



COPPE/UFRJ

**AGENDABUILDER: UMA FERRAMENTA PARA APOIAR O DESIGN DE
PROCESSOS COLABORATIVOS**

Marcos da Silva Ferreira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador(es): Jano Moreira de Souza

Adriana Santarosa Vivacqua

Rio de Janeiro

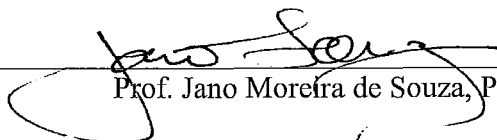
Setembro de 2009

AGENDABUILDER: UMA FERRAMENTA PARA APOIAR O DESIGN DE
PROCESSOS COLABORATIVOS

Marcos da Silva Ferreira

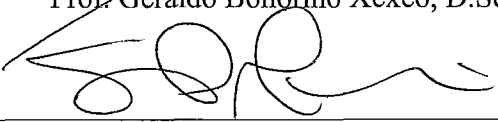
DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:


Prof. Jano Moreira de Souza, Ph.D.


Prof. Adriana Santarosa Vivacqua, D.Sc.


Prof. Geraldo Bonorino Xexéo, D.Sc.


Prof. Sergio Palma da Justa Medeiros, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2009

Ferreira, Marcos da Silva

AgendaBuilder: Uma ferramenta para apoiar o design de processos colaborativos / Marcos da Silva Ferreira. – Rio de Janeiro: UFRJ/COPPE, 2009.

XI, 91 p.: il.; 29,7 cm.

Orientador: Jano Moreira de Souza

Adriana Santarosa Vivacqua

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2009.

Referências Bibliográficas: p. 72-78

1. Trabalho Colaborativo Suportado por Computador.
2. Facilitação. 3. Reunião eletrônica. 4. Processos colaborativos. I. Souza, Jano Moreira de *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Agradecimentos

Em primeiro lugar, agradeço a Deus, por me manter sempre no caminho da retidão e por todas as oportunidades que abriu em minha vida, permitindo que eu alcançasse conquistas que nem imaginava conquistar um dia.

Agradeço à minha família, por todos os incentivos. Agradeço especialmente a minha mãe, por todo o carinho, sabedoria que me transmitiu durante minha vida e pela paciência nos meus momentos de dificuldades durante o mestrado.

Agradeço também a minha mãe espiritual, Jeanette Salgado, por toda a orientação e apoio nos momentos de necessidade.

Agradeço ao Prof. Jano de Souza e a Prof. Adriana Vivacqua, meus orientadores, por todo o suporte dado e a oportunidade para conclusão com sucesso desse trabalho.

Ao Prof. Geraldo Xexéo e ao Prof. Sérgio Palma, por se mostrarem dispostos a integrarem essa banca, mesmo com os muitos compromissos e pouco tempo disponível.

Agradeço aos meus colegas da Petrobras João Menezes, Lays Teixeira, Leandro Marques e Vitor do Valle, por sempre me incentivarem para a conclusão desse trabalho e não me deixarem desistir.

Agradeço também aos meus amigos e colegas de equipe na Petrobras Almir Gaspar, André Silva, André Rodrigues, Cátia Furtado, Jefferson Colares, Marcelo Fernandes, Marcos Freitas, Peeter Bonomo, Ramón Carius, por formarem uma excelente equipe, tornando o dia a dia do trabalho mais fácil. Agradeço também ao meu chefe, Rodrigo Panza, pela compreensão nos meus momentos de ausência em função do mestrado.

A todos os meus amigos não citados aqui, mas que sempre contribuíram para aliviar, nos momentos de descontração, as tensões inerentes à responsabilidade do mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

AGENDABUILDER: UMA FERRAMENTA PARA APOIAR O DESIGN DE
PROCESSOS COLABORATIVOS

Marcos da Silva Ferreira

Setembro/2009

Orientadores: Jano Moreira de Souza

Adriana Santarosa Vivacqua

Programa: Engenharia de Sistemas e Computação

Este trabalho visa propor uma abordagem para auxiliar facilitadores na criação de processos colaborativos. A abordagem proposta faz uso dos *ThinkLets*, que consistem em padrões de dinâmicas de grupos pré-definidos e catalogados. Um processo colaborativo pode ser composto por uma sequência encadeada de atividades, que podem ser executadas através de um *ThinkLet*. A ferramenta implementada busca auxiliar um facilitador na escolha adequada de um *ThinkLet* para a execução de cada atividade.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AGENDABUILDER: A TOOL TO SUPPORT THE DESIGN OF COLLABORATIVE
PROCESSES

Marcos da Silva Ferreira

September/2009

Advisors: Jano Moreira de Souza

Adriana Santarosa Vivacqua

Department: System and Computing Engineering

This dissertation proposes an approach for helping facilitators in the task of designing collaborative processes. The approach is based on the utilization of *ThinkLets*, which consist of pre-defined group dynamics patterns. A collaborative process can be composed of a sequence of activities that can be mapped to a *ThinkLet*. The implemented tool tries to support a facilitator in the adequate choice of a *ThinkLet* to each activity.

