

# Software Design Cooperativo: Um Estudo de Caso

Renato C. Duarte    Hugo Fuks    Carlos J. P. Lucena

hugo@Inf.puc-rio.br

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro

Rua Marquês de São Vicente 225

22453-900 RJ Brasil

## Resumo

O desenvolvimento de *software* é uma atividade tipicamente coletiva. Envolve uma equipe de pessoas, cada qual com responsabilidades, comunicando e colaborando com outros participantes da equipe. Este trabalho apresenta alguns aspectos relacionados ao *software design* realizado cooperativamente. Estes aspectos foram obtidos a partir da observação e análise de sessões de *software design* realizado por grupos de voluntários. A técnica de análise desta atividade cooperativa desenvolvida para este estudo é descrita e tem sua aplicação exemplificada através de um estudo de caso.

## Palavras-chave

*software design*, trabalho em grupo, *groupware*, metodologia de desenvolvimento de *software*

## 1. Introdução

O desenvolvimento de *software* é uma atividade que envolve vários profissionais, analistas e usuários na concepção, projetistas e programadores na elaboração e gerentes na coordenação do desenvolvimento. Além dos aspectos técnicos, o desenvolvimento de *software* é reconhecido como um processo social composto de interações entre os membros da equipe. Negociações e compromissos fazem parte do processo cooperativo de desenvolvimento de *software*.

O Trabalho Cooperativo Assistido por Computador (TCAC) é a área da Ciência de Computação que estuda as funções, atribuições e relações de trabalho entre grupos de pessoas e sistemas de computação (Greif 83). Algumas aplicações têm sido desenvolvidas com base nos estudos em TCAC. Sistemas de automação de escritório e sistemas para tele-conferência podem ser concebidos para uso de grupo de pessoas em seu ambiente de trabalho. Os *softwares* que fornecem suporte a atividades cooperativas são denominado *groupwares*.

O desenvolvimento de *software* também tem sido área de estudos em TCAC. (Mujica 91) explora as extensões necessárias para que um sistema de computação possa dar suporte às equipes de trabalho envolvidadas no projeto e realização de sistemas complexos. O DAIDA, outro ambiente experimental, é descrito em (Hahn, Jarke & Rose 91). Este ambiente é empregado no desenvolvimento de sistemas baseados em conhecimento e na manutenção de sistemas de informação baseados em uso intensivo de banco de dados. Uma modelagem conceitual integra as diferentes tarefas encontradas no desenvolvimento de *software*.

Algumas ferramentas são uma alternativa para a introdução de TCAC na atividade de desenvolvimento de *software*. Diferente dos ambientes para suporte de grande parte desta atividade, as ferramentas podem ser usadas em determinadas situações que ocorrem durante o desenvolvimento de *software*. O uso de um sistema de informação baseado em questões - gIBIS - em um projeto de desenvolvimento de *software* é relatado em (Yakemovic & Conklin 90). O gIBIS foi utilizado para registro e consulta de justificativas de projeto. A tecnologia de TCAC serviu para dar suporte às conversações e informações, que circulam ao longo do desenvolvimento, como fonte de justificativas de projeto. O sistema SIBYL descrito em (Lee 90) fornece suporte às tomadas de decisão através da representação e gerenciamento de aspectos qualitativos do processo de decisão: alternativas, metas e argumentos. SIBYL utiliza uma linguagem de representação de decisão (DRL).

Os trabalhos citados acima baseiam-se em modelos cuja intenção é representar aspectos inerentes à atividade de *software design* cooperativo. A abrangência destes modelos varia desde a representação de tomada de decisão (Yakemovic & Conklin 90, Lee 90) até a representação

conceitual de *design* cooperativo (Hahn, Jarke & Rose 91). O emprego efetivo destes ambientes e ferramentas é determinado pelo grau de adequação destes modelos à atividade de *software design* cooperativo. É nesta categoria que se insere a *Research Agenda on Software Design - RASD* (Lucena, Leite, Schwabe & Fuks 91), que tem como uma de suas metas a incorporação de aspectos cooperativos em metodologias de desenvolvimento de sistemas. Um estudo descritivo de atividades, em espaço de trabalho compartilhado, de *design* conceitual é apresentado em (Tang 89). Este estudo tem como premissa a necessidade de entendimento do que os participantes realmente fazem durante uma atividade de *design* cooperativo para poder guiar o desenvolvimento de tecnologia para dar suporte a esta atividade.

O estudo e compreensão da atividade de *software design* são, também, os objetivos da pesquisa descrita neste trabalho. Para alcançar estes objetivos foi realizado um experimento onde foi possível estudar atividades - sem o auxílio de computador - de *software design* cooperativo. O experimento, composto de observação de sessões de *software design*, foi realizado com oito grupos de voluntários. A técnica de observação destas atividades junto com a técnica de análise desenvolvida compõe uma metodologia proposta neste trabalho para o levantamento de aspectos inerentes ao domínio de *software design* cooperativo.

A seção 2 descreve o experimento - as sessões de *software design*. São abordados o processo de elaboração do experimento, o perfil dos membros dos grupos, a metodologia empregada e o *software*-exemplo projetado por cada grupo. Na seção 3 é relatada a técnica empregada no experimento para observar as sessões de *software design* e justifica o emprego desta abordagem comparando-a com outras possíveis abordagens. Na seção 4 é relatada a técnica desenvolvida para análise dos dados obtidos como resultado da aplicação da técnica de observação. Na seção 5 são descritos aspectos inerentes à atividade de *software design* cooperativo resultantes da análise dos dados levantados no experimento. Na seção 6 são sugeridos alguns requisitos para um sistema que dê apoio à atividade cooperativa de *software design*. Estes requisitos foram levantados após um estudo dos aspectos relacionados à *software design* cooperativo descritos na seção 5. E na seção 7 são apresentadas as conclusões deste trabalho. O apêndice contém uma reprodução do resultado do processo de análise de uma sessão de *software design* cooperativo.

## 2. O Experimento

O experimento realizado como subsídio ao estudo da atividade de *software design* cooperativo foi composto por oito sessões onde grupos voluntários realizaram o *design* conceitual de um *software*-exemplo. Este experimento possibilitou a aplicação de técnicas de observação e análise das atividades dos grupos. Como resultado da aplicação destas técnicas foi possível o levantamento de aspectos inerentes à atividade de *software design* cooperativo. O experimento foi elaborado e aplicado no curso Projeto de Sistemas de Software - 91.2 - Departamento de Informática, PUC-RJ, ministrado por um dos autores.

