

---

# Elicitando requisitos em projetos de Software Educativo

Alex Sandro Gomes, Eduardo Garcia Wanderley

Centro de Informática – Universidade Federal de Pernambuco (UFPE)

Caixa Postal 7851 – 50.732-970 – Recife – PE – Brazil

{asg, egw}@cin.ufpe.br

**Abstract.** *The article describes technique of educational software requirements elicitation. Authors call the attention for the educational software low pedagogical quality. Part of the problem is generated by the poor requirements identification. A process integrated and systematic of requirements elicitation is shown. An application on the development of software for teaching additives structures will be argued.*

**Resumo.** *O artigo descreve técnicas de elicitação de requisitos para softwares educativos. Autores chamam a atenção para a baixa qualidade pedagógica de softwares educativos. Parte do problema é gerada pela identificação equivocada de requisitos. Apresenta-se um processo integrado e sistemático de elicitação de requisitos de software educativo. Será discutida sua aplicação no desenvolvimento de um software para o ensino de estruturas aditivas*

## 1. Introdução

No início dos anos 90, Carraher (1990) concluía de forma pessimista quanto às expectativas de surgimento, num tempo breve, de softwares educativos de boa qualidade pedagógica. O autor observava, à época, que “a produção de software de qualidade técnica e, mais ainda, de qualidade pedagógica, é mais complexa do que se imaginava há pouco tempo, de tal modo que dificilmente surgirá, de forma rápida e espontânea uma quantidade de software de qualidade.” (ibid., p. 36).

Um dos motivos da dificuldade em produzir software educativo de qualidade parece estar ligado ao fato que no processo de concepção há uma diferença significativa entre as representações que designers, programadores e professores têm acerca dos processos de ensino e aprendizagem (Mandel, 1997).

A maior parte dos sistemas parece ser criada a partir de proposições *ad hoc* de especialistas. O desenvolvimento de aplicações educativas é há muito tempo considerado como sendo uma atividade quase artesanal. Em um artigo que analisa publicações do WIE nos últimos anos, Gomes, Tedesco e Castro-Filho (prelo) mostram que mesmo as soluções educativas geradas em centros de pesquisa tendem a abordar o problema do desenvolvimento de aplicações educativas de forma não sistemática.

Segundo Tchounikine (2002), o problema central da engenharia de software educativo está associado aos elementos que o designer dispõe e não é um problema do tipo: o programador faz o que o educador especifica. A interação entre programadores e educadores é problemática devido às dificuldades em compartilhar conceitos das diferentes áreas.

---

Em muitos projetos de aplicações educativas, os modelos cognitivos de referência sobre aprendizagem e desenvolvimento humano são indicados sob forma de referência geral e denotam, apenas, o tipo de abordagem seguida (Tchounikine, 2002). Essa apresentação geral não orienta nem influencia no design dos sistemas. O autor cita um conjunto de raros sistemas, cuja concepção releva problemas de aprendizagem e desenvolvimento que são descritos nos modelos de referencia.

O objetivo deste artigo é o de apresentar um processo de engenharia de software educativo que permite abordar o desenvolvimento de uma aplicação educativa, enquanto problema de engenharia de software, numa tentativa de eliminar o maior número de decisões *ad hoc*, substituindo-as por especificações bem fundamentadas. Com isso, espera-se criar interfaces educativas de boa qualidade pedagógica.

Este artigo está organizado da seguinte forma. Na seção 2, discutimos o processo de criação de software educativo e demonstramos a relevância de uma elicitação de requisitos inicial bem feita. Na seção 3, apresentamos um processo racional de identificação de requisitos no projeto de tais sistemas. Na seção 4, é apresentado um exemplo projeto no qual utilizou-se esse método. Na seção 5, concluímos acerca do impacto do processo de elicitação na criação do software.

## **2. Engenharia de *Software educativo***

Entendemos software educativo como sendo a classe de interfaces educativas ou conjunto de artefatos criados para funcionarem enquanto mediadores em atividades educativas de formação em áreas distintas do conhecimento. Podem ainda ser visto como artefatos usados autonomamente pelos aprendizes como instrumento à aprendizagem de algo. Interfaces educativas servem como meio à negociação de significado e à construção de conhecimentos específico em contextos específicos. A função de um software educativo é a de promover aprendizagem para o uso, mas também analisar a aprendizagem de conceitos específicos que ocorrem com o uso do software. Portanto, não se trata apenas aprender a FAZER algo com a interface (Rieman, 1996), mas de aprender a lidar com a interface para APRENDER algum conceito. Em outras palavras, as interfaces educativas contemplam seus requisitos quando os usuários aprendem a usá-las, e ao fazê-lo aprendem algo ou algum conceito. As interfaces educativas interferem, portanto, no desenvolvimento cognitivo dos usuários, tendo impacto na aprendizagem de campos conceituais determinados (Vergnaud, 1997).