Sumário

1	Introdução.....	1
1.1	Contextualização	3
1.2	Organização da Dissertação	4
2	Revisão da Literatura.....	5
2.1	Trabalho Cooperativo Suportado por Computador	5
2.2	Apoio a reunião para tomada de decisão.....	7
2.3	Facilitação	8
2.4	Engenharia de Colaboração.....	10
2.5	ThinkLets.....	14
2.6	Sistemas correlatos	16
2.6.1	GroupSystems	16
2.6.2	Meeting Works (MW).....	19
2.6.3	Ferramentas de apoio ao facilitador	21
3	Abordagem Proposta	23
3.1	ThinkLets como componentes do processo colaborativo.....	23
3.2	Mapeamento dos ThinkLets por padrões colaborativos.....	24
3.3	Mapeamentos dos ThinkLets por tipo de resultado.....	26
3.4	Mapeamento dos ThinkLets através de entradas/saídas.....	27
3.5	Outras regras para escolha de ThinkLets	36
3.6	O Sistema DynaMeeting	38
3.6.1	O módulo <i>AgendaBuilder</i>	40
3.6.1.1	Visão geral da ferramenta e sua interface.....	45
3.6.1.2	Interface de Administração.....	51
4	Implementação	55
4.1	A arquitetura do <i>AgendaBuilder</i>	55
4.2	Plano de experimento	61
4.2.1	Planejamento	61
4.2.2	Análise dos Resultados.....	63

4.2.2.1	Análise quantitativa	63
4.2.2.1	Análise qualitativa	63
4.3	Análise crítica de par	64
5	Conclusão	66
5.1	Trabalhos futuros.....	66
	Referências Bibliográficas.....	72
	Apêndice A - Tabela de recomendações para escolha de ThinkLet.....	79
	Apêndice B - Módulo core – Diagrama de Pacotes.....	84
	Apêndice C - Módulo de Administração – Diagrama de Pacotes	85
	Apêndice D - Diagrama de sequência	86
	Apêndice E - Modelo de Dados.....	87
	Apêndice F - Fonte de regras (Drools) – Tamanho do grupo.....	88
	Apêndice G - Fonte de regras (Drools) – Perfil do grupo	89
	Apêndice H - Fonte de regras (Drools) – Entrada do processo	90

Lista de Figuras

Figura 2-1 Matriz de contextualização de CSCW (adaptado de BURSTEIN & HOLSAPPLE, 2008)	6
Figura 2-2 Visão geral da Engenharia de Colaboração (adaptado de KOLFSCHOTEN, G. et al, 2007)	12
Figura 2-3 Categorização no GroupSystems	18
Figura 2-4 Agenda Planner, MeetingWorks	21
Figura 3-1 Mapeamento de <i>ThinkLets</i>	28
Figura 3-2 Mapeamento de compatibilidade baseado em entrada/saída	37
Figura 3-3 Regras para escolhas de ThinkLets	38
Figura 3-4 Visão geral do escopo suportado pela ferramenta <i>AgendaBuilder</i>	41
Figura 3-5 Princípio de funcionamento do <i>AgendaBuilder</i>	43
Figura 3-6 Contexto do processo colaborativo	44
Figura 3-7 Visão geral da ferramenta	46
Figura 3-8 Barra de ferramentas	46
Figura 3-9 Construção fluxo – exemplo	47
Figura 3-10 Registrando informações iniciais sobre o processo	48
Figura 3-11 Registrando informações sobre atividades	49
Figura 3-12 Interface de administração	52
Figura 3-13 Cadastro de <i>ThinkLets</i>	52
Figura 3-14 Cadastro de padrões colaborativos e tipos de resultado	53
Figura 3-15 Cadastro de compatibilidade entre <i>ThinkLets</i>	54
Figura 4-1 Arquitetura do <i>AgendaBuilder</i>	56
Figura 4-2 Template de um arquivo de regras no Drools	59

Lista de Tabelas

Tabela 2-1 Exemplo de mapa de escolha (adaptado de KOLFSCHOTEN <i>et al</i> , 2007).	13
Tabela 4-1 Template para tabela de análise quantitativa.....	64
Tabela 5-1 Critérios para SevenUp	70
Tabela 5-2 Critérios para SevenUp	70

Lista de Abreviaturas

DM – DynaMeetings

EC – Engenharia de Colaboração

CSCW – Trabalho Cooperativo Suportado por Computador (*Computer Supported Cooperative Work*)

SSDG – Sistemas de Suporte à Decisão em Grupo

GGG – *Group Support Systems*

GS – *GroupSystems*

MW – *Meeting Works*

1 Introdução

Reuniões consistem em um processo fundamental e constante em qualquer organização, sendo um importante veículo para contato pessoal. No entanto, reuniões podem representar um grande desperdício de tempo e recursos caso não sejam previamente planejadas e executadas conforme o plano.

Diversos fatores podem comprometer a produtividade em reuniões (VREEDE, G.J. *et al*, 2003). Por isso, algumas técnicas têm sido utilizadas com o objetivo de aumentar a produtividade de reuniões (KOLFSCHOTEN, G.L. *et al*, 2004). Uma estratégia consiste no emprego de facilitadores, ou seja, profissionais que coordenam reuniões e auxiliam o grupo a alcançar seus objetivos. Facilitadores atuam antes, durante e após a realização de reuniões.

Nos últimos anos tem se tornado mais comum o emprego destes profissionais. Diversas ferramentas foram desenvolvidas visando dar suporte à realização de reuniões, focando nas necessidades das pessoas envolvidas. No entanto, poucas propõem suporte ao facilitador, tornando o seu trabalho mais eficiente e aumentando a chance de sucesso da reunião.

O papel dos facilitadores consiste em projetar o processo de reunião e suas atividades de modo a garantir que os objetivos desejados sejam alcançados. Para isso podem ser utilizados padrões de dinâmicas de grupos pré-definidos, como os *ThinkLets* (VREEDE, G.J. *et al*, 2001). Os facilitadores também conduzem a reunião, garantindo que o grupo está procedendo conforme o esperado, as questões necessárias estão sendo abordadas e o planejamento está sendo seguido. Assim, um dos papéis do facilitador é de monitorar a reunião e corrigir qualquer desvio na dinâmica do grupo. Se o facilitador notar que o grupo está se distanciando de seus objetivos, ele deve tentar recolocar o grupo de volta no caminho correto.

