Uma Abordagem de Arquitetura Estilizada para Software Educacional

José Roberto Vasconcelos [1] , Ivan Luiz Marques Ricarte[2]
Gécen Dacome de Marchi[1], Rafael Alessandro Gatto[1]

[1] *Departamento de Informática, Universidade Estadual de Maringá*
Maringá – PR – Brazil

[2] *Departamento de Engenharia de Computação e Automação, Universidade Estadual de Campinas*
Campinas – SP – Brazil

E-mail: [1] {jrvasco, gdmarchi, ragatto}@din.uem.br, [2] ricarte@dca.fee.unicamp.br

III – Workshop de Tecnología Informática Aplicada en Educación

Resumo – O desenvolvimento baseado em componentes vem emergindo como um ponto de relevância na engenharia de software. Considerando este fato, este trabalho propôs, a partir do modelo de referência IEEE/LTSA, uma arquitetura para Software Educacional baseada em componente e fundamentada na comunicação entre eles através do estilo arquitetural de Tubos e Filtros (*pipe-and-filter*). Adota-se o conceito de reutilização promovendo o aumento da qualidade e a diminuição do esforço de desenvolvimento através criação de componentes adaptáveis e interfaces críticas de interoperabilidade.

Palavras Chaves – Arquitetura de software, componentes de software, software educacional, estilo arquitetural, reusabilidade.

I. INTRODUÇÃO

O interesse no uso de computadores em aprendizagem e na educação a distância tem motivado o desenvolvimento de ambientes educacionais, mas para que isto torne um caminho eficiente no processo de aprendizagem faz-se necessário a utilização de mecanismos que auxiliem no processo de elaboração de aplicações direcionadas ao domínio educacional. Com a rápida evolução tecnológica, ambientes de desenvolvimento "informais" ficam logo obsoletos, portanto é preciso ter meios de definir e descrever esses ambientes de forma mais estáveis e com nível de abstração adequado. Uma abordagem, que procura minimizar esses problemas, é a utilização de arquiteturas de software para um domínio específico.

A arquitetura de um sistema reflete o conjunto de decisões que devem ser tomadas, no projeto de um sistema computacional, determinando a organização e estrutura geral do sistema. Uma definição clássica de arquitetura de software diz o que é o sistema em termos de componentes computacionais e os relacionamentos entre eles [1].

A importância de arquitetura de software para aplicações educacionais foi reconhecida pelo *Institute of Electrical and Electronics Engineers* (IEEE), que propôs como parte de seus esforços de padronização, a definição de um modelo de referência *Learning Technology Systems Architecture* (LTSA). A proposta deste trabalho relaciona requisitos necessários, tanto os funcionais como os de qualidade, para

compor um relacionamento entre componentes com um nível de abstração menos elevado que o modelo LTSA.

Em virtude de seu alto grau de abstração, no modelo LTSA, fica difícil traduzir uma especificação desejada de um ambiente para uma especificação de desenvolvimento. Neste trabalho, propõe-se uma especialização do LTSA focada para um desenvolvimento baseado em componentes. Nesta especialização adotou-se o estilo de Tubos e Filtros (*pipe-and-filter*), que fundamenta-se na comunicação entre componentes. Isto possibilitou a identificação de funcionalidades e serviços dos componentes de maneira mais clara e precisa, que permite o desenvolvimento e integração de componentes mais facilmente.