Gomes, Tedesco e Castro-Filho (prelo) observam que uma questão importante envolvendo o desenvolvimento de aplicações educativas está na especificidade dos requisitos não-funcionais de usuários, que, muitas vezes, parecem pouco integralizados neste desenvolvimento. O conjunto de requisitos deve observar não só aspectos do processo de aprendizagem dos alunos, mas também aspectos do processo de mediação a ser promovida pelo professor, o qual pode beneficiar-se de funcionalidades específicas do sistema, como o registro de passos ou a prévia organização de seqüência de problemas. Da mesma maneira, funcionalidades, muito rígidas ou pouco adaptativas, podem tornar um ambiente por demais diretivo e inadequado aos professores e suas práticas (Lins e Gomes, 2003).

---

Elicitação de Requisitos consiste na identificação dos requisitos necessários à implementação de um sistema. Este processo engloba atividades de descoberta, refinamento, modelagem, documentação, especificação e manutenção do conjunto de requisitos. O processo da Engenharia de Requisitos pode ainda ser dividido nas atividades estruturadas de Elicitação, Análise e Negociação, Validação e Gerenciamento (Kotonya e Sommerville, 1998). O levantamento de requisitos de um sistema é normalmente feito por engenheiros. Em sua formulação original recomendava-se que a lista de requisitos fosse construída pelo usuário usando sua linguagem. Trabalhos recentes tentam implicar intensamente o usuário no processo de identificação de requisitos de um sistema.

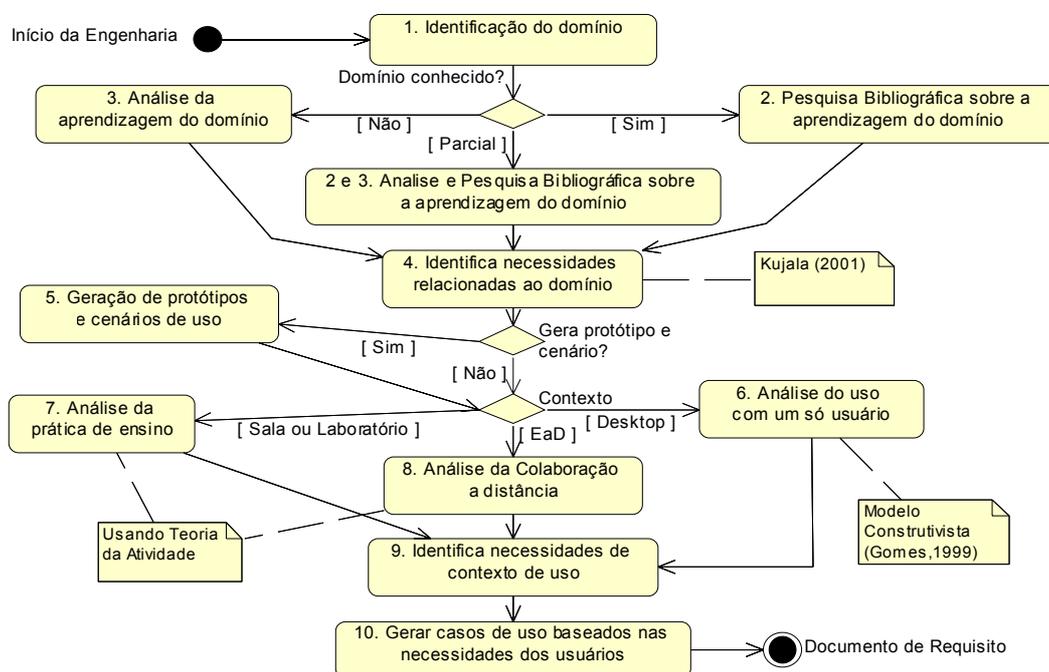
Há diversas dificuldades encontradas por engenheiros de software para elicitar requisitos em domínios muito específicos. A abordagem mais tradicional ainda ocorre de forma não-sistemática, por meios de análise simples ou por meio de entrevistas feitas com os clientes e usuários. No entanto, entrevistas não são suficientes para captar os requisitos de um sistema, já que ela não permite ao pesquisador ter acesso a informações precisas do perfil dos usuários ou do contexto de uso do novo artefato. Vários aspectos importantes para o sistema podem passar despercebidos, ou mesmo não terem sua devida importância garantida com esta técnica. Além de o envolvimento dos entrevistados (clientes e usuários) ficar reduzido.