O desenvolvimento de ferramentas CASE foi definido como trabalho de curso. Quatro metodologias foram apresentadas no início do curso: Análise Estruturada, SADT, Orientação a Objetos e Jackson System Development (JSD). A ferramenta CASE a ser implementada deveria dar suporte a uma destas metodologias apresentadas. Devido ao porte do trabalho, a opção de trabalho em grupo tornou-se mais adequada. A escolha da metodologia ficou a cargo de cada grupo assim como a definição dos componentes dos grupos. Foram formados oito grupos, cada um com quatro participantes em média. Cada grupo teria que realizar um *design* conceitual de um *software*-exemplo utilizando a metodologia escolhida na presença do professor. Desta forma seria possível validar o conhecimento de cada grupo.

O *design* do *software*-exemplo deveria ser a única preocupação do grupo durante a atividade de *design* cooperativo. Portanto, o *software*-exemplo deveria se referir a um domínio de aplicação conhecido pelo grupo. Evitaria perda de tempo durante a etapa de interpretação da especificação do *software*-exemplo, o que poderia comprometer o experimento. Para garantir que o *software*-exemplo estivesse no domínio de aplicação conhecido pelo grupo foi solicitado que cada grupo elaborasse e entregasse uma especificação de um *software* para uma determinada aplicação à escolha do grupo.

As especificações propostas pelos grupos foram suficientemente modificadas para tornarem-se os *software*-exemplo do experimento. Os domínios de aplicações foram mantidos para cada grupo. Para os grupos que escolheram Análise Estruturada e SADT alguma funcionalidade foi alterada, alguns procedimentos retirados e outros acrescentados. Para os grupos que escolheram JSD novas entidades foram acrescentadas e outras retiradas. As sessões de *software design* cooperativo foram iniciadas com a entrega para os grupos da especificação final do *software*-exemplo.

Os alunos tiveram a liberdade de formar os grupos. O objetivo era que os alunos usassem o critério de familiaridade para a escolha dos componentes de cada grupo. A formação de um grupo onde todos os participantes se conhecem torna espontânea a interação entre os membros e propicia maior coesão do grupo no desenvolvimento da atividade. Era necessário que os componentes participassem da atividade da forma mais espontânea possível. A diversidade de interações que surge pode trazer uma riqueza de situações que, após análise, podem ajudar a identificar aspectos considerados intrínsecos à atividade de *software design* cooperativo.

O critério de escolha da metodologia foi a variedade de referências e exemplos de aplicação da metodologia. Os domínios de aplicação do *software*-exemplo variavam de sistemas para bibliotecas até sistema para consultório dentário. Todos os sistemas, entretanto, tinham em comum as características de sistemas de controle administrativo baseados em banco de dados simples, acompanhados dos devidos procedimentos de manutenção e consulta às informações.

### **3. Técnica de Observação**

Durante as sessões de *software design* cooperativo ocorre uma variedade de informações não estruturadas. Estas informações podem ser as frases ditas, as ações realizadas, a tarefa de cada membro etc. O objetivo de se aplicar uma técnica de observação é identificar as informações que são relevantes no contexto da atividade, isto é, os dados de interesse no estudo de *software design* cooperativo. A técnica de observação prescreve de que forma serão levantados os dados de interesse durante as sessões. O exame dos resultados da observação pode propiciar um entendimento maior das características intrínsecas da atividade de *software design* cooperativo para formulação de questões que motivem estudos mais profundos. A abordagem observacional deste trabalho consistiu no registro em vídeo da atividade realizada pelo grupo. Duas outras abordagens - experimental e análise de protocolos - foram descartadas por não se adequarem à natureza da atividade de *software design* cooperativo.

#### **3.1. Abordagem Experimental**

Os dados a serem observados devem ser determinados a priori para uma verificação experimental. É necessário formular uma hipótese que relacione o resultado com variáveis independentes, bem como estruturar uma situação que manipule de forma controlada as variáveis independentes. Após o experimento deve-se confirmar ou refutar esta hipótese comparando os resultados obtidos. Para garantir a validade da hipótese deve-se discutir os resultados, buscando uma generalização além da situação examinada. A validade do estudo experimental de atividades humanas depende da capacidade de manipular o comportamento dos participantes para construir diferentes condições. Deve haver a capacidade de medir com precisão os resultados de cada condição.

Esta abordagem é inadequada para este trabalho pois é difícil obter-se métodos confiáveis para a manipulação do comportamento dos participantes ou para a medição, com precisão dos resultados produzidos pela equipe. Eliminar a complexidade pode significar eliminar a essência da atividade criativa de *design*. Esta abordagem pode ser efetiva quando aplicada às atividades onde são conhecidas as variáveis independentes e sabe-se como manipulá-las.

#### **3.2. Abordagem Protocolos Verbais**

Consiste na expressão verbal da informação usada durante um processo de *design* realizado por um indivíduo. O indivíduo deve relatar constantemente tudo que se passa em sua mente. Com uma análise posterior das informações levantadas pode ser possível identificar elementos-chaves de uma atividade de *design* (Tang 89). A abordagem de protocolos verbais assume *design*

como um processo cognitivo que ocorre dentro da mente do indivíduo, que pode ser descrito através de planos, estratégias, operadores etc.

A atividade em grupo está relacionada com a interação entre indivíduos. Num grupo, os indivíduos se comunicam através de palavras e gestos. Qualquer gesto ou palavra dita por um indivíduo pode ser interpretada por outros indivíduos do grupo. Cada gesto ou palavra pode ser, portanto, entendido pelos outros como uma comunicação intencional, isto é, o indivíduo teve a intenção de passar alguma idéia para os outros participantes do grupo.

A aplicação desta abordagem no estudo de atividade cooperativa apresenta algumas dificuldades. Pode haver confusão durante a interação do grupo quando os integrantes falassem tudo que passa em suas mentes ao mesmo tempo que também falam outras coisas com a intenção de comunicar. Cada expressão externa da informação na mente do indivíduo poderia ser interpretada, erroneamente, pelo resto do grupo como uma intenção de comunicação. Esta abordagem, portanto, não é adequada ao estudo da atividade de *design* cooperativo. Está orientada ao estudo de atividades individuais e não em grupo.