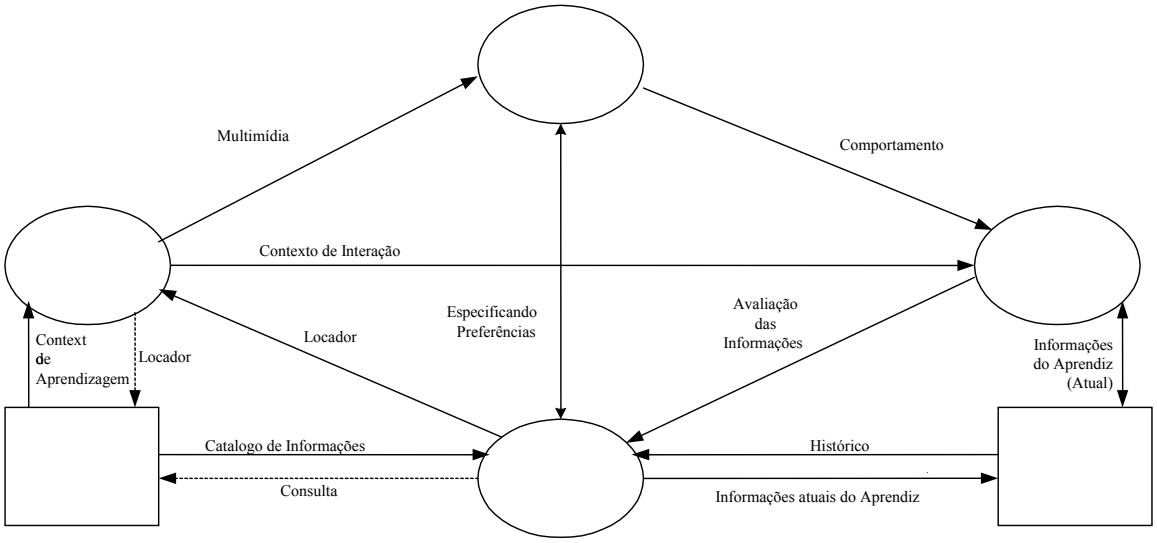
II. O MODELO DE REFERÊNCIA IEEE/LTSA

O modelo LTSA especifica uma arquitetura de alto nível para tecnologia de informação de aprendizagem, instrução e sistemas de treinamento. Este modelo de referência abrange uma grande área de sistemas, que envolve tecnologia de aprendizagem, educação, treinamento baseado em computador, instrução apoiada por computador, tutoração inteligente, entre outros. O modelo é pedagógica e culturalmente neutro, independente de plataforma e fornece uma estrutura para compreender sistemas futuros e já existentes, promovendo a interoperabilidade e a portabilidade, identificando interfaces críticas do sistema [2].

Em geral, a finalidade de desenvolver arquiteturas de sistema é criar descrições de alto nível para compreender determinados tipos dos sistemas, seus subsistemas e suas interações com sistemas relacionados. Uma arquitetura não é um plano para projetar um único sistema, mas uma estrutura para projetar uma escala dos sistemas sobre o tempo, para a análise, comunicação e a comparação destes sistemas. Com isto, a arquitetura provê componentes que podem ser compartilhados entre diferentes sistemas, no nível correto de generalização. Uma arquitetura promove o projeto e a execução dos componentes e dos subsistemas que são reusáveis, de custo efetivo e adaptáveis, com a identificação de serviços e com interfaces de alto nível de interoperabilidade [2].

O modelo LTSA descreve processos, repositórios e fluxos de informção. Processos são descritos em termos de limites, entradas, procedimentos (funcionalidades) e saídas. Repositórios são descritos pelo tipo de informação armazenado, busca, recuperação e métodos de atualização. Os fluxos são descritos em termos de conectividade (um ou dois sentidos, conexões estáticas e dinâmicas) e o tipo de informação que flui [2].

Os componentes da arquitetura LTSA podem ser observados na Figura 1. Nesta representação, elipses correspondem a processos, retângulos a repositórios e seta a fluxos.



O processo Entidade Aprendiz é uma abstração de um aprendiz humano, que pode representar um ou vários alunos, aprendendo individualmente ou em colaboração, sendo possível a adoção de regras diferentes através do objetivo de aprendizagem.

O processo Avaliação cria informação de desempenho que é armazenada nos repositórios, isto é realizado através da integração com o contexto para disponibilizar um ambiente de comportamento para a entidade aprendiz, de modo a determinar uma avaliação apropriada.

O repositório de Registro de Aprendiz armazena informações de desempenho e preferências do aprendiz. Que pode vir dos processos de avaliação e de técnico, também pode permanecer informações sobre o passado, presente e futuro do processo de aprendizagem da Entidade Aprendiz.

O processo Técnico pode solicitar recursos, negociar preferências de aprendizagem, como: estilos, estratégias, entre outras. Através do recebimento de informações o Técnico pode fazer escolhas futuras utilizando-se de experiências de aprendizagem. Este processo também mantém atualizadas informações sobre os Registros de Aprendiz e determina como será apresentado o conteúdo a Entidade Aprendiz. O termo Técnico pode se referenciar ao professor, mentor, instrutor, tutor ou outra entidade relacionada com aprendizado, sendo ela computacional ou não.

O repositório de Recursos de Aprendizagem pode armazenar reprensetações de conhecimento, apresentações, tutoriais, ferramentas e outros materiais de aprendizagem. Este processo será útil para a disponibilidade do conteúdo para entrega a Entidade Aprendiz.

O processo Entregador pode transformar a informação obtida, via conteúdo de aprendizagem, em uma apresentação que pode ser obtida, via conteúdo de aprendizagem, em uma apresentação que pode ser transmitida a entidade aprendiz via mídia (ou multimídia). A apresentação pode ser estática, interativa, colaborativa, envolvendo experimentos e descobertas, entre outras [2].

III. ARQUITETURA PROPOSTA

Esta seção apresenta uma proposta de desenvolvimento de uma arquitetura para software educacional baseada conceitos de reutilização e componentes de software. No desenvolvimento desta arquitetura,

primeiramente foi necessário encontrar um conjunto específico de informações que compõem a tecnologia de aprendizagem e, então, especificar o conjunto de componentes que a arquitetura suporta, as exigências funcionais que emergem e a especificação da forma de conexão utilizada para prover a interação entre componentes da arquitetura.