Segundo a Engenharia de Colaboração (KOLFSCHOTEN, G. *et al*, 2007), a atividade de elaboração do processo de reunião poderá ser dividida entre diversas etapas (maiores detalhes no Capítulo 2). Entre elas, estão a decomposição do processo de reunião em uma sequência de passos ou atividades e, posteriormente, a escolha de um *ThinkLet* para cada uma dessas atividades.

Assim, a atividade de planejamento de uma reunião se resume em um processo de escolha a partir de um catálogo disponível de *ThinkLets*, onde estão documentadas as descrições, configurações e passos necessários para a execução de cada um deles, bem como orientações sobre a aplicabilidade ou não de acordo com a situação. Apesar de simplificar bastante o processo de elaboração de um processo colaborativo, essa abordagem ainda impõe um desafio considerável, devido à extensão do catálogo e suas múltiplas possibilidades.

Atualmente, existem em torno de 40 *ThinkLets* conhecidos e catalogados. Para a utilização dessa abordagem, o facilitador deverá conhecer em detalhes as características de cada um desses *ThinkLets*, de forma a ser capaz de elaborar um processo consistente, considerando plenamente todas as possibilidades que eles oferecem. Caso o facilitador não possua esse conhecimento necessário, ele não será capaz de fazer escolhas apropriadas.

Para exemplificar, imaginemos uma situação em que um facilitador deva elaborar uma atividade para um grupo de pessoas levantar, analisar e selecionar idéias para melhoria de um processo produtivo de uma dada empresa. Como uma primeira etapa, será necessário fazer com que o grupo passe por um *brainstorming* de forma a gerar um conjunto inicial de idéias. Para atividades de *brainstorming*, podem ser aplicados diversos *ThinkLets*: *FreeBrainstorm*, *OnePage*, *ComparativeBrainstorm*, *LeafHopper* e *DealersChoice*. Como realizar uma escolha? Assim, o facilitador precisará ter uma certa experiência para poder começar a fazer escolhas e, ainda assim, dada a extensão de possibilidades, os resultados podem ser comprometidos caso o facilitador não disponha de nenhum suporte.

Neste trabalho, nos dedicamos a analisar e propor uma solução computacional que dê suporte ao facilitador nessa tarefa de estruturação inicial do processo de reunião. Para isso, iremos nos basear nos conceitos da Engenharia de Colaboração e *ThinkLets*.

1.1 Contextualização

Este trabalho está inserido no projeto *DynaMeetings*, cujo objetivo é a criação de um sistema (de mesmo nome) que ofereça suporte para um facilitador na elaboração e condução de um processo colaborativo. O sistema será composto por diversos módulos e ferramentas, cada uma com a finalidade de prestar suporte em um determinado aspecto do trabalho do facilitador.

Um módulo visa fornecer suporte ao facilitador durante a execução de cada atividade de reunião do grupo. Este módulo possui funcionalidades que permitirão acompanhar o andamento da atividade em relação aos seus objetivos previamente definidos e a evolução do grupo. O acompanhamento é feito através de diversas informações estatísticas fornecidas pela ferramenta, como, por exemplo, o nível de conflito entre os participantes e a velocidade de geração de idéias. Esse módulo do sistema é descrito em detalhes em (MARQUES, L.M., 2009).

Este trabalho foca em um módulo que permite ao facilitador organizar e montar um processo colaborativo, através da definição de uma sequência de atividades a serem realizadas por um grupo. Cada atividade do processo colaborativo pode ser executada através da utilização de um *ThinkLet* (que será discutido posteriormente). A ferramenta auxilia o facilitador na definição apropriada de cada atividade, dados os objetivos a serem atingidos e um conjunto de informações contextuais, como por exemplo, o tamanho do grupo.

Outros dois módulos propostos visam: 1) auxiliar o facilitador na condução do processo, permitindo a ele realizar alterações em seu planejamento inicial, de acordo

com a necessidade; 2) fornecer recomendações sobre participantes para uma reunião através da análise de redes sociais.

1.2 Organização da Dissertação

O restante desta dissertação está dividido nos seguintes capítulos:

No Capítulo 2, apresentamos um estudo sobre Trabalho Cooperativo Suportado por Computador e, mais especificamente, conceitos e técnicas relacionadas a suporte a reuniões. Tentamos não apenas definir os conceitos utilizados por toda a dissertação, mas também contextualizar um cenário onde nossa abordagem está inserida. Também são apresentadas algumas abordagens para a construção de ferramentas utilizadas para dar apoio a atividades colaborativas descritas na literatura.

O Capítulo 3 apresenta a abordagem proposta para a construção de uma ferramenta computacional com o objetivo de apoiar um facilitador em suas atividades de especificação de um processo de reunião. Em seguida, é apresentado o funcionamento da ferramenta implementada, contextualizando através de imagens de suas principais interfaces com o usuário.

O Capítulo 4 apresenta a implementação da ferramenta proposta, mostrando a arquitetura adotada, bem como componentes específicos utilizados, como JGraph e Drools. Em seguida é apresentada também um plano para um experimento que visa validar os conceitos aplicados pela ferramenta implementada como parte desta proposta.

O Capítulo 5 encerra a dissertação apresentando as suas principais contribuições e limitações, listando ainda os possíveis trabalhos futuros.

2 Revisão da Literatura

Este capítulo apresenta um estudo sobre os conceitos que serviram de base para o trabalho proposto. Inicialmente, é feita uma breve análise dos conceitos de Trabalho Cooperativo Suportado por Computador. Em seguida, são abordados os aspectos específicos relacionados a apoio a reunião, facilitação para, finalmente, serem tratadas as propostas de Engenharia de Colaboração. Ao final, é feita uma breve análise de ferramentas correlatas.

