendizagem: Uma Abordagem para o Ensino de Computação em Cursos Técnicos. In: XXI Simpósio Brasileiro de Informática na Educação, 2010, João Pessoa, PB. Anais do XXI Simpósio Brasileiro de Informática na Educação. Porto Alegre, RS : Sociedade Brasileira de Cpmputação.

_____.; Cruz, C. M. ; Motta, C.L.R. ; Oliveira, C.E.T. (2010 b) Modulo de Desenvolvimento de Objetivos de Aprendizagem em Metodologias Ágeis e Scaffoldings. Revista Brasileira de Informação na Educação, v. 18, p. 17-32, 2010.

_____.; Motta, C.L.R. ; Gomes, J. O. ; Oliveira, C.E.T. (2010 c) Modelo Ergonômico de Desenvolvimento de Interface de Software para Treinamento em Sistemas Complexos. In: III Congresso Latino-Americano de Ergonomia, ULAERGO 2010 / XVI Congresso Brasileiro de Ergonomia, 2010. Anais do III Congresso Latino-Americano de Ergonomia, LAERGO 2010 / XVI Congresso Brasileiro de Ergonomia, 2010.

_____.; Cruz, C. M. ; Motta, C.L.R. ; Gomes, J. O. ; Oliveira, C.E.T. (2010 d) Modelo Ergonômico de Levantamento de Requisitos de Software para Treinamento em Sistemas Complexos. In: III Congresso Latino-Americano de Ergonomia, ULAERGO 2010 / XVI Congresso Brasileiro de Ergonomia, 2010. Anais do III Congresso Latino-Americano de Ergonomia, ULAERGO 2010 / XVI Congresso Brasileiro de Ergonomia, 2010.

_____.; Motta, C.L.R. ; Oliveira, C.E.T. (2010 e) Modelo de Desenvolvimento de Objetos de Aprendizagem Baseado em Metodologias Ágeis e Scaffoldings. In: LACLO - 5º Congresso Latino Americano de Objetos de Aprendizagem, 2010, São Paulo, SP. Anais do LACLO - 5º Congresso Latino Americano de Objetos de Aprendizagem, 2010.

_____.; Pereira, L.D. ;Sampaio, F.F. ; Motta, C.L.R. ; Oliveira, C.E.T.. (2010 f) Ateliê de Objetos de Aprendizagem Uma metodologia para o Ensino de Computação em Cursos Técnicos. In: WEI - Workshop de Educação em Informática, 2010, Belo Horizonte. Anais do WEI - Workshop de Educação em Informática, 2010. p. 907-916.

LONGMIRE (2001), W. A Primer On Learning Objects. American Society for Training & Development. Virginia. USA. 2001.

LOWGREN, J., and Stolterman, E. (2004) *Thoughtful Interaction Design: A Design Perspective on Information Technology*. MIT Press, Cambridge, MA.

MOREIRA, M. A. (1999). Teorias de Aprendizagem. São Paulo: EPU.

MORIN, Edgar. (2005). Introdução ao pensamento complexo. Editora Sulina.

NICK, E.; CABRAL, A. (2006). Dicionário Técnico de Psicologia. 14ed, São Paulo: Cultrix.

NORMAN, D.; (1985) Hutchins, E. L., and J. D. Hollan, Direct manipulation interfaces. Human-Comput. Interact.

_____.; (1986) Cognitive Engineering. In D. Norman & S. Draper (eds.) User Centered System Design. Hillsdale, NJ. Lawrence Erlbaum. pp.31-61.

_____.; (1988) *Psychology of Everyday Things*. BasicBooks. HarperCollins Publishers.

_____.; (1991) “Cognitive Artifacts”. In Carroll (ed.) *Designing Interaction: Psychology at the Human-Computer Interface*. pp.17-38.

_____.; (1993) *Things that make us smart: defending human attributes in the age of the machine*. Reading, MA: Addison-Wesley.

_____.; (2005) Human-centered design considered harmful. jnd.org. Disponível em http://jnd.org/dn.mss/human-centered_design_considered_harmful.html Acesso em 16 de outubro de 2009.

_____.; (2006) Logic Versus Usage: The Case for Activity-Centered Design. jnd.org. Disponível em http://jnd.org/dn.mss/logic_versus_usage_the_case_for_activity-centered_design.html Acessado em out. 2009.

_____.; (2007) Three Challenges for Design. jnd.org. Disponível em http://jnd.org/dn.mss/three_challenges_for_design.html Acessado em out. 2009.

NORTH D., (2007 a) *Introducing: Behaviour-driven development*, Disponível em: <http://dannorth.net/introducing-bdd/> Acessado em fev. 2009.

_____. (2007 b) *Introducing: Behaviour-driven development*, Disponível em: <http://dannorth.net/whats-in-a-story/> Acessado em fev. 2009.

PAVARD, B e DUGDALE, J. A (2005) Introduction to Complexity in Social Science. Disponível em: www.irit.fr/COSI/training/complexity-tutorial.htm. Acessado em ago. 2009.

PERROW, C. (1999). Normal accidents: living with high-risk technologies. New Jersey Princeton University Press.