### 3.3. Abordagem Observacional

O objetivo desta abordagem é coletar o mais naturalmente possíveis as informações relevantes que ocorrem durante a atividade cooperativa de *design*. Esta técnica que tem como base o registro completo da atividade de *software design* cooperativo no espaço de trabalho do grupo, pressupõe a não interferência no processo criativo do grupo. Não são impostas regras de comunicação entre os membros nem são pré-definidas situações.

O espaço de trabalho do grupo constitui-se de elementos físicos necessários para a interação entre os membros do grupo. O espaço de trabalho é escolhido pelo grupo dentre uma série de possibilidades à disposição do grupo. Para a realização do experimento preparou-se uma sala contendo um quadro negro, uma prancheta com papéis para escrita, canetas de cores diferentes, uma mesa e cadeiras (figura 1). Cada grupo escolhia de acordo com seu estilo de interação quais elementos do espaço seriam utilizados. O registro das sessões foi feito através de filmagem com uma câmera VHS. A câmera deve estar localizada de forma a permitir que todo o grupo possa ser filmado em seu espaço de trabalho. Antes do experimento, para garantir que os grupos agissem de forma espontânea com a presença da câmera, foi explicado o motivo da filmagem. Os grupos tornaram-se, portanto, voluntários uma vez que permitiram que suas atividades fossem registradas. Mesmo assim, alguns membros ficaram inibidos devido a presença da câmera.

Foram relacionados, antes do experimento, alguns aspectos que acreditávamos serem interessantes de observar. Estes aspectos auxiliaram na identificação de algumas situações inerentes à atividade de *software design* cooperativo:

- Organização da equipe: - Existe organização formal/informal?  
- Qual(is) o(s) critério(s) para esta organização?
- Interação entre os membro do grupo: - Como a passagem da palavra é controlada?  
- Como são resolvidos os conflitos?
- Registro das deliberações: - Existe algum critério que oriente quais deliberações serão registradas?  
- De que forma são registradas?  
- São classificadas?
- Uso do material para consulta: - Quando e de que forma são utilizados?
- Uso do material impresso: - De que forma o grupo interage com a informação escrita?  
(textos e gráficos) - Os membros da equipe utilizam alguma área de trabalho particular?

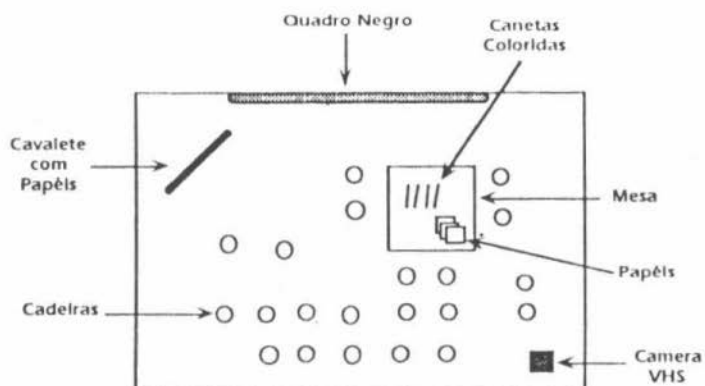


Figura 1: Ambiente Físico do Experimento

#### 4. Técnica de Análise

A técnica de análise desenvolvida para este trabalho tem como base a organização de uma sessão de *design* em situações que foram classificadas nas categorias: *software design* propriamente dito e o processo de interação.

O ponto de partida para o desenvolvimento desta técnica foi a percepção de que a atividade de *software design* cooperativo pode ser vista através de dois enfoques: do ponto de vista de um projetista de *software* e do ponto de vista de um especialista em interação. O projetista de *software* teria a preocupação de identificar - durante a análise dos registros da sessão - situações características de *software design* (não somente em relação ao emprego da metodologia bem como em relação as atitudes tomadas e recursos empregados) e classificá-las. O especialista em interação deveria identificar e classificar situações intrínsecas ao processo de interação dos membros entre grupo (padrões e recursos empregados na comunicação).

Durante as observações dos registros em fita VHS das sessões realizadas podem ser identificadas algumas situações que frequentemente ocorrem em diversas ocasiões. Estas situações podem ser classificadas sob o ponto de vista de *software design* e sobre o processo de interação. Não existe uma regra de como identificar as situações relevantes. A identificação pode partir do interesse no *software design* ou no processo de interação.



Figura 2: Diagramação do Apêndice

Para representar o produto desta análise usou-se uma tabela dividida em três colunas. Na coluna central são resumidas situações ocorridas durante a sessão. Nas duas colunas restantes são identificados e classificados aspectos relativos ao *software design* e ao processo de interação entre os participantes do grupo (figura 2). Um resultado da aplicação desta técnica de análise dos registros é relatado no Apêndice.

## 5. Aspectos de Software Design Cooperativo

Após a aplicação da técnica de análise das sessões, obtém-se um conjunto de aspectos inerentes à atividade de *software design* cooperativo. Estes aspectos podem estar relacionados com diversas situações diferentes, indicando uma diversidade de possibilidades de ocorrência deste aspectos. Nesta seção são relatados e classificados os aspectos identificados a partir do experimento com os oito grupos mencionado na seção 2.

### 5.1. Aspectos Relacionados ao Software Design

A forma da equipe se organizar, o papel da liderança, os planos, a reutilização de outros trabalhos, as justificativas não registradas, o desenvolvimento incremental, o *backtracking* no *design*, a flexibilidade na aplicação da metodologia e a divisão em sub-grupos são os aspectos que possuem uma relação com o domínio da atividade de *software design*.

#### 5.1.1. Organização do Grupo

A organização do grupo envolve a definição das funções que cada membro deve realizar. Através da definição destas funções pode ser identificado o papel que cada membro deve assumir. Um membro que se preocupa, com exemplo, com a aplicação correta da metodologia assumiria o papel de consultor da metodologia. Os membros podem ainda assumir mais de um papel. A organização pode possuir um caráter estático ou dinâmico em relação ao momento em que as funções são definidas e atribuídas aos membros. O grupo pode definir as funções antes (organização estática) ou durante (organização dinâmica) o processo de *software design*.

A definição das funções e atribuição dos papéis para cada membro pode ser um processo voluntário ou não. Um membro pode se voluntariar para cumprir um determinado papel acreditando ter competência para cumprí-lo. Por outro lado, um papel pode ser atribuído a um determinado membro sem que este tenha manifestado interesse em cumprí-lo. Uma organização pode ainda ser formal ou informal. É formal quando cada membro tem seu papel definido e todo grupo tem conhecimento do papel de cada membro. Em uma organização informal os papéis são assumidos de forma implícita. As funções são identificadas sem que o papel fique explicitamente definido para todo o grupo.