2.1 Trabalho Cooperativo Suportado por Computador

Nesta seção serão abordados os conceitos básicos relacionados à área de pesquisa de trabalho colaborativo suportado por computador, contexto maior no qual este trabalho se insere. O termo CSCW (do inglês: *Computer Supported Cooperative Work*) surgiu pela primeira vez em 1984, em um *workshop* onde se reuniram diversas pessoas interessadas no papel crescente da tecnologia em ambientes de trabalho (GRUDIN, J., 1994). Desde então, diversos pesquisadores e desenvolvedores se dedicaram a explorar este tema. Trata-se de uma área de pesquisa multidisciplinar que, frequentemente, abrange conceitos de psicologia, sociologia e outros, além de ciência da computação.

Apesar de o termo ter surgido oficialmente na década de 80, alguns conceitos subjacentes a essa área já datavam da década de 60, como no trabalho desenvolvido por ENGELBART (1962), que já buscava implementar edição compartilhada de documentos e videoconferência. No entanto, a área passou a ganhar mais destaque conforme as tecnologias de comunicação evoluíram, sobretudo com a revolução trazida pela *Internet* (BEAUDOUIN-LAFON, M., 1999). Com o surgimento da rede mundial de computadores, criou-se um terreno propício para as tecnologias de *groupware*.

Groupware é o lado aplicado dos conceitos do CSCW (EHRLICH, K, 1999). As aplicações e ferramentas de *groupware* visam dar suporte ao trabalho de grupos de pessoas envolvidas em tarefas ou objetivos comuns, frequentemente provendo como interface um ambiente compartilhado. As ferramentas de *groupware* podem funcionar tanto de forma síncrona (quando os usuários interagem no mesmo espaço de tempo), quanto assíncrona (quando a interação se dá em diferentes momentos no tempo).

Para a análise e categorização dos tipos de sistemas de *groupware* que visam dar suporte ao trabalho cooperativo é comumente utilizada uma matriz, introduzida por JOHANSEN (1988). A matriz é construída levando-se em consideração duas dimensões: tempo e espaço físico. Ou seja, se o grupo está fisicamente em um mesmo local ou distribuído; e se o grupo irá trabalhar síncrona ou assincronamente.

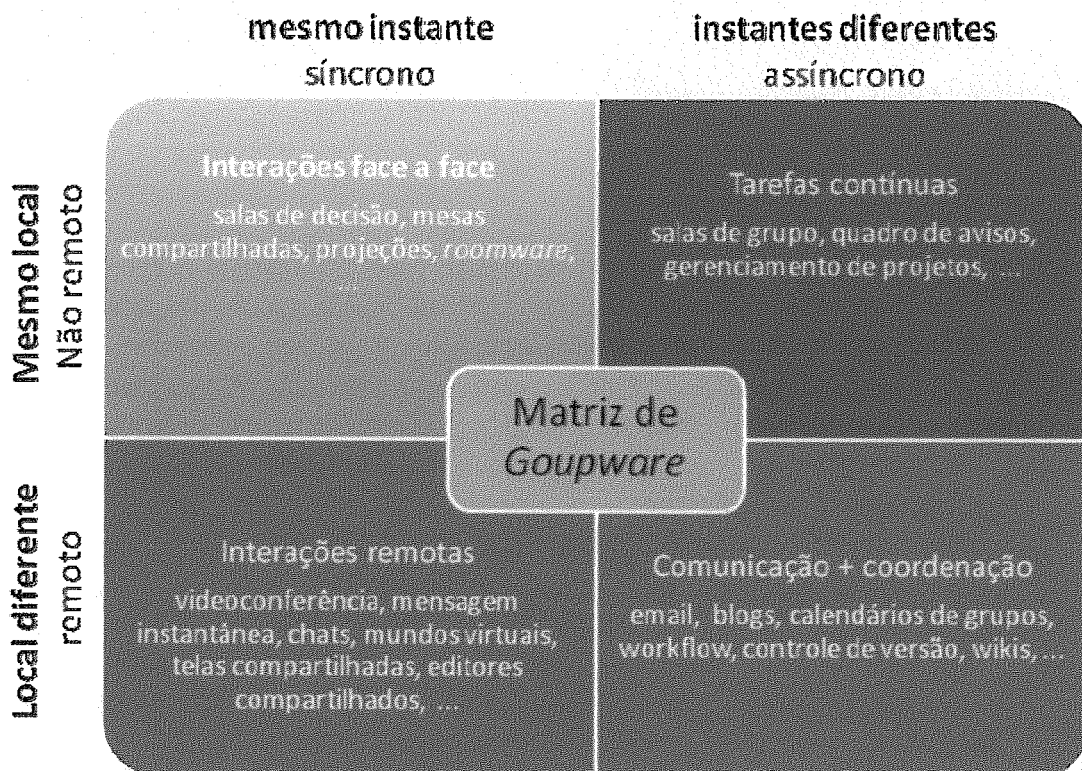


Figura 2-1 Matriz de contextualização de CSCW (adaptado de BURSTEIN & HOLSAPPLE, 2008)

Como podemos ver em (Figura 2-1), quando o grupo estiver em um mesmo local, ao mesmo tempo (interações face a face), tecnologias como quadros digitais, salas de decisão, mesa compartilhada ou salas eletrônicas (*roomware*) podem ser utilizadas. Para interações ocorrendo em um mesmo tempo, mas distribuídas fisicamente (interações remotas), tecnologias como videoconferência e ferramentas de mensagem instantânea, *chats*, telas compartilhadas e editores compartilhados podem dar suporte ao trabalho cooperativo. Para comunicação e coordenação de tarefas assíncronas em diferentes localidades, um exemplo clássico de aplicação é o correio eletrônico e outras aplicações possíveis são *blogs*, calendários compartilhados, *workflows*, *wiki*, entre outros. Para tarefas contínuas que se dão em uma mesma localidade mas em momentos distintos, temos como exemplos salas de grupos, quadros de avisos e gerenciamento de projetos, entre outros.