O objetivo é apresentar uma arquitetura que facilite a modelagem de aplicações de âmbito educacional. Este trabalho concentra-se na modelagem abstrata dos subcomponentes que serão identificados a partir do modelo LTSA. A definição de interfaces é realizada de forma a identificar algumas das principais funcionalidades e serviços destes subcomponentes. Os componentes da arquitetura proposta foram especificados com base nos conceitos a seguir.

O que torna algum artefato de software um componente não é uma aplicação e nem uma tecnologia de implementação específica; assim, qualquer dispositivo de software pode ser considerado um componente, desde que possua uma interface definida. Esta interface deve ser uma coleção de pontos de acesso a serviços, cada um com uma semântica estabelecida [3].

Além da interface provida, isto é o conjunto de métodos de componentes que podem ser invocados, uma descrição de interface do componente também deve estabelecer a interface requerida, isto é, os métodos que o componente invoca [4].

Também é de grande importância, que a descrição da interface de um componente seja clara, completa e concisa, caso contrário pode provocar uma má interpretação do componente e assim este termina por não servir a necessidade. Métodos formais e semiformais podem auxiliar para especificar as interfaces com o objetivo de que estas não venham a estar ambíguas [5].

Para finalizar, a especificação das interfaces dos componentes (serviços providos e requeridos) e dos conectores (comunicação entre componentes) descrevem também o comportamento do componente [6].

É importante ressaltar, que o primeiro passo na elaboração de um componente, é projetar as interfaces dos componentes que irão determinar quais são funcionalidades necessárias que deverão ser implementadas no componente e de que forma essas se comunicarão [7].

A partir destes conceitos básicos e também do modelo de referência IEEE/LTSA é que a arquitetura proposta se fundamenta. A seguir é mostrada a especificação dos componentes da arquitetura proposta.

Para a Entidade Aprendiz não se encontrou a necessidade de uma divisão em subcomponentes, além dos já existentes. Esta entidade representa todas as entidades que interagirão com qualquer aplicação desenvolvida a partir da arquitetura proposta. A definição para esta entidade a trata como a abstração de uma pessoa como um aprendiz, ou melhor, esta entidade representa um único aprendiz, um grupo de aprendizes que aprendem colaborativamente ou individualmente.

O componente Técnico realizará a filtragem dos dados para os componentes Administrador, Tutor e Coordenador ou para um outro componente que pode ser incorporado na arquitetura (um componente abstrato), ou melhor, realizará a distribuição de tarefas entre os componentes, conforme pode ser verificado na Figura 2.

O objetivo principal no refinamento (especialização) foi a possibilidade de agrupar funcionalidades e serviços do componente Técnico em subcomponentes (componentes) que tivessem funcionalidades e serviços bem determinadas, caracterizando as partes específicas de um Técnico. Com isto espera-se que a incorporação destes novos elementos torne mais fácil a identificação de componentes reusáveis, ou até mesmo a implementação destes.

O subcomponente Administrador é responsável pelo armanezamento e manutenção dos recursos de aprendizagem, pela validação e pelo monitoramento das atividades do sistema (componentes) em uso.

O conjunto de ações de responsabilidade do Coordenador se caracteriza pelo planejamento,

desenvolvimento e execução do sistema para o processo de aprendizagem. Nas ações que atuam sobre a orientação do aprendiz podem ser associadas ao subcomponente Tutor, que agirá como um facilitador (mediador) e negociador de preferências do aprendiz, entre outros serviços.
Cabe ressaltar aqui que a especificação destes subcomponenes não tornou o modelo direcionado, pois os mesmos continuam sendo descritos e especificados de maneira abstrata.

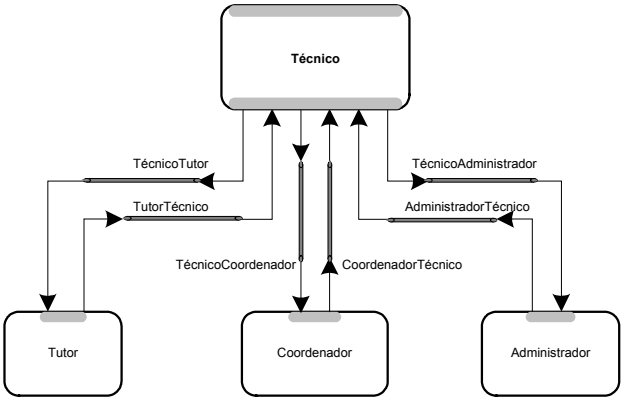


Figura 2 – Componente Técnico.

Analogamente ao componente Técnico, o componente Avaliação realizará a filtragem de dados para os componentes, Avaliação do Sistema, Avaliação de Navegabilidade e Avaliação Pedagógica, ou melhor, realizará a distribuição de tarefas entre os componentes, como pode ser observado na Figura 3.
Para verificar os subcomponentes foram analisados os aspectos de possibilidade de adaptação do sistema, onde a Avaliação do Sistema permitiria que o sistema se auto-reorganizasse. Como é comum em um sistema de aprendizagem a verificação do comportamento através das atitudes (passos) que o aprendiz realiza, foi apresentada a Avaliação de Navegabilidade, que consiste em acompanhar os caminhos do aprendiz. E finalizando foi acrescentada a Avaliação Pedagógica que será composta juntamente com o material, verificando as estratégias de ensino que se pretende enfocar na aplicação educacional.