Foram classificados alguns papéis observados nas sessões de *software design* cooperativo:

*Consultor do domínio*: - auxiliar o grupo na interpretação da especificação do *software*;  
- validar o *design*, durante sua construção, com relação à especificação.

*Consultor da metodologia*: - auxiliar o grupo no emprego da metodologia;  
- certificar o emprego correto da metodologia.

*Líder de projeto*: - determinar planos para alcançar a meta: o *design* correto do *software*;  
- motivar os membros no cumprimento de suas tarefas;  
- cobrar resultados dos membros.

*Escreva*: - manipular a representação escrita do *design*.

#### 5.1.2. Liderança

O papel do líder pode ser formalmente constituído ou não. O líder formalmente constituído assume este papel explicitamente, isto é, todos os membros sabem quem é o líder e quais são as suas funções. O líder informalmente constituído é um membro que realiza as funções de líder implicitamente, sem ficar claro para o grupo qual é seu papel. Nos grupos organizados informalmente surge a figura do sub-líder, outro membro que realiza eventualmente as funções de líder.

Um líder pode ter atitudes democráticas ou não. Uma atitude é democrática quando o líder toma a atitude depois de justificar para o grupo as razões que o levaram a tomar esta atitude. O líder assume uma postura não democrática quando toma uma atitude antes de justificar para o grupo. Um líder pode combinar estas duas posturas durante a atividade de *software design*. O líder pode assumir atitudes mais democráticas em situações de dúvida. Quando o líder não sabe o que fazer, fica mais disposto a aceitar as opiniões do grupo antes de tomar atitudes.

### 5.1.3. Planejamento

Planejamento é a tarefa de definir os passos necessários - o plano - para alcançar uma determinada meta. O planejamento pode ser estático ou dinâmico com relação ao momento em que ocorre. O planejamento pode ainda envolver o grupo todo na discussão do plano ou membros isolados. O planejamento é uma tarefa que pode ser realizada ou não pelo líder, mas a decisão final é tomada pelo líder. O planejamento estático ocorre antes de cada sessão de *design*. São definidos quais passos do *design* devem ser realizados e em qual ordem. A metodologia atua como uma orientação sobre o plano. O planejamento dinâmico ocorre durante a sessão de *design*. As ações e passos são definidos de acordo com o estágio atual do *design*. As restrições de tempo, isto é o cumprimento de um cronograma, atuam como um fator incentivador de re-planejamento.

### 5.1.4. Reuso

Durante uma sessão de *design* ocorrem consultas a referências que contêm outros *designs* e suas respectivas especificações. Estas consultas ocorrem em momentos de dúvida do grupo ou de membros isoladamente. A consulta pode ter como objetivo fornecer uma idéia inicial (síntese) sobre o *design* ou para comparar o *design* com outro semelhante (avaliação).

Da forma como o experimento foi estruturado (conhecimento prévio do domínio do problema), percebeu-se que a atividade de *design* se iniciou, frequentemente, com uma tentativa de reuso. Os grupos tentam usar como ponto de partida alguma referência (um *software* semelhante, por exemplo). Esta referência pode fornecer uma idéia de como estruturar uma primeira etapa do *design* ou ajudar na definição de quais os passos devem ser seguidos (planejamento).

### 5.1.5. Justificativas de Design (Design Rationales)

O processo de resolução durante uma tomada de decisão culmina em uma deliberação. Durante este processo as opções, sugeridas por membros do grupo, são analisadas e avaliadas. Os resultados da análise e avaliação das ações constituem as razões pelas quais determinadas decisões foram tomadas. O registro das justificativas possibilita a documentação da tomada de decisão, servindo de auxílio para a manutenção e evolução do *software* bem como para identificação de erros que ocorreram em fases anteriores do processo de *software design*.

Durante sessões de *design* as tomadas de decisão podem ter ou não a participação de todos os membros do grupo. Nem todos membros sugerem, analisam ou avaliam sugestões. O membro que sugeriu não é, frequentemente, o mesmo que avalia. O processo de resolução pode ou não ser iniciado pelo líder, mas é encerrado por ele. É o líder que determina qual decisão tomar segundo sua própria avaliação ou consenso do grupo.

Dos oito grupos observados somente em um deles houve a preocupação de anotar em papel algumas deliberações (registradas pelo líder). As razões que levaram às deliberações, entretanto, não foram registradas. Os grupos que não registraram suas deliberações tiveram dificuldades em lembrá-las. Durante o processo de *software design* ocorrem divisões das metas em sub-metas de contextos diferentes. Como as deliberações são associadas a contextos e o processo é recursivo, quando um contexto é retornado as deliberações, não registradas, deste contexto são esquecidas.

### 5.1.6. Desenvolvimento Incremental

O processo de *software design* é exploratório e o *design* é desenvolvido de forma incremental. O *design* evolui seguindo uma direção determinada pela ordem crescente de certeza do grupo. O grupo toma primeiro as atitudes que acreditam ter mais certeza de levarem ao objetivo da atividade. As atitudes mais corretas são aquelas onde suas justificativas são mais bem aceitas pelo grupo.

### 5.1.7. Backtraking

A evolução do *design* do *software* ocorre em etapas. Cada etapa é associada a uma configuração da representação do *design*. As alterações na configuração do *design* determina uma mudança de etapa. O *design* em uma etapa pode ser abandonado e ser restaurada uma configuração do *design* de uma etapa anterior. Isto acontece quando são percebidos erros no *design* provenientes de decisões não corretas em uma etapa anterior. Pode ainda acontecer o retorno a um *design* de uma etapa anterior para correção de erros sem abandonar o *design* na etapa atual. Neste caso, são necessárias alterações na configuração da etapa atual de acordo com as modificações realizadas na configuração do *design* da etapa anterior.

### 5.1.8. A Metodologia

O *design*, durante momentos, pode não se apresentar sintaticamente correto segundo as regras de representações da metodologia empregada. Durante fases exploratórias (síntese) algumas regras sintáticas da metodologia são flexibilizadas. Alguns diagramas podem não estar condizentes com a metodologia. Isto permite que o grupo possa refletir sobre o *design*, buscando meios de expressar primeiramente a semântica desejada sem se ater a regras sintáticas.