2.2 Apoio a reunião para tomada de decisão

Uma das principais ferramentas para trabalho colaborativo são os chamados SSDGs (Sistemas de Suporte a Decisão em Grupo). Essas ferramentas são comumente empregadas para apoiar atividades de grupos em processos de tomada de decisão, onde estão envolvidas etapas de discussão e levantamento de idéias, avaliações e construção de soluções. Os SSDGs provém diretamente dos conceitos de CSCW e, geralmente, se apóiam nas tecnologias de *groupware* no que diz respeito ao suporte de comunicação do grupo. Os SSDGs, diferentemente de outras tecnologias puramente de *groupware*, são focados no suporte a tarefa em si, não apenas no suporte a comunicação.

O principal SSDGs existente é o GGS (Group Support Systems) e tem sido largamente utilizado. Frequentemente os dois termos se confundem tal o reconhecimento atingido pelo GSS na comunidade.

Nas últimas décadas, pesquisas têm mostrado que, nas circunstâncias apropriadas, a adoção dos SSDGs pode poupar em torno de 50% do esforço de um grupo, reduzindo tempo e custo de projetos (GROHOWSKI, R., 1990). Através dos

SSDGs um grupo pode produzir muito mais idéias em uma reunião do que através dos métodos tradicionais baseados em papel (GALLUPE, R.B., 1992).

No entanto, os resultados práticos em campo tem se mostrado conflitantes e confusos. Enquanto alguns estudos refletem positivamente a utilização dos SSDGs, outros não (VREEDE, G.J., 2003). A principal causa dos resultados conflitantes, apesar de as pesquisas indicarem que a adoção dos SSDGs leva a maior produtividade, está no fato de faltar clareza sobre como transferir adequadamente o conhecimento de pesquisas para o mundo organizacional (BRIGGS, R.O., 2001).

Um dos recursos críticos para um SSDGs é o facilitador. Seu papel envolve as atividades de preparação (tanto no planejamento das atividades quanto dos recursos necessários) da reunião, a condução da execução da reunião, mantendo-a sempre de acordo com o planejado.

2.3 Facilitação

O papel do facilitador tem o objetivo de conduzir um grupo através de um processo de reunião, auxiliando-o a compreender o que deve ser feito e atingir seus objetivos (VILLER, 1991). Esse papel é consideravelmente importante para a qualidade de uma reunião eletrônica, segundo alguns estudos (CLAWSON et al., 1993). Os benefícios da utilização de um facilitador em um grupo são reconhecidos tanto em reuniões face a face quanto em reuniões à distância (CLAWSON & BOSTROM, 1993).

O papel do facilitador pode ser desempenhado tanto por uma pessoa integrante do grupo, quanto por uma pessoa externa, às vezes até mesmo externa à organização. No entanto, essa pessoa deve possuir uma boa aceitação pelo grupo, pois disso depende o sucesso do processo. Ele deve ser o responsável pela resolução de conflitos, visando evitar resultados indesejáveis em uma reunião (VREEDE et al., 2003).

O facilitador é responsável por todas as etapas de uma reunião, desde o seu planejamento, a condução de sua execução até as atividades posteriores a sua realização (CLAWSON & BOSTROM. 1993), (ANTUNES & HO, 2001). Na etapa de planejamento, ele deve definir o propósito da reunião, seu formato, quais são os participantes necessários, a pauta, ou seja, os tópicos a serem discutidos e o tempo dedicado a cada um. Ele deve também, após o planejamento, garantir que todos os participantes recebam a pauta da reunião para análise prévia. Deve também, como parte do planejamento, garantir quaisquer recursos necessários para sua realização, como sala e equipamentos.

Durante a etapa de condução da reunião, o facilitador possui as seguintes tarefas/atividades:

- Introduzir a tecnologia utilizada (se houver) e garantir que todos os participantes tenham um correto entendimento sobre ela, respondendo sempre que necessário as dúvidas surgidas;
- Deve cumprir a pauta da reunião, buscando sempre manter o foco nos pontos que devem ser discutidos;
- Faz a mediação da discussão dos tópicos propostos na pauta e controla o tempo gasto em cada um;
- Deve incentivar a colaboração por parte do grupo, assim como tentar criar uma sinergia entre os participantes;
- Deve se manter neutro, sem tomar posição em relação ao tema, discussões e afins;
- Deve identificar e mediar problemas ou conflitos o mais rápido possível, garantindo que o grupo não desvie do objetivo original da reunião.

Após a realização da reunião, o facilitador é o responsável por consolidar toda a discussão ocorrida e gerar a ata (enviando a todos os participantes para devida aprovação). Ele pode ser o responsável também por acompanhar o andamento e execução das ações propostas e definidas pela reunião.

Pode-se então concluir que o sucesso ou insucesso de uma reunião está bastante relacionado ao trabalho do facilitador. É importante lembrar que conduzir uma reunião requer mais que apenas o conhecimento do conteúdo e do processo: requer um entendimento dos problemas individuais, sociais e políticos que podem estar presentes ou vir a ocorrer (MACAULAY, 2005).

2.4 Engenharia de Colaboração

A Engenharia de Colaboração é uma abordagem para o *design* e implantação de processos colaborativos que possam ser executados pelos seus próprios praticantes em atividades recorrentes e de alto valor agregado (KOLFSCHOTEN, G. *et al*). Um engenheiro de colaboração desenvolve os processos colaborativos e os transfere para os praticantes de uma organização.

A Engenharia de Colaboração frequentemente emprega tecnologias de SSDGs visando maior eficiência dos processos colaborativos em organizações. Para que tais tecnologias tenham um retorno adequado é necessário que sejam corretamente aplicadas e moderadas por um profissional em facilitação experiente. Esse profissional precisa ter conhecimentos suficientes na condução dos processos colaborativos para que sua aplicação possa se dar com sucesso em uma organização.