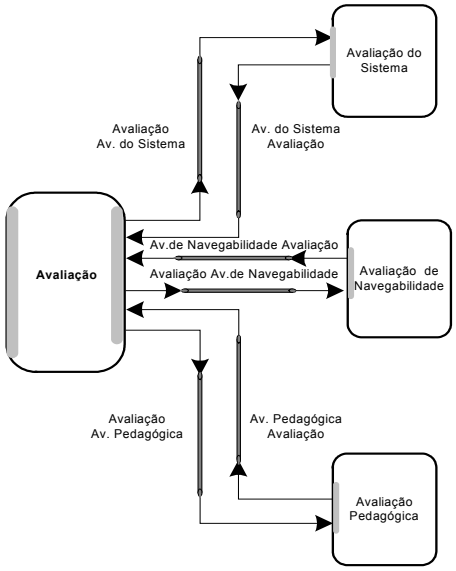


Figura 3- Componente Avaliação.

O componente Entregador realizará a filtragem de dados para os componentes, Entregador de Áudio, Entregador de Vídeo e Entregador de Texto e realizará a distribuição de tarefas para os componentes, conforme observa-se na Figura 4.

A funcionalidade e serviços dos componentes identificados aqui possuem as mesmas características, sendo diferenciados na mídia em questão. Estes serviços são interpretar, gerar e enviar apresentações que irão compor o material a ser entregue à Entidade Aprendiz. Geralmente é comum ter um componente específico para apresentar uma mídia, o qual é ligado (*plugin*) a outros componentes.

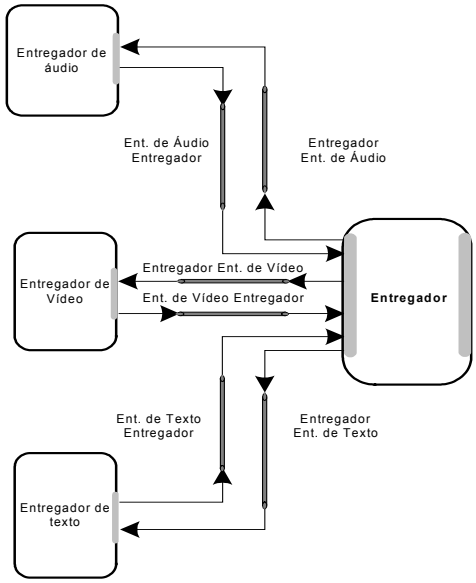


Figura 4 – Componente Entregador.

Na arquitetura proposta foi previsto a possibilidade de incorporação de outros componentes (novos ou já existentes) para permitir a reutilização destes. Para isto, no modelo proposto pode-se ter a necessidade que esse componente seja adaptado para se ajustar aos requisitos do sistema ao qual será acoplado. Neste caso utiliza-se a abordagem de colagem (*glueing*) que representa uma forma de interconectar componentes, ou melhor, possibilitar a operação conjunta de componentes originalmente incompatíveis (Figura 5). Esta incompatibilidade pode estar associada à sintaxe das interfaces, heterogeneidade de plataformas, necessidade de extensões ou alterações funcionais. O tratamento dado ao problema é a inclusão de um novo elemento, a cola (*glue*), entre os componentes incompatíveis e o modelo, possibilitando sua operação conjunta. A Figura 5 ilustra a compatibilização de componentes através de colagem.