A maior dificuldade parece esta relacionada em listar um conjunto adequado de requisitos e na criação de casos de uso que atenda às reais necessidades dos usuários. A modelagem inicial de um software educativo esta relacionada à aprendizagem dos conceitos visados, a prática de ensino no contexto de uso e ao currículo (MacDougall e Squires, 1994). Uma primeira característica desse conjunto de requisitos é que ele recebe influência do tipo de domínio de conhecimento para a qual a interface está sendo criada. Em segundo lugar, requisitos e especificações devem levar em consideração a estrutura da atividade dos diferentes usuários que utilizarão a interface (Lins e Gomes, 2003). Para aceder a dados precisos sobre a atividade cognitiva, seja relacionada ao domínio (aprendizagem), seja relacionada ao contexto de ensino (atividade), faz-se necessário que a abordagem do sujeito seja orientada por modelos cognitivos adequados. A seguir é descrito um processo de elicitação de requisitos de domínio (aprendizagem) e de contexto (atividade) para projetos de software educativo.

### **3. Elicitação de Requisitos de Software Educativo**

Dada a complexidade da natureza dos fatores humanos envolvidos no projeto de um software educativo, mesmo técnicas sistemáticas apresentam problema em sua execução. No projeto de tais sistemas, conforme visto, há uma grande variedade de tipos de requisitos a serem identificados, relacionados ao domínio (aprendizagem de conceitos) e ao contexto de uso (atividade). Os requisitos que serão abordados neste artigo são dois, que identificaremos como sendo: domínio e atividade. A modelagem cognitiva da ação permite aceder a informações sobre o domínio, sua aprendizagem. Já a modelagem cognitiva de atividades permite identificar requisitos relacionados às práticas sociais nas quais participam os usuários com artefatos similares àqueles em desenvolvimento. A análise de ambos ocorre mediante orientações de modelos, abordagens construtivistas.

Na Figura 1 apresentamos um diagrama de atividade do processo que temos aplicado no projeto de diversas aplicações educativas.



**Figura 1 - Fluxo do Processo de Criação de um Software educativo**

O processo encontra-se descrito da forma mais genérica possível. Sua aplicação em projetos específicos ocorre quando respondemos a três perguntas: quanto se sabe sobre a aprendizagem do domínio? É necessário criar e estudar o uso de protótipos? Que tipo de contexto de uso deve ser analisado? Descrevemos o projeto de uma interface educativa criada para o ensino de conceitos do campo das estruturas aditivas (Vergnaud, 1997).

### 3.1. Identificação do Domínio

O primeiro momento de um projeto de um software educativo corresponde à identificação do domínio a ser veiculado, em termos de conceitos ou campos conceituais. Entendemos por domínio o campo conceitual a ser apresentado na interface. O domínio de aplicação de um software pode ser distinto como geometria e linguagem de programação. Para o projeto de um software educativo são identificados os domínios desde o início do projeto. Os requisitos estão relacionados com o que a interface deve prover em termos de aprendizagem.

Em primeiro lugar, são essas interfaces que servem para veicular situação e viabilizar contextos nos quais as ações dos usuários são possíveis. Em segundo lugar, porque são essas interfaces que servem de meio mediador para os profissionais de ensino na tarefa de ensinar.

### 3.2. Pesquisa Bibliográfica sobre a aprendizagem do domínio

As fontes de informações de mais simples acesso que informam sobre a elicitação de requisitos relacionados ao domínio são periódicos sobre a aprendizagem de conceitos

específicos. De uma forma geral, a eficácia do software em provocar a aprendizagem “dependerá do grau em que lida com o conhecimento e com concepções que os alunos têm” (Carragher, 1990, p. 36). Esse tipo de fonte de informação já era recomendado por Carragher (1990) que sugeria que equipes de produção de software educativo precisariam incluir especialistas em aprendizagem e desenvolvimento cognitivo, mas que tenham entendimento e domínio sobre os resultados oferecidos pelas publicações sobre aprendizagem de conceitos específicos.

Em áreas como o ensino da Matemática e Ciências, os dados de pesquisas acerca da aprendizagem de conceitos são abundantes. Algumas áreas são menos privilegiadas em termos de resultados de pesquisa, como é o caso de áreas técnicas de formação profissionalizante. A criação de software educativo para essas áreas aproveita uma quantidade menor de informações sobre desenvolvimento cognitivo no domínio. No caso da ausência de literatura suficiente acerca do processo de aprendizagem do conceito do domínio, a alternativa é a realização de estudos sistemáticos. O desenvolvimento da interface analisada na seção 4, o Gérard, beneficiou-se desse tipo de pesquisa. Seu objetivo visa a inserir-se em práticas de ensino-aprendizagem de estruturas aditivas (soma e subtração) que são conceitos muito estudados na literatura.