Esses profissionais geralmente são muito caros e por suas habilidades são promovidos rapidamente. A Engenharia de Colaboração tenta contornar esse problema, buscando reduzir a dependência em relação a esses profissionais. De acordo com BRIGGS et al. (2003), Engenharia de Colaboração é uma abordagem para o planejamento de atividades de trabalho colaborativo com tarefas recorrentes, com alto valor agregado, permitindo a transferência desses planos para que sejam conduzidos pelos próprios praticantes em uma organização, sem a necessidade de facilitadores profissionais externos.

Uma prática de trabalho pode ser entendida como um conjunto de tarefas que são sempre repetidas por uma atividade da organização (VREEDE & BRIGGS, 2005) (KOLFSCHOTEN & VREEDE, 2007). Uma tarefa é dita:

- Colaborativa: se sua finalização com sucesso depende do esforço em conjunto dos indivíduos.
- De alto valor agregado: se ela tem uma grande importância para a organização, ou se ela evita riscos ou uma grande perda ao ser completada com sucesso.
- Recorrente: se deve ser conduzida repetidamente e um processo similar pode ser utilizado sempre.

A criação de um processo colaborativo frequentemente envolve diversos passos, como: identificação do problema, análise, busca e avaliação de alternativas, escolha e implementação. Na Engenharia de Colaboração, são utilizados padrões de *design* chamados *ThinkLets*, que consistem de descrições de soluções conhecidas e reutilizáveis. O propósito dessa abordagem é, ao invés de consumir o tempo de planejamento buscando e avaliando todas alternativas existentes, reutilizar essas soluções já conhecidas. Essa abordagem é semelhante a adoção de *Design Patterns*, comumente utilizada na Engenharia de Software. Assim, a atividade de planejamento se resume a simplesmente escolha. Na Figura 2-2 podemos ver a sequência de passos proposta pela Engenharia de Colaboração.

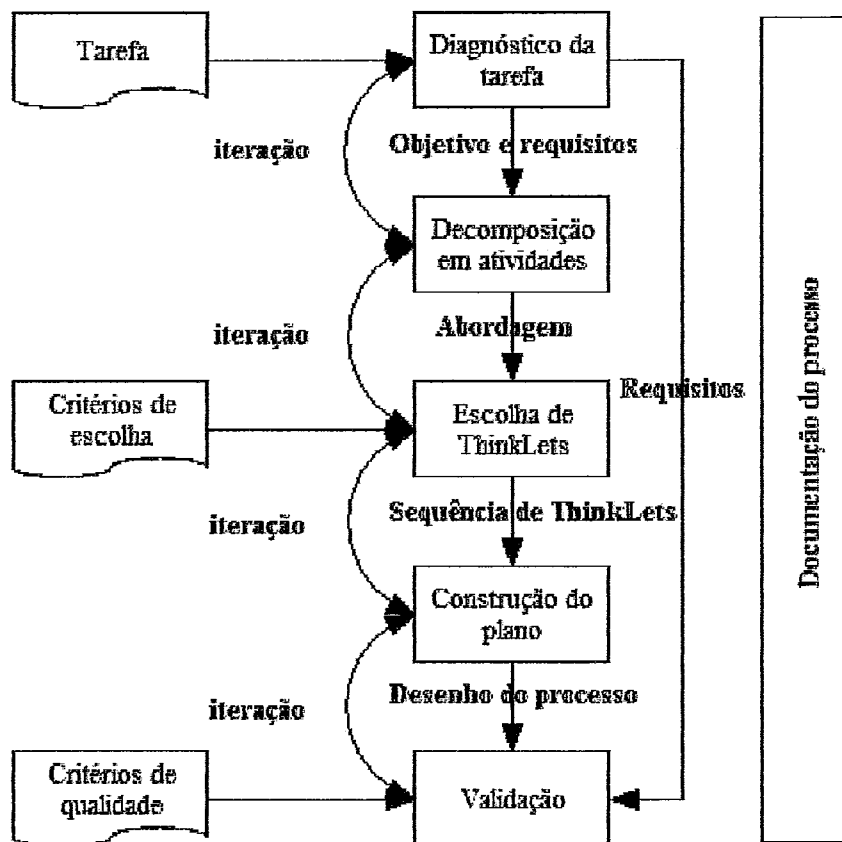


Figura 2-2 Visão geral da Engenharia de Colaboração (adaptado de KOLFSCHOTEN, G. et al, 2007)

Segundo esta abordagem, a atividade de planejamento pode ser decomposta nas seguintes atividades:

- **Diagnóstico da tarefa:** Nesta atividade, o facilitador identifica as premissas, os requisitos da tarefa, as restrições, os *stakeholders* envolvidos, os recursos envolvidos e os participantes. Nesta atividade são documentados:
 - a. Objetivos e entregas, a partir da análise da tarefa;
 - b. Grupo, interesses, papéis e necessidades, a partir da análise dos *stakeholders*;

- c. Tempo, conhecimento e esforço, a partir da análise dos recursos.
- d. Habilidades, experiência, personalidade, especialidade, a partir da análise dos participantes.

- **Decomposição da tarefa em atividades:** Nesta atividade, tendo entendido claramente os requisitos e objetivos, o facilitador irá começar a traçar o desenho do processo, decompondo a tarefa em atividades.
- **Escolha de ThinkLet:** Após a decomposição da tarefa, as atividades podem ser mapeadas em *ThinkLets*. A tarefa de escolha é complexa e deve levar em consideração diversos fatores, que podem tornar a atividade bastante complexa. Para auxiliar, foram propostos por Kolfshoten *et al* (2007), dois mapas (Tabela 2-1).