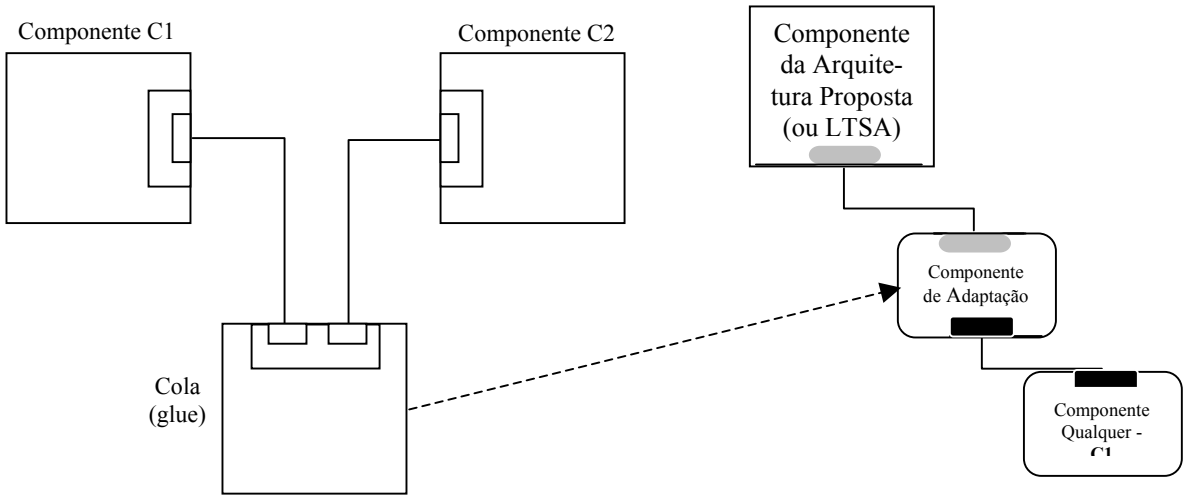


Figura 5 – Adptação de Componente.

Como exemplo, suponha que o Componente C1 será incorporado a arquitetura e foi implementado em C++, e os componentes do sistema em Java. Portanto, tem-se a necessidade de adaptação para solucionar a heterogeneidade entre os componentes, que será feito através do componente cola, o qual funcionará como um conector de componentes.

IV. CONECTORES

Para representar a comunicação entre os componentes da arquitetura utiliza-se o estilo arquitetural Tubos e Filtros *(Pipe and Filter)*. Abaixo será descrita a topologia de comunicação da arquitetura. Considera-se que os componentes descritos anteriormente sejam vistos como filtros, para melhor representar o relacionamento entre componentes.

Este estilo considera a existência de uma rede pela qual fluem dados de uma extremidade a outra. O fluxo de dados se dá através de tubos, que interligam entradas e saídas de filtros, e os dados sofrem transformações quando processados nos filtros, que são artefatos de software com entradas por onde recebem fluxo de dados e saídas por onde disponibilizam fluxos de dados resultantes de sua atuação

sobre os fluxos de entrada. A forma de interação característica deste estilo arquitetural é o fluxo de dados unidirecional [1, 8].
A fim de definir o comportamento de um filtro, devem-se saber os dados de entrada e de saída das portas e o tipo de dados que podem ser passados ao longo de cada portas. Esta última informação pode ser representada através do subconjunto de dados das portas, considerando o universo de dados possíveis de seu alfabeto [9].
O papel do filtro na etapa computacional é transformar alguns dados de entrada em dados de saída. A ordem dos dados é preservada, assim os dados de entrada são consumidos na ordem que chegaram e os dados da saída são mantidos na mesma ordem que são produzidos [9]. O resultado da etapa computacional para o filtro é a remoção de alguns dados das portas de entrada, ou seja, uma transformação daqueles dados, isto depende do estado interno atual do filtro, então se acrescentam os dados transformados às portas de saída [10].
Tubos conectam as portas de dados dos filtros; assim, um tubo representa uma transmissão de dados de um filtro a outro. Cada tubo tem um *source* (origem) e um *sink* (destino) distintos para receber e emitir dados.
O protocolo de um tubo é definido através de sua política de transmissão. Em algum momento, o tubo tem alguns dados na porta *source* se movendo para a porta *sink* [10].
A Figura 6 apresenta uma parte da arquitetura modificada do modelo LTSA, sendo os demais componentes similares a esta abordagem. Neste estilo, os componentes Avaliação e Técnico são filtros que transformam o fluxo de dados e fornecem uma interface de entrada e saída. Esta interface representa o protocolo de comunicação entre os componentes. Na Figura 6, DadoSaída são resultados gerados pelo componente Avaliação, que serão utilizados pelo DadoEntrada do Componente Técnico; neste caso, ambos devem ser compatíveis.

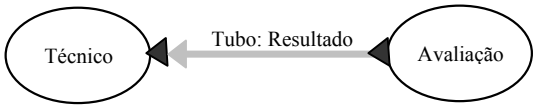


Figura 6 – Estilo Tubos e Filtros.

O conector Resultado é um tubo que descreve o relacionamento binário entre os dois filtros e os protocolos de transferência de dados. Um tubo possui duas interfaces: *source* (origem) que somente poderá ser conectada com a interface de saída do filtro e *sink* (destino) que se conectará somente com a interface de entrada do filtro. Esta representação garante a compatibilidade de dados que serão transmitidos entre os componentes.
Outros problemas que podem ocorrer são a incompatibilidade de interfaces com relação à linguagem de implementação e plataformas diferentes. Dessa maneira, a melhor representação é ilustrada na Figura 7.

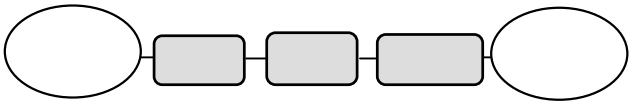


Figura 7 – Adaptação de Interfaces.