Em vários momentos são citadas e mostradas as regras de aplicação da metodologia. Este acesso tem por objetivo facilitar a atividade do grupo evitando tomadas de decisão incorretas quando ocorrem dúvidas sobre aplicação da metodologia.

### 5.1.9. Divisão em Sub-grupos

Um grupo pode dividir-se em sub-grupos. Esta divisão pode ser formalmente decidida ou não. É formalmente realizada quando os objetivos de cada sub-grupo são definidos e todo grupo conhece os objetivos de cada sub-grupo. Não é formalmente realizada quando o grupo é dividido sem uma definição explícita dos objetivos dos sub-grupos. Os membros podem participar voluntariamente ou não dos sub-grupos. Um membro pode ser designado para um sub-grupo à revelia de seu desejo ou pode se voluntarizar para participar de um sub-grupo. Uma conversa informal entre membros próximos pode caracterizar a formação voluntária de um sub-grupo.

Dependendo da localização dos sub-grupos é possível que um sub-grupo possa acompanhar em segundo plano (*background*) as atividades de outros sub-grupos. Mesmo sem participar das atividades de outros sub-grupos, um sub-grupo pode saber como anda os outros grupos se for possível vê-los e ouvi-los.

## 5.2. Aspectos Relacionados ao Processo de Interação

Foram identificadas diversas situações que ocorrem durante a atividade dos grupo. Estas situações foram classificadas sob o enfoque do processo de interação entre os membros dos grupos. As formas e os protocolos de comunicação e os mecanismos de interação são os aspectos obtidos a partir da classificação de diversas situações.

### 5.2.1. Formas de Comunicação Empregadas na Interação

As formas de comunicação são os recursos empregados para a comunicação entre membros do grupo. Após a análise das situações foi possível identificar três tipos de recursos. A comunicação pode ocorrer através de expressões verbais, expressões corporais ou com o uso de recursos físicos.



### 5.2.1.1. Expressões Verbais (Locuções)

As expressões verbais foram classificadas de acordo com o objetivo da comunicação no contexto em que elas ocorrem. Frases inteiras podem ser classificadas como locuções. Estas locuções são dependentes do contexto, isto é, são empregadas de acordo com outras locuções e com a intenção do que se pretende comunicar. Quando um membro diz algo com intuito de discordar de uma afirmação de outro membro, ele está empregando a locução **refutar**. Abaixo estão relacionadas as locuções empregadas pelos grupos seguidas de alguns exemplos de frases características.

*Questionar* - Por que você colocou este fluxo de dados?  
- Este processo está correto?

*Responder* - Sim, o processo está correto.

*Justificar* - Coloquei este fluxo porque este processo trata o dado.

*Afirmar* - O processo Sócio tem realiza três ações.

*Aceitar* - Concordo com o que você disse.

*Refutar* - Discordo do que você disse.

*Pedir Sugestão* - O que fazemos agora? Não sei o que fazer.

*Sugerir* - Que tal se explodíssemos este processo primeiro?

- Não estou bem certo, mas acredito que devemos fazer o dicionário de dados em paralelo.

*Pedir Clarificação* - Não entendi o que vocês fizeram. Me expliquem.

*Clarificar* - Fizemos isto porque a especificação indica claramente a existência desta ação.

*Chamar atenção* - Gostaria que todos me escutassem.  
(focus of attention)

### 5.2.1.2. Expressões Corporais (Ações)

As expressões corporais são classificadas de acordo com as ações realizadas pelos membros do grupo. As ações são relevante pois são usadas efetivamente no processo de comunicação. Pode envolver o sujeito da ação (geralmente um membro) e um objeto (elemento físico da interação). Um membro pode apagar um pedaço do *design* em vez de responder verbalmente que concorda com uma sugestão de outro membro. Algumas ações observadas estão abaixo relacionadas.

*Escrever* - O membro X escreve a entidade Y no *design*.

*Apagar* - O membro X apaga a entidade Y do *design*.

*Mostrar* - O membro X mostra para o grupo um exemplo de outro *design* de *software*.

*Dar atenção* - O líder para de falar com o membro X para poder ouvir o membro Y.

*Apontar* - O membro X aponta para um trecho do *design* enquanto fala com o resto do grupo.

*Tomar área de escrita* - O membro X vai ao quadro com a intenção de escrever sua idéia.

*Liberar área de escrita* - O membro X se afasta do quadro, permitindo que outros membros possam usá-lo para escrever.

### 5.2.1.3. Recursos Físicos da Interação

Os recursos físicos da interação são objetos do mundo real que podem ser utilizados no processo de comunicação. Alguns dos objetos identificados são apresentados a seguir:

Membros do grupo

*Design* (representação escrita do *design* no papel ou no quadro negro)

Referência (um livro ou artigo)

Metodologia (um manual da metodologia)

Área de escrita (quadro negro ou folha de papel)

Os recursos físicos podem ser manipulados através das expressões corporais. Um *design* pode ser alterado ou um membro pode dar atenção a outro membro. Os recursos físicos podem ser citados por expressões verbais. Uma referência pode servir de base para uma sugestão ou uma justificativa pode citar trechos do manual da metodologia.

### 5.2.2. Protocolos de Comunicação

Um protocolo é uma composição de formas de comunicação (expressões verbais, corporais e físicos) empregadas durante uma situação de comunicação entre membros. Os protocolos podem ser classificados de acordo com o contexto e o objetivo da comunicação. Um protocolo de negociação poderia ser composto de uma sequência de louções (sugestões, refutações, afirmações e justificativas) e ações (apagar, apontar e escrever no *design*) entre dois membros.

### 5.2.3. Mecanismos de Interação

Uma rede de comunicação é constituída por membros que estão envolvidos num mesmo processo de interação. As redes permitem identificar com quem cada membro está se comunicando. Quando todos os membros do grupo estão participando de uma discussão, uma única rede abrange todos os membros. Quando sub-grupos são formados, sub-redes são também formadas para estes sub-grupos. Uma rede de comunicação deve envolver pelo menos dois participantes.

Quando sub-grupos são formados, ocorrem processos paralelos de interação. Estes processos podem eventualmente compartilhar de forma concorrente um mesmo elemento físico do espaço de trabalho (o quadro ou uma folha de papel).