Tabela 2-1 Exemplo de mapa de escolha (adaptado de KOLFSCHOTEN *et al*, 2007)

Padrão: Generate	Resultado: Input											
FreeBrainstorm	FreeBrainstorm	Combinação entre Thinklet: Boa = 2 Complicada = 1 Impossível = 0	FreeBrainstorm	OnePage	Comparative	LeafHopper	DealersChoice	PlusMinus	TopFive	BranchBuilder	TheLobbyist	
OnePage	OnePage		Ponto de partida	2	2	2	2	2	0	0	2	0
Comparative	Comparative		FreeBrainstorm	0	0	2	1	0	1	0	1	2
LeafHopper	LeafHopper		OnePage	0	0	1	1	1	2	0	2	2
DealersChoice	DealersChoice		Comparative	0	0	0	1	1	1	0	1	2
PlusMinus	PlusMinus		LeafHopper	0	0	0	2	2	2	0	1	2
TopFive	TopFive		DealersChoice	0	0	0	2	2	2	0	1	1
BranchBuilder	BranchBuilder		PlusMinus	0	0	0	1	1	1	0	1	2
TheLobbyist	TheLobbyist		TopFive	0	0	0	0	0	1	1	1	2
DimSum	DimSum		BranchBuilder	0	0	0	0	0	1	0	1	2
PointCounterP.	PointCounterP.	TheLobbyist	0	0	0	0	0	0	0	0	0	
	StrawPoll											
	MultiCriteria											
	CheckMark											
	StakeHolderPoll											
	BucketVote											

Esses mapeamentos serão abordados com maiores detalhes no capítulo seguinte. Basicamente, ele apresenta uma forma de identificar possibilidades de utilização dos *ThinkLets* a partir de:

1. Padrão colaborativo: padrão de colaboração caracteriza a forma como um grupo interage em uma atividade de forma a atingir seus objetivos.
 2. Tipo de resultado: tipos de resultados obtidos por uma atividade colaborativa.
 3. Combinação entre *ThinkLets*: grau de qualidade da combinação de dois *ThinkLets*, quando a saída de um é utilizada como entrada para o próximo. Na Tabela 2-1, os *ThinkLets* dispostos verticalmente utilizam como entrada as saídas dos *ThinkLets* dispostos horizontalmente.
- **Definição da agenda:** Além da definição da sequência de *ThinkLets*, o processo colaborativo deve incluir também outros passos, como aberturas, atividades de quebra de gelo, introdução na tecnologia utilizada, exercícios de aquecimento etc. Nesse passo, o facilitador deve finalizar o processo introduzindo quaisquer atividades auxiliares conforme a necessidade e deve determinar todas as informações necessárias para a execução de cada *ThinkLet*, como a descrição da atividade, o que se pretende atingir com ela, as questões para o grupo abordar, tempo estimado para cada atividade.
 - **Validação:** Etapa de validação do processo colaborativo construído.
 - **Documentação:** Atividade que visa registrar todas as informações relevantes para o participante sobre o processo colaborativo.

2.5 ThinkLets

Um ThinkLet é a menor unidade de capital intelectual requerido para criar um padrão repetível e previsível de pensamento entre as pessoas que buscam um objetivo em comum (BRIGGS et al., 2001). Ele é composto por três elementos:

- Ferramenta: uma tecnologia de *hardware* e/ou *software* necessária para criar um padrão de pensamento.
- Configuração: a especificação de como a ferramenta (*hardware* e *software*) devem ser configurados para criar o padrão de interação entre os indivíduos.
- Script: uma sequência de eventos e instruções dadas para o grupo criar o padrão de pensamento/interação.

ThinkLets são geralmente utilizados para incentivar a colaboração entre as pessoas de um grupo (KOLFSCHOTEN et al., 2004). Os *ThinkLets* são organizados em seis classes de acordo com o padrão de pensamento associado: divergência, convergência, organização, esclarecimento, criação de consenso e avaliação. Cada *ThinkLet* pode ser classificado em uma ou mais dessas classes, que são diretamente associadas às dinâmicas de grupo em uma reunião.

Os *Thinklets* foram inicialmente formalizados, extensamente documentados e testados por Briggs *et al* (2001). No entanto, antes disso eles já eram presentes na comunidade de pesquisa como conhecimento tácito. Diversos pesquisadores e facilitadores entrevistados já os reconheciam como padrões já aplicados, apesar de não possuírem nomes definidos para eles.

A documentação disponível atualmente descreve em torno de 40 *Thinklets*, catalogados com as seguintes informações:

- Um nome que evoca as dinâmicas criadas por ele;
- Critérios para decidir quando aplicar e quando não aplicar um *ThinkLet*;
- A ferramenta, configuração e *script* necessária para aplicá-lo;
- Percepções e orientações obtidas a partir de conhecimento empírico;
- Casos de sucesso para ajudar a clarear as circunstâncias sob as quais cada *ThinkLet* pode ser útil;

- Uma explicação do nome, de forma a tornar fácil a memorização tanto do nome quanto do *ThinkLet*.

ThinkLets foram concebidos como um modo de ajudar o facilitador no planejamento e condução das reuniões. O processo de criação de uma reunião aplica em conjunto determinados *ThinkLets* para alcançar os resultados pretendidos, e conduzir a reunião é uma questão de conduzir o grupo através dos *ThinkLets*, utilizando os scripts fornecidos em suas definições. Assim, a tarefa de definição de um processo colaborativo se resume em, a partir da situação e do objetivo a ser atingido, selecionar de um catálogo os *ThinkLets* mais apropriados para serem utilizados. Para isso, o facilitador deve conhecer razoavelmente as características de cada um e seus critérios de aplicabilidade.

2.6 Sistemas correlatos

Nesta seção serão apresentadas algumas ferramentas que visam dar suporte a processos de decisão em grupos. Esses sistemas são conhecidos como SSDGs (Sistemas de Suporte a Decisão em Grupo). Serão analisados em seguida os dois SSDGs mais conhecidos atualmente.