O conector Resultado está representado através dos componentes *sink* (destino), *source* (origem) e tubo. Nesta abordagem as interfaces de saída e entrada incompatíveis podem se comunicar através da implementação de componentes de adaptação. O componente *source* fará a adaptação do filtro Avaliação para o Tubo e o *sink* a adaptação do Tubo para o filtro Técnico, o que tornará possível um protocolo de comunicação entre os componentes, independentemente de plataforma e linguagem utilizadas. Aqui, o componente de adaptação fundamenta-se na abordagem de colagem com Tubos e Filtros, como exemplo de implementação, poderia utilizar-se da tecnologia CORBA.

V. UTILIZAÇÃO DO MODELO

Nesta seção será ilustrado o desenvolvimento de um sistema de aprendizagem hipotético, baseado na *Web*. O diagrama ilustra os mapeamentos individuais dos componentes para o modelo LTSA, conforme Figura 8. A Entidade Aprendiz é identificada como um ser humano, os fluxos Multimídia e comportamento são mapeados para uma interface do ser humano (sistema operacional). Os componentes Entregador, Avaliação, Técnico, Contexto de Interação, Locador e Avaliação das Informações são mapeados para uma ferramenta de apresentação (Navegador). Registros de Aprendiz são mapeados para Registros do Estudante (Banco de Dados), os Recursos de Aprendizagem foram identificados como Ferramentas do Curso e Banco de Dados (Servidor *Web*).

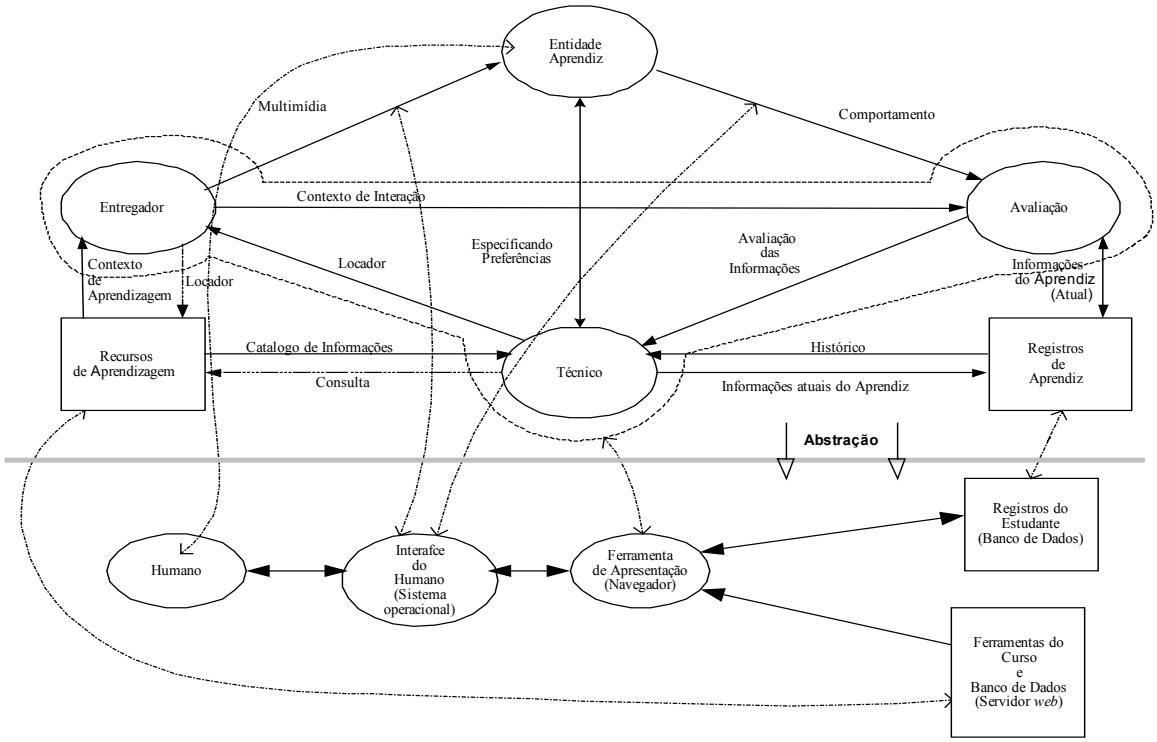


Figura 8 – Sistema Hipotético Mapeado para o Modelo IEEE/LTSA.

Para tornar menos abstrato o mapeamento realizado na Figura 8, os Registros do Estudante, Ferramentas do Curso e Banco de Dados estão representados na Figura 9 de uma maneira que facilita o entendimento e melhora a visão da Arquitetura. Outro ponto relevante é que foi introduzido um novo componente abstrato na arquitetura, o Tutorial, que pode representar diferentes funcionalidades e serviços a serem incorporados a componentes já existentes nesse sistema.