Estudos realizados com alunos do ensino fundamental mostram que a utilização dos diagramas de Vergnaud (1997) (ver Figura 2), em relação ao uso da representação gráfica de parte-todo e o uso do material concreto auxilia, na resolução de problemas envolvendo estruturas aditivas (Vasconcelos,1998).



**Figura 2 - Diagramas de Gérard Vergnaud**

Vasconcelos mostra que o uso dos diagramas promove um melhor índice de desempenho em relação a outras duas quanto à resolução de problemas aditivos. Essas informações orientaram a criação da interface.

### 3.3. Análise da aprendizagem do domínio

A análise da aprendizagem do domínio permite identificar requisitos associados ao domínio visado da aprendizagem quando esses não são encontrados na literatura. Esta complementa ou compensa a ausência de fontes bibliográficas de dados. O designer de aplicações educativas pode criar experimentos a partir dos quais possam ser observados efeitos do uso da interface sob desenvolvimento na aprendizagem de conceitos específicos.

O processo de criação de softwares educativos demanda a identificação dos conhecimentos que provavelmente emergem nas ações dos usuários com as interfaces, o

---

que ocorre de forma não sistemática (Gomes, 1999). Para se ter uma visão sistemática acerca da relação entre o uso da interface e da aprendizagem é necessário aplicar um modelo teórico de análise que descreva o processo de organização de ação com a interface de forma inteligível. Essa modelagem deve ocorrer em termos de elementos teóricos que revelam a adaptação progressiva dos usuários à interface e os conhecimentos que emergem durante o uso.

A escolha do modelo cognitivo a ser usado neste estado sistemático pode variar muito. Poucos estudos utilizam modelos construtivistas de desenvolvimento cognitivo para modelar as ações dos usuários com os sistemas e orientar, a partir desses resultados, o projeto de aplicações educativas. Há vantagens em adotar-se tal perspectiva como referencial teórico, pois já existe um extenso trabalho nessa tradição. Um modelo construtivista parte do pressuposto de que existem representações internas de conhecimento nos esquemas mentais; a inferência desse conhecimento é tecnicamente possível para um especialista e representa o dado principal do estudo da aprendizagem mediada pelo uso da interface (Gomes, 1999). De uma forma geral, conceitos como esquemas mentais e instrumentos (Vergnaud, 1997; Rabardel, 1995) são adequados à modelagem da aprendizagem que ocorre no uso de uma interface.

Em diversos estudos, adotou-se o modelo de esquema mental proposto por Vergnaud (1997), o qual permite descrever a organização das ações e identificar, por meio de inferências, os elementos e a dinâmica internos de tais esquemas mentais. Esse autor propõe que a representação interna a um esquema seja composta de informações acerca da realidade que rodeia o usuário e do uso de artefatos, de seus objetivos e conhecimentos identificáveis como científicos, mesmo que implícitos e não explicitáveis pelos usuários. Os elementos que constituem um esquema refletem aquilo que é representado pelo sujeito ao agir: é a representação que antecede a ação dos usuários. Esse modelo de esquema mental permite analisar a aprendizagem de conceitos específicos que ocorre numa ação com interfaces, pois o mesmo prevê a existência de conteúdos na organização das ações. Em estudos iniciais com protótipos usam estes conceitos para verificar se a interface promove aprendizagem adequada.

De forma ainda mais precisa, é possível modelar a relação que há entre ações em uma interface e aprendizagem (Gomes, 1999). Propõe-se um modelo que articula atividade material e cognitiva, como ocorre no uso de interfaces (*Ibid.*). Analisam-se as ações dos usuários com as interfaces como sendo ações instrumentais, o que implica adoção do conceito de instrumento assim como definido pela teoria da gênese de instrumentos (Rabardel, 1995), ambos de orientação construtivista.

A partir da utilização dos modelos de esquema e instrumento, é possível estudar a relação que pode haver entre o uso de uma interface e o conjunto de propriedades de conceitos que podem ser aprendidos. Este segundo conjunto é definido como o campo de aplicação das interfaces (Gomes, 1999).

### **3.4. Identifica necessidades relacionadas ao domínio**

A partir dos resultados obtidos na pesquisa bibliográfica e/ou com estudos empíricos relacionados ao domínio, pode-se sintetizar as necessidades dos usuários a serem atendidas pela interface relacionadas ao domínio.