_____. (1986) Complex Organizations: A Critical Essay. New Jersey Princeton University Press.

PIAGET, Jean. (1977) O desenvolvimento do Pensamento: Equilíbrio da Estruturas Cognitivas. Lisboa: Publicações Dom Quixote.

_____. (1987) Psicogênese dos Conhecimentos e a sua Significação Epistemológica. Teorias da Linguagem, Teorias da Aprendizagem. Lisboa: Edições.

_____. (1905) Abstração Reflexionante: Relações Lógico Aritméticas e Ordem das Relações Espaciais. Porto Alegre: Artes Médicas.

PRESSMAN, R. S. (2001). *Software Engineering: A Practitioner's Approach – Fifth Edition*. Editora McGraw-Hill.

PREECE, Jennifer; ROGERS, Yvone; SHARP, Helen. (2005) *Design de Interação: Além da Interação Homem-Computador*. Editora Bookman, São Paulo: 2005.

REZENDE, Denis Alcies; ABREU, Aline França de. (2008) *Tecnologia da Informação Aplicada a Sistemas de Informação Empresariais*. 3ª Edição, Editora Atlas.

QUINTANA, C.; J. B. Reiser, E. A Davis, J. Krajcik, R. Golan, E. Kyza, D. Edelson, and E. Soloway. (2000) *Structured Definition for Learner-Centered Design*. In B. Fishman and S. O'Connor-Divelbiss (Eds.), *Fourth International Conference of the Learning Sciences* (pp. 256-263). Mahwah, NJ: Erlbaum.

_____.; J. B. Reiser, E. A Davis, J. Krajcik, R. Golan, E. Kyza, D. Edelson, and E. Soloway. (2002) *Evolving a Scaffolding Design Framework for Designing Educational Software*. *Keeping Learning Complex: The Proceedings of the Fifth International Conference of the Learning Sciences*. (ICLS). Mahwah, NJ: Erlbaum

REEVES, T. C. e Hedberg, J. G. (2003) *Interactive learning systems evaluation*. Educational Technology Publications, New Jersey, USA.

REISER, J. B. (2002) *Why Scaffolding Should Sometimes Make Tasks More Difficult for Learners* In Stahl, G. (ed.) *Computer support for collaborative learning foundations for a CSCL community*. *Proceedings of CSCL 2002, Boulder, Colorado, USA (January 7-11, 2002)*, 255-264. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.

RASMUSSEN, J. (1987) *Cognitive Control and human error mechanisms*. Em: Rasmussen, J., Duncan, K. e Leplat, J. *New technology and human error*. New York: John Wiley & Sons.

_____. (1994). *Cognitive Systems Engineering*. New York: John Wiley & Sons.1994.

SEDIGHIAN K.; M. Westrom. (1997) Direct Object Manipulation vs. Direct Concept Manipulation: Effect of Interface Style on Reflection and Domain Learning. In: Thimbleby, Harold, O’Conaill, Brid, Thomas, Peter J. (ed.): Proceedings of the Twelfth Conference of the British Computer Society Human Computer Interaction Specialist Group - People and Computers XII. August, 1997, Bristol, England, UK. p.337-357.

_____.; _____.; M. Klawe, and (2001) Role of interface manipulation style and scaffolding on cognition and concept learning in learnware. ACM Transactions on Computer-Human Interaction.

SEMINÉRIO, Franco Lo Presti. (1984) Infra-estrutura da cognição (I): Linguagens e Canais Morfogenéticos. Rio de Janeiro: FGV, ISOP, nº 4.

_____. (1985) Infra-estrutura da Cognição II. Linguagens e Canais Morfogenéticos. Rio de Janeiro: FGV, ISOP, nº 8.

VYGOTSKY, L. (1991). A Formação Social da Mente. São Paulo: Martins Fontes.

WILENSKY, U. Resnick, M., & (1995). New Thinking for New Sciences: Constructionist Approaches for Exploring Complexity. *American Educational Research Association, San Francisco, CA.*

_____. (1998). Diving into complexity: Developing probabilistic decentralized thinking through role-playing activities. *Journal of Learning Sciences.*

_____. (1999). Thinking in Levels: A dynamic systems perspective to making sense of the world. *Journal of Science Education and Technology*, 8, 3-18.

_____, Goldstone, R. L., (2008). Promoting transfer by grounding complex systems principles. *The Journal of the Learning Sciences*, 17, 465-516.

WILEY, D. A., (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *The Instructional Use of Learning Objects*:
Versão online disponível em: <http://reusability.org/read/chapters/wiley.doc>, último acesso em março de 2009.

WOODS, D. *et al.* (1994). *State-of-the-art-report: behind human error: cognitive systems, computers, and hindsight*. Columbus: CSERIAC.

_____, Hollnagel, E. and Leveson, N. (2006 a) “Resilience Engineering: Concepts And Precepts”, Aldershot, UK: Ashgate.

_____, Hollnagel, E. and Leveson, N. (2006 b) “Joint Cognitive Systems: Patterns in Cognitive systems Engineering. Aldershot. Boca Raton, FL: Taylor & Francis.



**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

CCMN - Bloco C - Cidade Universitária - Ilha do Fundão
Rio de Janeiro - RJ CEP: 21941-916
www.ppgi.ufrj.br