Para que sub-grupos se unam é necessário que ocorra um sincronismo entre os processos de comunicação de cada sub-grupo. Uma forma de sincronização, é um sub-grupo X solicitar a atenção de outro sub-grupo Y. O sub-grupo Y interrompe suas atividades para dar atenção ao sub-grupo X. Neste momento os dois sub-grupos estão unidos em um mesmo processo de comunicação.

Um membro pode ser o responsável por determinar quem vai falar e quando. Por outro lado, a comunicação pode ser não controlada, ocasionando muitas interações concorrentes (todos membros falando ao mesmo tempo com o líder). Neste caso é necessário dotar os protocolos de mecanismos que garantam a efetividade dos protocolos concorrentes (o líder pode ser capaz de ouvir todos os membros falando simultaneamente, mesmo não dando atenção total a nenhum membro específico).

Os processos de comunicação podem ser interrompidos durante o período que ocorre. Um membro pode ser capaz de intervir na comunicação de outro membro, interrompendo-o e iniciando outro processo de comunicação.

## 6. Requisitos Preliminares para um Sistema de Apoio ao Software Design Cooperativo

A partir do estudo dos aspectos identificados após a análise dos registros das sessões foi possível levantar um conjunto de requisitos que acreditamos serem adequados a um sistema que dê suporte à atividade cooperativa de *software design*. Ao longo desta seção, este sistema será denominado Sistema de Apoio ao *Software Design Cooperativo* (SDCoop). A definição do SDCoop deve ser orientada por algumas diretrizes necessárias para garantir que este sistema atenda seu objetivo de forma satisfatória.

A primeira diretriz diz respeito a forma de utilização do sistema pelos membros da equipe. A interação do sistema com os membros deve ser a mais natural possível (como se a tecnologia não estivesse presente no trabalho do grupo). A segunda diretriz diz respeito à abrangência do SDCoop. O sistema seria empregado durante a etapa de desenho do *software*, deixando de fora as etapas de especificação e implementação do *software*.

Por fim, o SDCoop deveria ser concebido de forma a fazer uso de um sistema de computação adequado para seu objetivo. O sistema seria baseado em estações de trabalho individuais interligadas em rede, permitindo a comunicação entre as estações. Estas estações não poderiam

estar em locais distintos, com um telão para a exibição da janela pública. O ambiente que acabamos de descrever foi usado em (Mujica 91). Observe-se que o nosso experimento não permite tirar conclusões sobre o que ocorre na cooperação sem o elemento face-a-face.

Os requisitos descritos a seguir devem atender às necessidades de suporte que correspondam às características já discutidas: a organização social dentro da equipe, as tomadas de decisão, ao uso de metodologia, a evolução da atividade, as formas de comunicação e as necessidades de interação entre os membros da equipe.

### **6.1. Organização Social**

O sistema não deve se basear em modelos que definem os papéis a serem cumpridos pelos membros da equipe. Estes modelos podem impor restrições na operação do SDCoop pela equipe violando a diretriz de naturalidade de manuseio. Desta forma a própria equipe indicaria para o sistema quais os tarefas e o papel que um membro estaria desempenhando. Ao sistema caberia apenas manter um registro do papel e das tarefas realizadas por cada membro. Deve ser explicitado para o sistema qual membro realizaria o papel de líder. O membro líder teria um status diferente no sistema. Alguns recursos do sistema estariam disponíveis somente para o líder (prioridade de comunicação, acesso à todas as informações etc). Mesmo dividida em sub-grupos, todos os membros teriam acesso às interações que ocorrem dentro da equipe (inclusive dentro de outros sub-grupos).

### **6.2. Suporte à Decisão**

O sistema deve apoiar ações visando o planejamento das atividades da equipe. Permitir a atribuição de membros para as tarefas a serem realizadas e a definição de passos, metas e prazos a serem cumpridos. O sistema deve permitir o registro de decisões tomadas e suas respectivas justificativas. O sistema não deve identificar automaticamente as decisões e suas justificativas. Estas decisões e justificativas a serem registradas devem ser informadas para o sistema. O sistema deve manter uma biblioteca de outras atividades de *software design* realizados para reuso em outras ocasiões. Estes registros seriam compostos de um histórico das etapas realizadas (com uma representação do *design*) e das decisões e justificativas de cada etapa.

### **6.3. Metodologia**

O sistema deve auxiliar a equipe fornecendo informações sobre as regras e exemplos de como aplicar a metodologia. Para isso, o sistema deve ter acesso a uma base que contenha estas informações. Como o SDCoop deve ser expansível, com relação as metodologias empregadas, a criação e manutenção desta base deve ser externa ao sistema. Além disso, o sistema deve permitir a aplicação incorreta da metodologia em certos momentos. O sistema, entretanto, deve ser capaz de verificar o *design* quanto a aplicação correta da metodologia quando solicitado pela equipe. A tarefa de verificação também deve ser externa ao sistema pois depende da metodologia empregada pela equipe.

### **6.4. Evolução**

O sistema deve apoiar o desenvolvimento incremental do *design*. A equipe identificaria e solicitaria ao SDCoop o registro das etapas realizadas ao longo da atividade cooperativa. O sistema deve permitir consulta à etapas anteriores (já registradas) do *design*. Deve possibilitar, também, o retorno à etapas anteriores (descartando as etapas intermediárias) quando for solicitado.

### **6.5. Formas de Comunicação**

O sistema deve fornecer um conjunto de formas de comunicação (locuções, ações e elementos físicos). Este conjunto deve conter o mínimo necessário de locuções e ações capazes de atender às necessidades de comunicação da equipe. As locuções ficariam explicitamente à disposição dos membros como opções de comunicação. Algumas ações seriam implícitas de acordo com a interação entre o SDCoop e os membros (apontar, por exemplo). Outras seriam explicitamente solicitadas ao sistema (dar atenção, por exemplo). Os elementos físicos poderiam ser

manipulados pelas ações (alterar desenho, por exemplo) ou referenciados pelas locuções (afirmação citando uma regra da metodologia, por exemplo).

O sistema não deve orientar a comunicação entre os membros, restringindo as locuções ou ações disponíveis a cada momento de acordo com o contexto de cada situação. Este tipo de orientação requer o uso de um modelo pelo sistema. Os modelos propostos atualmente restringem o uso de *groupwares* por grupos tornando menos natural e intuitivo o uso destes *groupware's* pelos grupos.