---

A identificação das necessidades relativas ao domínio tem a função de orientar na elaboração de casos de uso, funcionalidades, que estão em conformidade com a proposição de situações Adequadas à aprendizagem dos conceitos do domínio. Casos de uso são gerados de forma orientada a necessidades dos usuários elevando o número de dados sobre a aprendizagem dos usuários que é considerada na especificação do sistema.

### **3.5. Geração de protótipos e análise de cenários de uso**

Durante a realização de estudos para elicitación de requisitos e identificação de necessidades dos usuários, duas alternativas de procedimentos podem ser ponderadas à decisão. A primeira é a realização de um estudo para análise das atividades no contexto, sem o uso de protótipos. Uma segunda opção seria realizar estudos nas quais tenta-se construir um cenário estimado do produto final usando como artefato um sistema similar. Essa corresponde à técnica de cenários (Bødker, 2000). Segundo a autora, as razões para usar cenários em procedimentos de estudo dessa natureza justificam-se quando se pretende encontrar e apresentar soluções, ilustrar soluções alternativas ou identificar problemas potenciais antes do produto estar terminado.

A análise de cenários de uso permite a realização de estudos iniciais da adequação de interfaces educativas (Gomes *et al.* 2002) para verificar se a mesma promove a emergência de situações favoráveis à aprendizagem de conceitos do domínio. Trata-se de uma maneira de antecipar e permitir o estudo de fatores humanos da interface para identificação de requisitos do domínio e relativos ao processo de mediação a ser promovido com a mesma.

Deve-se evitar confundir os sentidos da palavra cenários nesse contexto. Em UML (*Unified Modeling Language*), um cenário corresponde às seqüências básicas e alternativas de um caso de uso. Numa abordagem de projeto com o envolvimento dos usuários, cenários podem ser criados para compor uma situação de estudo, servindo a antecipar uma situação futura e para testar alternativas. O resultados das análises desses estudos geram casos de uso, cujos cenários, seqüências básicas e exceções, são descritas de tal maneira a atenderem as necessidades dos usuários em seu contexto.

### **3.6. Análise do uso do usuário Aluno**

Parte dos requisitos de um software educativo visa facilitar o uso dos usuários finais. Tais aplicações podem ser usadas em diversos contextos e por diferentes conjuntos de usuários. Pode ser usado por um usuário apenas, ou por mais de um usuário. Estes podem encontrar-se lado a lado ou estarem distantes uns dos outros interagindo através de ambientes virtuais (Bricker, 1993).

Entender a prática no contexto de uso da aplicação orienta igualmente o projeto da interface (Nardi, 1996). A partir da análise de estudo do contexto, torna-se explícita a classe de requisitos que são destinados a informar o designer de como as funções atendem as necessidades dos usuários no contexto de uso.

Adota-se para tal os preceitos da Teoria da Atividade (Leontev, 1975) dado seu alto grande de abstração utilizada para a modelagem da atividade de usuários e grupos de usuários. Para essa teoria, qualquer atividade humana é mediada e, portanto, suas estruturas são passíveis de análise. A utilização desse modelo permite expressar as

---

relações que existem entre os elementos que estruturam uma atividade. A partir da identificação desses elementos, fica mais simples inferir requisitos que uma interface, a ser inserida no contexto da mesma atividade, deve possuir (Martins e Daltrini, 1999).

A atividade é analisada segundo a hierarquia de sua estrutura. Atividades são orientadas por motivos ou necessidades. Ao agir, os usuários praticam ações motivadas por objetivos. Finalmente, ao concretizar uma ação, os usuários realizam operações que são mediadas por algo, algum artefato. A modelagem dessa estrutura hierárquica também orienta na elicitação de requisitos de usuário.

Métodos de pesquisa etnográfica ficaram populares no campo de estudo de interação usuário máquina. Estes métodos de pesquisa são usados no design de uma grande variedade de produtos (Kujala, Kauppinen e Rekola, 2001). O uso destes métodos para o desenvolvimento de produto caracteriza-se como forma de reunir exigências de clientes para avaliação de campo de uso de novos produtos e serviços. A execução destes métodos inclui trabalho de campo em condições reais, com uma perspectiva objetiva e com ricas descrições das pessoas, ambientes e interações existentes. Engloba o entendimento amplo das variáveis ambientais e do contexto de uso dos produtos que estão sendo projetados.

Um dos problemas da pesquisa etnográfica é que ela consome muito tempo, elemento raro no ciclo de desenvolvimento do produto. Na busca de resolver este problema de tempo, utilizam-se técnicas de pesquisa etnográfica rápida (Millen, 2000). A etnografia rápida cresceu baseada em inúmeras experiências da pesquisa de campo em um número de projetos diferentes.