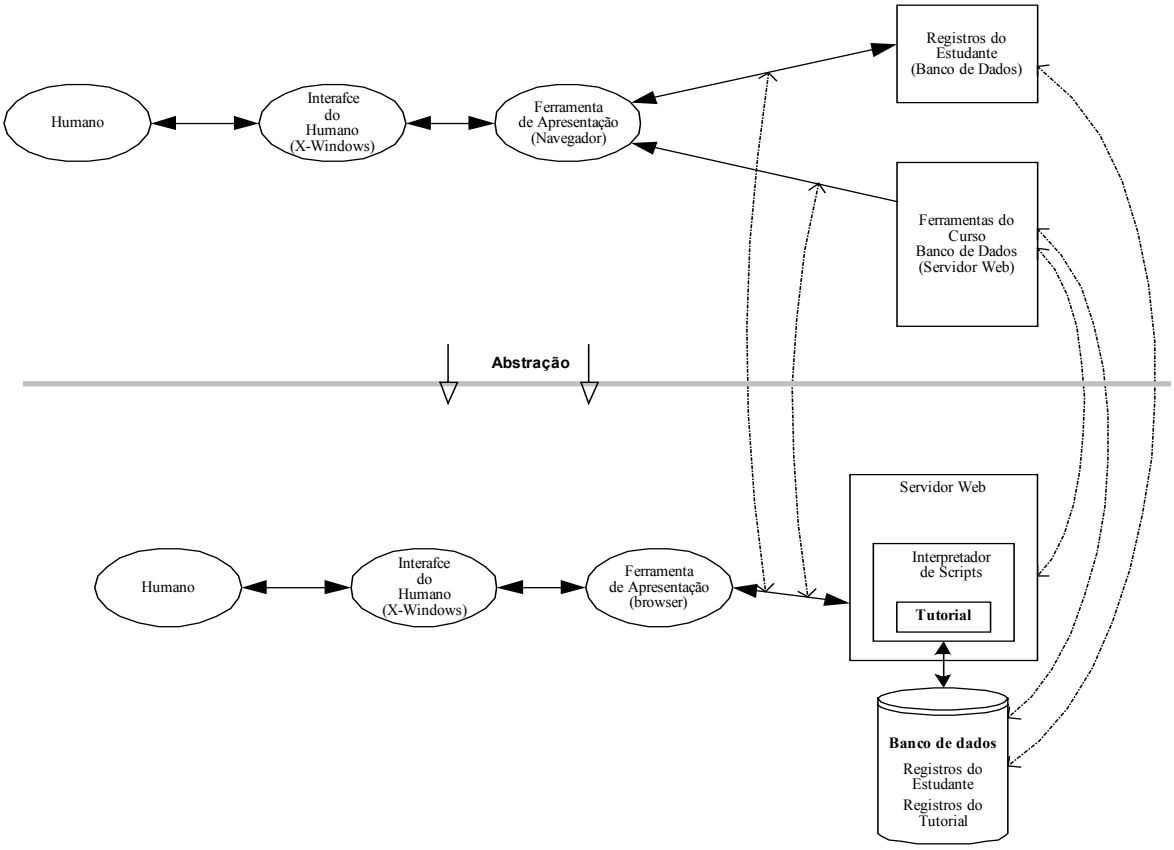


Figura 9 – Refinamento dos Repositórios de um Sistema Hipotético.

No estudo de caso, o Tutorial representou uma aplicação para validação de usuários e apresentação de lições, portanto necessitando da implementação de vários componentes, os quais precisaram interagir com outros componentes apresentado na Figura 9.

Na arquitetura proposta o Tutorial tem vários componentes associados, um deles é o Administrador para validação de usuários, que contem suas características pertinentes. O Técnico, da arquitetura proposta, é um filtro que "delega" as tarefas ao Administrador através de suas interfaces (provida e requerida).

Como exemplo, o componente Entregador foi refinado em outros três componentes na arquitetura proposta, no Tutorial esses componentes são responsáveis por apresentar as lições. Neste caso, pode-se identificar o Entregador de Áudio e de Vídeo como *plugins* que se comunicam com o Navegador através de suas interfaces definidas pelo fabricante, conforme a Figura 10.

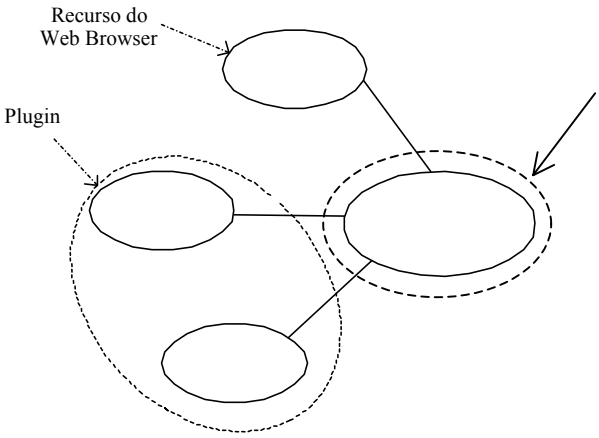


Figura 10 - Mapeamento da Apresentação de Lições para a Arquitetura Proposta.