## 6.6. Mecanismos de Comunicação

O sistema deve ser capaz de identificar com quem cada membro está interagindo. A divisão do grupo ou conversas paralelas entre membros devem ser tratadas, pelo sistema, através do conceito de redes de comunicação. Cada rede seria constituída pelos membros que participassem do mesmo processo de interação. O SIDCoop deve permitir acesso compartilhado à recursos utilizados concorrentemente por membros distintos da equipe ou de sub-grupos.

O sistema deve possuir um mecanismo que possibilite a sincronização dos processos de comunicação de sub-grupos distintos, unindo-os num único processo de comunicação. A locução chamar atenção poderia ser utilizada para avisar outro subgrupo da intenção de sincronização. A ação dar atenção, em resposta àquela locução, completaria o processo de sincronização.

O sistema poderia ser configurado para permitir tanto o controle da palavra por um determinado membro (decide quem e quando fala) quanto a comunicação não controlada (fala quem e quando quer). Além disso, o sistema deveria permitir a intervenção e/ou interrupção por um membro no processo de comunicação de outros membros.

## 7. Conclusão

Neste trabalho foi desenvolvida e empregada uma técnica de análise de sessões de trabalho cooperativo para o estudo da atividade de *software design*. A aplicação desta técnica permite a identificação de aspectos característicos do domínio da atividade de *software design* cooperativo. A técnica de análise objetiva separar os aspectos semânticos (relativos ao domínio da atividade) e os aspectos sintáticos (relativos ao processo de interação). A aplicação desta técnica requer a organização dos dados (obtidos da observação) levando em conta diferentes situações. Do ponto de vista sintático, a preocupação é identificar as formas e mecanismos de interação dos membros do grupo com o ambiente externo. Do ponto de vista semântico, a preocupação é a identificação de elementos do domínio da atividade.

Quando separados os aspectos segundo os dois focos, pode-se perceber a semelhança de *software design* com a atividade de solução de problemas. Algumas características do espaço de problemas em *design* podem ser identificadas: decomposição de solução, tomada e propagação de decisões e emprego de sistemas de símbolos (Goel & Pirolli 89).

Sobre o ponto de vista do processo de interação foram identificados alguns elementos relacionados à forma de comunicação entre os membros do grupo. Estes elementos (locuções e ações) quando agrupados definem protocolos de comunicação. Este trabalho não pretende obter protocolos comunicação a partir desses elementos. Para tal seria necessário desenvolver um modelo que descreva o processo de comunicação entre indivíduos realizando *software design* cooperativamente. O modelo de interação pode ser desenvolvido de forma semelhante ao descrito por (Laufer, Fuks & Schwabe 92). Elementos primitivos (modificadores de locução) são associados para obter protocolos de comunicação (clichês). O modelo deveria, entretanto, ter como elementos primitivos as locuções, ações e os elementos físicos do espaço de trabalho. Deveria abranger n-participantes e prever situações de concorrência e paralelismo ao longo da atividade.

A partir do estudo de atividades cooperativas de *software design* são sugeridos alguns requisitos para um sistema que dê apoio a esta atividade. Este trabalho não pretende abordar a viabilidade de construção deste sistema com estes requisitos. A possibilidade de elaboração de um sistema com estes requisitos é ainda uma tarefa para futuros estudos.

## **Agradecimentos**

Gostaríamos de agradecer a Julio C. P. Leite pelas observações e críticas feitas a uma versão deste trabalho.

## **Referências**

- Greif, I.: Computer Supported Cooperative Work: A Book of Readings; Morgan Kaufman Publishers, 1988.
- Goel, V. & Pirolli, P.: Design within Information Processing Theory: The Design Problem Space; AI Magazine, pp 19-36, Spring 1989.
- Hahn, U., Jarke, M. & Rose, T.: Teamwork Support in a Knowledge-Based Information Systems Environment; IEEE Transactions on Software Engineering, Vol.17, n<sup>o</sup>.5, pp 467-482, May 1991.
- Laufer, C. C., Fuks, H. & Schwabe, D.: Representação de Clichês de Conversação para Cooperação; Monografias em Ciência da Computação n<sup>o</sup> 1/92, Departamento de Informática, PUC-RJ, Março 1992.
- Lee, J.: SIBYL: A Tool for Managing Group Decision Rationale; CSCW 90 Proceedings, pp 79-92, October 1990.
- Lucena, C. J. P., Leite, J., Schwabe, D. & Fuks, H.: A Research Agenda for Software Design; Monografias em Ciência da Computação n<sup>o</sup> 29/91, Departamento de Informática, PUC-RJ, Março 1991.
- Mujica, S. T.: A Computer-Based Environment for Collaborative Design; PhD Thesis, University of California, Los Angeles, 1991.
- Tang, J. C.: Listing, Drawing, and Gesturing In Design; a Study of the Use of Shared Workspaces by Design Teams; Xerox Parc, SSS-89-3, April 1989.
- Yakemovic, K. C. B. & Conklin E. J.: Report on a Development Project Use of an Issue-Based Information System; CSCW 90 Proceedings, pp 105-118, October 1990.

## Apendice: Análise da Sessão do Grupo 7 (JSD, Clube Social)

<i>Software Design</i>	Trechos de sessão de <i>software design</i> cooperativo	Processo de Interação
	(JSD:Lista de Entidades e Ações)	
	R vai ao quadro.	Tomar área de escrita (a) <sup>1</sup>
Reuso	C sugere baseado em um <i>design</i> anterior.	Sugerir (l) <sup>2</sup>
Escreva voluntário	R escreve no quadro.	Escrever (a)
	R para de escrever. R "Por que isso?"	Questionar (l)
Reuso	C justifica citando o <i>design</i> anterior.	Justificar (l)
	R se convence continuando a escrever no quadro.	Aceitar (l)
	Membros lançam questões. Outros (ou todos) respondem.	Questionar (l) Responder (l)
	R escreve e sugere. Questões direcionadas para R.	
	R comunica a todos o seu ponto de vista.	Afirmar (l)
Deliberação	As idéias são lançadas ao grupo. Alguns concordam "Tem razão", "concordo") e outros se omitem (silêncio).	
	Não há política de controle de palavra. Fala quem tem alguma ideia. Alguns falam ao mesmo tempo.	Controle da palavra (c) <sup>3</sup>
Negociação	R "Você não quer fazer assim.."	Sugerir (l)
	C "Não sei."	Responder (l)
	Momento de dúvida. O grupo não sabe o que fazer. Todos refletem.	Não ocorre interação
	R pede sugestões.	Pedir sugestão (l)
	C faz pergunta para o grupo.	Questionar (l)
	D responde com uma idéia.	Sugerir (l)
	R lança nova dúvida para C.	Questionar (l)
Liderança	(Fica claro que C está se sobressaindo no grupo. C realiza e responde muitas questões)	
	M faz pergunta ao grupo.	Questionar (l)
	Todos respondem de forma desordenada.	Responder (l)