### **3.7. Análise da prática de ensino**

Alguns aspectos da interface de software educativos interferem na atividade de sala de aula, ou em laboratório de informática (Lins e Gomes, 2003). Observamos que a maioria dos softwares é construída dentro de um paradigma específico que projeta o software num contexto da atividade com apenas um usuário. Parece não haver uma preocupação em observar a atividade normal de um professor num laboratório de informática educativa para, a partir dessa observação gerar requisitos que promovam o design de ferramentas adequadas, num processo conhecido como design centrado no usuário. De outra forma, observa-se que os softwares parecem ser construídos sem que haja a preocupação com a estrutura da atividade na qual o mesmo será inserido. Assim, a análise da estrutura da atividade em um laboratório de informática educativo deveria ser considerada para fins de orientar o design das funcionalidades de um futuro software educativo, atendendo novamente as necessidades de professores e alunos no contexto.

Após análise da estrutura das atividades, procede-se novamente com a análise de necessidades associadas a cada uma das dimensões da atividade. Identificam-se necessidades relativas: aos usuários (*user needs*), às regras (*rules needs*), à comunidade (*community needs*), à divisão do trabalho (*work needs*), ao objeto (*object needs*).

### **3.8. Análise da Colaboração a Distância**

Com relação ao contexto de uso, há diferentes enquadramentos da relação entre o usuário e o sistema. Para cada um dos enquadramentos, diferentes questões relativas à relação podem ser pertinentes. A primeira delas é representada pela relação direta entre

---

o usuário e a interface do sistema, num contexto de uso no qual apenas um usuário está envolvido. A análise da colaboração é realizada em estudos sistemáticos envolvendo a observação e também podem ser analisados pela teoria da atividade.

### **3.9. Identifica necessidades de contexto de uso**

A análise das práticas de aprendizagem e ensino culmina com a identificação de um novo conjunto de necessidades desses usuários.

De posse dos resultados obtidos nos estudos preliminares podemos iniciar a fase de especificação das interfaces. Esta especificação deve considerar os resultados até então obtidos. Os dois conjuntos de resultados são necessidades dos usuários. Tabelas de necessidade do usuário é uma maneira útil de representar necessidades do usuário para fazê-las compreensíveis e úteis para os designers (Ibid.). Nas tabelas de necessidade do usuário unem-se problemas e possibilidades dos usuários a seqüências de tarefas. Tabelas de necessidade do usuário ajudaram os designers a traduzir necessidades do usuário sejam relacionadas a domínio ou a contexto de uso em exigências do usuário e escrever casos de uso a partir do ponto de vista do usuário. Os designers não receberam meramente informação sobre o que os usuários precisam, mas eles vêem ainda como processos podem ser mudados e o contexto de uso considerado.

O processo aqui descrito amplia o alcance do método tradicional pelo fato de estar voltado à identificação de necessidades dos usuários em contexto de uso. Diferente da forma como casos de uso são propostos num processo não centrado no usuário, os cenários resultantes de uma abordagem centrada no usuário correspondem a formas de funcionamento que visam atender aos requisitos de domínio e do contexto de uso (atividade). Novamente nesse nível, procedemos com a organização de duas classes de necessidade. A primeira delas correspondendo a necessidades características e associadas ao domínio visado. A segunda delas relativa às necessidades de usuários com relação ao contexto de uso.

### **3.10. Gerar Casos de Uso baseado nas necessidades do usuário**

As necessidades do usuário se referem à diferença entre metas dos usuários e também às presentes condições, que são manifestadas por problemas e possibilidades dos usuários, e o contexto de uso, que inclui as características de futuros usuários, suas tarefas atuais e seu ambiente. Esse contexto de uso necessita ser considerado e respeitado.

O diagrama de caso de uso faz parte dos chamados modelos de requisitos. O diagrama de caso de uso possui uma descrição que mostra como o sistema e o ator vão interagir através de fluxos. O fluxo principal descreve as interações normais (sem erros) entre o ator e o sistema. A descrição ainda contém fluxos alternativos, que são os erros, que podem ocorrer quando o ator e o sistema interagem, e esses erros podem ser previstos em estudos como descritos em 3.2, 3.3, 3.7 e 3.8. Os diagramas de caso de uso são usados para demonstrar as funcionalidades que o sistema a ser criado irá prover. Esse diagrama revela ainda a forma como atores (pessoa ou outro sistema) interagem com o sistema proposto.