No Tutorial foi utilizado o *plugin* Flash que possui uma interface compatível com o navegador Internet Explorer 6.0, a compatibilidade de interfaces possibilita que o navegador utilize os serviços do *plugin* para apresentar a mídia de áudio e vídeo. A mesma idéia do componente Técnico é utilizada no Entregador que servirá de filtro para seus componentes (subcomponentes), como é observado na Figura 10.

O desenvolvimento desta aplicação serviu para verificar a concepção de sistemas partindo-se de um nível mais alto de abstração para uma abstração mais próxima de componentes existentes (ou implementáveis). Para isto, foi mostrado o processo de desenvolvimento (*top-down*) de uma aplicação "simples", porém possível de associação de seus componentes com o modelo LTSA e o refinamento para a arquitetura proposta, a qual possibilita a comunicação e interoperabilidade entre componentes.

VI. CONCLUSÃO

O estudo realizado dos conceitos e definições de arquitetura, e componentes de software possibilitou a compreensão do modelo de referência IEEE/LTSA. Adicionalmente a essa tarefa, realizou-se a identificação e especificação de funcionalidades e serviços dos componentes que formam a base da arquitetura proposta. Também foi proposto um modelo representado através da utilização do estilo arquitetural de Tubos e Filtros, que conforme suas caracteristicas demonstraram a forma como deverá ser realizada a comunicação entre filtros (componentes).

O resultado deste trabalho vem descrever de forma menos abstrata os componentes do modelo LTSA, sem descaracteriza-lo, pois os componentes mais específicos da arquitetura proposta continuam tendo as mesmas funcionalidades e serviços do modelo de referência.

Os componentes propostos foram baseados em artefatos de software presentes na maioria dos sistemas educacionais existentes e também no modelo tradicional de aprendizagem presencial. O intuito de verificar fatores e serviços existentes no modelo tradicional é tornar possível o uso de tecnologias para auxiliarem no aprendizado.

Na arquitetura proposta, o estilo de Tubos e Filtros veio possibilitar a apresentação de uma topologia de conexão entre os componentes dessa arquitetura, facilitando assim o entendimento de como o fluxo de dados "trafega" por entre os componentes.

Através da utilização da arquitetura proposta, no estudo de caso, pode-se observar que em geral o desenvolvimento de aplicações, com domínio específico (educacional), tem uma compreensão melhor com a abordagem baseada em componentes, principalmente com a visão mais geral da aplicação para um refinamento mais detalhado desta visão, porém ainda abstrato e com uma aproximação melhor de projeto e implementação da aplicação.
A arquitetura proposta minimiza os problemas e dificuldades de interação entre componentes, além de possibilitar a inclusão de novos componentes que alguma aplicação específica necessite, através da adaptação de componentes abstratos. Assim, a arquitetura resultante é mais robusta e eficiente, e com possibilidade de utilizar várias metodologias no processo de desenvolvimento de software.

VII. REFERÊNCIAS

[1] M. Shaw and D. Garlan, ***Software Architecture – perspective on an emerging discipline***. Upper Saddle River: Prentice Hall, 1996.

[2] IEEE/LTSA – Institute of Electrical and Electronics Engineers - Learning Technology Standards Committee, IEEE Computer Society, ***Draft Standard for Learning Technology — Learning Technology Systems Architecture (LTSA)***. IEEE P1484.1/D9, 2001-11-30.

[3] C. Szyperski, ***Sumary of the Second International WorkShop on Component-Oriented Programming,*** In International Workshop On Componed-Oriented Programing (WCOP) 1., 1996, Linz. Proceedings Linz: [s.n.], 1996.

[4] A. Ólafsson and B. Doug, ***On the need for "required interfaces" of components***, In Special Issues in Object-Oriented Programming, Workshop of the ECOOP, 1996.

[5] C. Pfister, ***A Case Study using BlackBox Components***. Disponível por WWW em: http:/www.oberon.ch/docu/case_study/ index.html. (17/08/1998).

[6] D Garlan and R. Allen, ***A Formal Basis for Architectural Connection***, ACM Transactions on Software Engineering and Methodology. v. 6, n. 3, p. 213-249, July 1997.

[7] C. Szyperski, ***Component Software Beyond Object-Oriented Programmin***g. Editora Addison-Wesley & ACM Press, New York, 1998.

[8] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad and M. Stal***, Pattern-Oriented Software Architecture, Volume 1: A System of Patterns A System of Patterns***, Wiley, 1996.

[9] D. Garlan, G. Abowd and R. Allen, ***Formalizing Style to Understand Descriptions of Software Architecture,*** ACM Transactions on Software Engineering and Methodology, pp. 319-364, October 1995**.**

[10] D. Garlan, R. T. Monroe, D. Kompanek and R. Melton, ***Stylized Architecture, Design Patterns, and Objects.*** http://www-2.cs.cmu.edu/afs/cs/project/compose/ftp**,** September 1996**.**