<sup>1</sup> (a) Ação - Expressão corporal

<sup>2</sup> (l) Locução - Expressão verbal

<sup>3</sup> (c) Mecanismo de interação

Reuso	C clarifica uma dúvida usando um exemplo do material de consulta (C narra sua idéia sem mostrar para o grupo qual exemplo em que sua idéia se baseia).	Clarificar (I)
	O grupo continua discutindo. (O grupo não conseguiu entender o que C disse).	
Reuso	C mostra para o grupo o exemplo do material de consulta.	Mostrar (a)
	C justifica com o exemplo sua posição.	Justificar (I)
	Todos se calam.	Acceptar através do silêncio
	<i>(JSD: Diagrama do Processo Sócio)</i>	
Reuso	R escreve no quadro um diagrama de outro <i>software</i> desenvolvido anteriormente pelo grupo. Todos sugerem alterações no diagrama.	Escrever (a) Sugerir (I)
Deliberação	R coloca no quadro o que é consentimento do grupo.	Escrever (a)
Planejamento	D e M sugerem fazer o texto estruturado.	Sugerir (I)
	R concorda. R escreve no quadro.	Acceptar (I) Escrever (a)
Planejamento	C interrompe R que estava escrevendo. C diz que R, D e M querem pular passos do JSD.	Intervir (c) Refutar (I)
	Todos concordam com a agumentação de C.	Acceptar (I)
Liderança informal	(C exerce uma liderança sobre o grupo).	
	<i>(JSD: Diagrama do Processo Quadras)</i>	
Flexibilização de regras da metodologia	R escreve um diagrama incompleto (algumas caixas sem ligações e rótulos).	
Desenvolvimento incremental	(Alterações pouco a pouco no <i>design</i> : tentativa e erro).	
	Momento de dúvida. Todos param e pensam no que fazer.	
	R aponta o <i>design</i> . R sugere uma alteração.	Apontar (a) Sugerir (I)
Sub-grupos informais	R e M conversam. C e D conversam outro assunto.	Concorrência (c)
	R sugere. M discorda. M justifica.	Sugerir (I) Refutar (I) Justificar (I)
União de sub-grupos	C e D escutam a conversa de R e M. C, e D entram na discussão de M e R.	Focar (a) Sincronismo (c)
	R, C e D discordam de M.	Refutar (I)

	M sugere outra idéia. O grupo não entende o que M disse.	Sugerir (I) Pedir clarificação (I)
Escriba voluntário	M vai ao quadro. M desenha outro diagrama. M explica apontando um novo diagrama.	Tomar área de escrita (a) Escrever (a) Apontar (a) Clarificar (I)
	R, C e D não esperam o fim da explicação de M: refutam a sugestão de M.	Intervenção (c) Refutação (I)
	M concorda em silêncio.	Aceitar através do silêncio.
	M se afasta do quadro.	Liberar área de escrita (a)
Escriba voluntário	C vai ao quadro. C escreve sua sugestão de <i>design</i> .	Tomar área de escrita (a) Escrever (a)
	O grupo discorda de C.	Refutar (I)
	R vai ao quadro e altera o <i>design</i> sugerido por C.	Tomar área de escrita (a) Escrever (a)
Líderança democrática	C aceita e volta atrás em sua posição.	Aceitar (I)
	(JSD: Diagrama do Processo Secretária)	
Escriba voluntário	M dita para R escrever no quadro as ações da secretária.	Afirmar (I) Escrever (a)
	(JSD: Diagrama do Processo Diretoria)	
Deliberação	R escreve o <i>design</i> enquanto o resto do grupo, em consenso, diz para R o que escrever.	Afirmar (I) Escrever (a)
	(JSD: Diagrama SSD)	
	Antes de escrever R comunica sua dúvida para o grupo.	Questionar (I)
	C faz uma sugestão para esclarecer a dúvida de R.	Sugerir (I)
Desenvolvimento incremental	R esboça um modelo inicial no quadro.	Escrever (a)
Planejamento	O grupo inicia o texto estruturado do processo Sócio.	
	R escreve o texto no quadro.	Escrever (a)
	R para de escrever. R tem dúvidas sobre um nome (nomeclatura).	Pedir sugestão (I)
Rationales	O grupo percebe que R não sabe qual nome usar no texto. O grupo sugere vários nomes para R.	Sugerir (I)



Deliberação	R sugere um nome. Todos concordam. R escreve esse nome no texto estruturado.	Sugerir (l) Aceitar (l) Escrever (a)
	R termina o texto estruturado. Pergunta ao grupo: "É isso?"	Pedir confirmação (l)
	O resto do grupo responde: "É."	Aceitação (l)
	C sugere alterações no <i>design</i> .	Sugerir (l)
	M e R perguntam a C por quê.	Pedir clarificação (l)
Escreva voluntário	C vai ao quadro.	Tomar área de escrita (a)
	C justifica sua posição falando e apontando no diagrama.	Justificar (l) Apontar (a)
	(JSD: <i>Diagrama das Funções</i> )	
	C esboça o diagrama e depois justifica.	Escrever (a) Justificar (l)
	C pergunta ao grupo: "É isso ?"	Pedir confirmação (l)
Acesso à metodologia	M discorda se baseando nas regras da metodologia.	Refutar (l)
Acesso à metodologia	O resto do grupo diz a M que ele fez uma interpretação incorreta do exemplo do livro.	Justificar (l)
Desenvolvi- mento incremental	C esboça o <i>design</i> aos poucos.	Escrever (a)
	M sugere.	Sugerir (l)
	C discorda. C justifica falando e apontando o <i>design</i> .	Refutar (l) Apontar (a)
	M concorda com C.	Aceitar (l)
Rationales	(Para cada incremento no <i>design</i> são necessárias justificativas).	
	C escreve e justifica.	Escrever (a) Justificar (l)
	M escreve uma sugestão no <i>design</i> .	Escrever (a)
	R aponta o diagrama e pergunta: "Por que isso?"	Apontar (a) Questionar (l)