Casos de Uso ajudam designers a ganhar uma visão coerente do produto. Dessa maneira, detalhes indefinidos perdidos foram identificados e o trabalho de definição não procedeu tão rápido para os detalhes técnicos. Os designers podem usar o caso de uso

como checklist para guiar o trabalho de especificação e implementação do sistema. Exigências são descrições formais. São divididas em exigências do usuário e exigências técnicas. As exigências do usuário são escritas a partir de seu ponto de vista. Elas descrevem qualquer função, restrição ou outra propriedade que deve ser provida para satisfazer as necessidades do usuário. Portanto, as exigências do usuário descrevem como um futuro produto pode ajudar usuário a alcançar suas metas efetivamente, eficientemente, e com satisfação em seu contexto de uso. Exigências que orientam como o produto deverá ser implementado para conhecer as exigências do usuário.

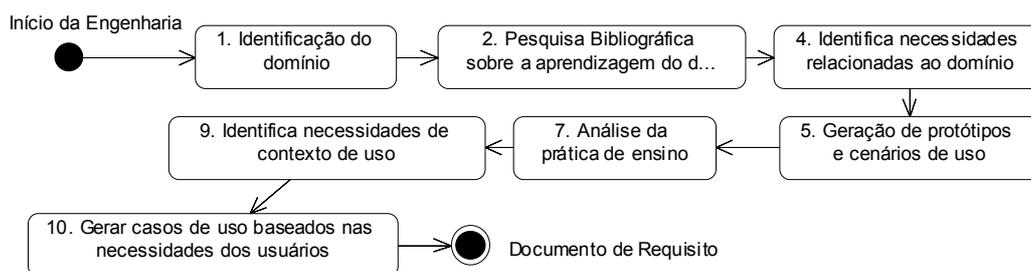
#### 4. Estudos de Caso

A interface denomina-se Gérard em homenagem ao pesquisador Gérard Vergnaud, propositor da Teoria dos Campos Conceituais (Vergnaud, 1997). A interface tem o propósito de permitir aos alunos a resolução de problemas no campo das estruturas aditivas a partir do uso de representações criadas pelo usuário, seguindo a legenda proposta por Vergnaud para as estruturas aditivas. As estruturas aditivas são classificadas de forma a obter-se uma estrutura teórica para compreender as diferentes representações simbólicas da adição e subtração.

A interface fornece o ambiente gráfico necessário à criação e manipulação direta dos símbolos gráficos que representam os conceitos aditivos. Segundo Vergnaud, a utilização de diagramas auxilia na sistematização do problema, durante o cálculo relacional. A interface fornece ainda a possibilidade de gerar diagramas para resolução de problemas a partir da estrutura aditiva criada na interface gráfica do Gérard. A utilização da notação de diagramas propostos por Vergnaud permite a diferenciação entre os diversos tipos de problemas existentes e enfatiza o reconhecimento de toda a classe de situações problemas. Isso leva professores e alunos a analisar, de forma mediada e explícita as relações envolvidas, as operações de pensamento e os procedimentos necessários para resolver cada classe de problemas.

O Gérard permite a modelagem dos diferentes tipos de relações existentes, o que favorece a emergência de diferentes sentidos de números (Nunes e Bryant, 1997) como comparação, transformação e quantidade. O sistema verifica a consistências da estrutura editada pelo usuário a cada modificação. A Figura 3 resume as etapas usadas no projeto dessa interface e que já foram discutidas.

Identificamos três problemas principais na construção do Gérard. O primeiro está relacionado com a construção e manutenção de inúmeros componentes com funções semelhantes. O segundo está relacionado com a necessidade de manipulação desses componentes, individualmente ou em grupo. O terceiro problema é a forma como a verificação da consistência do diagrama gerado pode ocorrer.



**Figura 3 - Fluxos de Tarefas para Elicitação do Gérard**

---

A partir dos itens (1) e (3), conseguiu-se identificar uma quantidade elevada de requisitos da interface –uso de gráficos, necessidade de manipulação e o tipo de forma de interação, a forma como as representações contidas na interface são manipuladas. As necessidades da interface com relação à aprendizagem foram observadas na construção da interface. Além disso, a identificação de requisitos influenciou na modelagem do sistema, em particular na definição de padrões de projeto mais adequados aos requisitos. Um protótipo foi gerado (5) e o mesmo está sendo submetido a testes de usabilidade (6) para permitir a análise do uso pelo aluno. A geração de casos de uso (9) ocorreu baseada nos resultados de (7) e dos itens (1) e (3).

## 5. Conclusões

Este artigo apresenta um processo de elicitação de requisitos para a criação de software educativo. Discutiu-se o estado da arte de processos de criação de interfaces de sistemas de software educativo. Apresentaram-se as etapas de um processo de elicitação que está sendo aplicado em diversos projetos de aplicações educativas (*learnware*). Observou-se que a parte mais difícil do projeto de sistemas educativos não é construí-los, mas saber o que se vai construir. Designers acham que sabem tudo sobre usuários e suas necessidades, quando na verdade eles não sabem e não se dão conta disso. Portanto, nós temos apresentado uma maneira de se representar, na qual as necessidades do usuário e seu ponto de vista podem ser apresentados nas definições de exigências, e pela qual a ligação entre necessidades do usuário e suas exigências pode ser reduzida.

O problema da elicitação de requisitos para o desenvolvimento de aplicações educativas é um problema conhecido (Gomes, Tedesco e Castro-Filho, prelo). De uma forma geral, os softwares são especificados a partir da representação que designer e programadores têm daquilo que acreditam deva ser levado em consideração no design da interface (Mandel, 1997). Durante o projeto do Gérard, as decisões de projeto não são tomadas de forma *ad hoc*. Essas foram substituídas por resultados de pesquisas e inferência de resultados de estudos empíricos. Com isso, esperam-se criar interfaces educativas de boa qualidade pedagógica de forma sistemática.

## Agradecimentos

Projeto financiado pelo CNPq/ProTeM-CC Proc. n. 680210/01-6 e n. 477645/2001-1.

## Bibliografia

- Bødker S., Scenarios in user-centred design-setting the stage for reflection and action, *Interacting with Computers*, 13 (2000) 61-75.
- Bricker L., *Designs for Computer-Supported Cooperative Learning in Mathematics*. Partial fulfillment of the University of Washington Generals exam, 1993.
- Carraher, D.W. (1990). O que esperamos do Software Educacional? *Acesso - Revista de Educação e Informática*, Ano II, n. 3, jan./jun. 1990, issn 0103-0736;
- Gomes, A. S., Tedesco, P. A.; Castro Filho, J. A., Ambientes de Aprendizagem em Matemática e Ciências. In: Edla Maria Faust Ramos (Org.). *Informática na Escola: um Olhar Multidisciplinar* (prelo), Fortaleza: Editora UFC.

- 
- Gomes, A.S. (1999) *Développement conceptuel consécutif a l'activité instrumentée - L'utilisation d'un système informatique de géométrie dynamique au collège*, Thèse de doctorat, Université Paris V, Paris [www.cin.ufpe.br/~asg].
- Kotonya, Gerald e Sommerville, Ian.: *Requirements Engineering: Processes and Tecniques*, John Wiley & Sons, Ltd, 1998;
- Kujala, S., Kauppinen, M., e Rekola, S. (2001). Bridging the Gap between User Needs and User Requirements. In Avouris, N. e Fakotakis, N. (Eds.) *Proceedings of the Panhellenic Conference with International Participation in Human-Computer Interaction PC-HCI 2001*, Typorama Publications, pp. 45-50.
- Leont'ev, A.N. (1975). *Actividad, conciencia, personalidad*. Habana: Editorial Pueblo y Educación, edition in Spanish 1981, edited from original in Russian 1975.
- Mandel T., *The Elements of user interface*, John Wiley and Sons: New York, 1997;
- Martins L. E. G. e Daltrini B. M., *Utilização dos preceitos da Teoria da Atividade na Elicitação dos Requisitos do Software*, SBES'1999, pp. ;
- McDougall, A. e Squires, D. (1995) *An Empirical Study of a New Paradigm for Choosing Educational Software*, *Computers and Education*, (25) 3, pp. 93-103.
- Millen D. R., *Rapid Ethnography: Time Deepening Strategies for HCI Field Research*, DIS '00, Brooklyn, New York, 2000.
- Nardi, B.A.(1996) *Activity Theory and Human-Comp. Interaction*. London: MIT Press.
- Rabardel P., *Les hommes et les technologies - Approche cognitive des instruments contemporains*, Paris: Armand Colins, 1995;
- Rieman J., *A field Study of Exploratory Learning Strategies*, *ACM Transactions on Computer-Human Interaction*, (3) 3, pp. 189-218, 1996;
- Tchounikine P., *Pour une ingénierie des Environnements Informatiques pour l'apprentissage humain*, *Information-Interaction-Intelligence*, vol. 2, n. 1, 2002.
- Vasconcelos, L. (1998) *Problemas de adição e subtração: modelos teóricos e práticas de ensino*, In Analúcia Schliemann e David Carraher (Orgs.) *Compreensão De Conceitos Aritméticos*, Papirus: São Paulo, pp. 32-45;
- Vergnaud G. (1997) *The nature of mathematical concepts*. In T. Nunes e P. Bryant (Eds.), *Learning and teaching mathematics: An international Perspective*, Psychology Press, Hove, pp. 5-28.