2.6.1 GroupSystems

O *GroupSystems* é certamente o sistema de suporte a reunião e *brainstorming* mais difundido e também o de maior sucesso comercial. Foi originalmente desenvolvido por uma equipe de pesquisadores da Universidade do Arizona, e utilizado em inúmeros experimentos. É largamente difundido, sendo utilizado por diversas organizações governamentais, comerciais ou educacionais. Entre elas, podemos citar o Pentágono, o Exército Norte-Americano, IBM, Intel, Gartner Group, Universidade do Colorado e Universidade de Nebraska.

No *GroupSystems* o papel do facilitador é bem delineado. Cabe ao facilitador planejar a agenda, escolher as ferramentas mais adequadas, estabelecer limites de tempo, selecionar e convidar os participantes certos para a reunião. No decorrer da reunião, cabe ao facilitador iniciar, gerir e finalizar todas as atividades previstas. É ele que mantém o controle sobre todo o sistema, podendo inclusive determinar o conteúdo visualizado pelos participantes, de forma a influenciar o andamento da reunião.

O sistema disponibiliza várias ferramentas para o desenvolvimento das atividades do grupo. Segue uma breve análise de cada uma delas:

- **Electronic Brainstorming (EB)**: Utilizada nas tarefas de geração livre de ideias, baseando-se no modelo de *brainstorming* tradicional. Permite a criação de páginas de discussão eletrônicas, que vão rodando pelos vários participantes, podendo estes acrescentar novas ideias à lista, ou fazer comentários às ideias existentes.
- **Topic Commenter (TC)**: Esta ferramenta é indicada para tarefas de geração e organização de ideias sobre tópicos específicos. Permite criar uma página de discussão relativa a cada tópico, onde os participantes podem acrescentar as suas ideias e comentários.
- **Categorizer (CAT)**: Esta ferramenta é indicada para tarefas de classificação de ideias, e também para tarefas de geração mais estruturadas. Permite a criação de categorias e a classificação de ideias em tempo real de acordo com essas categorias.
- **Group Outliner (GO)**: Esta ferramenta é indicada para tarefas de geração e organização de ideias segundo uma estrutura hierárquica. Permite a criação de tópicos e sub-tópicos.
- **Vote (VOT)**: Esta ferramenta é indicada para tarefas de avaliação e de tomada de decisão. Trata-se de uma ferramenta bastante flexível, permitindo diversos métodos de avaliação.

- **Opinion Meter (OM)**: Esta ferramenta é indicada para tarefas de decisão rápida sobre tópicos simples, fornecendo um conjunto reduzido de métodos de votação.
- **Survey (SV)**: Esta ferramenta é indicada para a realização de pesquisa. Permite três formas de interação: (1) como uma atividade integrada numa agenda normal do GS, (2) através da Internet, e (3) a gravação da pesquisa separadamente, de forma a que possa ser distribuído pelos participantes.
- **Cross Analysis (CA)**: Esta ferramenta é indicada para tarefas de avaliação e seleção de critérios múltiplos.

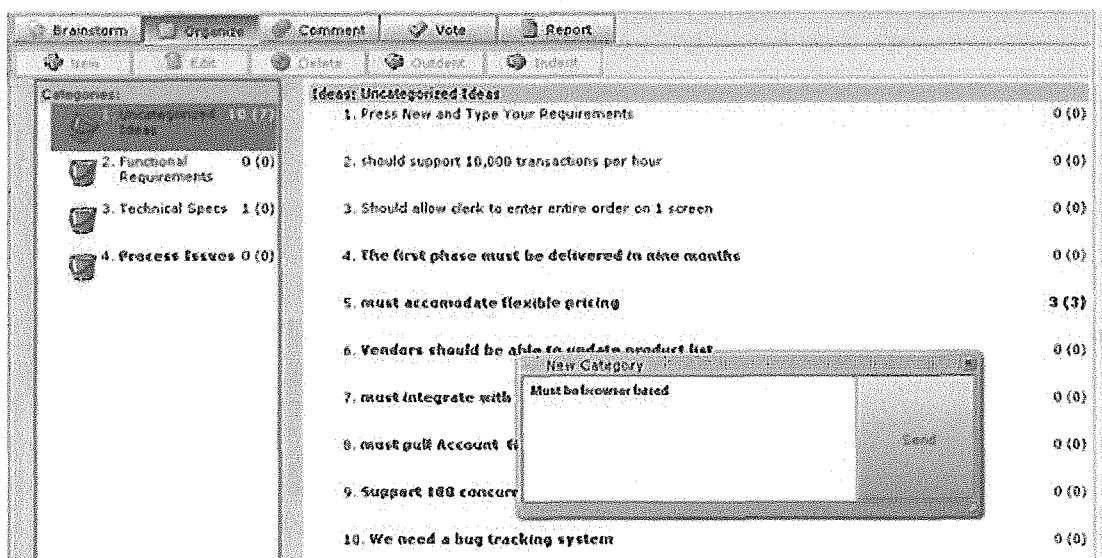


Figura 2-3 Categorização no GroupSystems

Podemos ver em (Figura 2-3) um exemplo de tela de categorização no *GroupSystems*. Pode-se observar que o participante está criando uma nova categoria na janela em foco. No topo, são mostradas as fases da reunião, e é possível notar que a reunião está na fase de organização, que é a aba em destaque. Na esquerda, ficam as categorias já criadas, enquanto na direita estão as ideias geradas pelo grupo.

Outras características bastante úteis no *GroupSystems* são a capacidade de transferência de itens ou de resultados de uma atividade para outra (por exemplo, a lista de ideias geradas com o EB pode ser transferida para uma tarefa de organização com o

GO), a produção de relatórios e registros sobre todas as atividades executadas, e o armazenamento automático de todas as sessões decorridas.

2.6.2 Meeting Works (MW)

O Meeting Works é uma ferramenta de suporte a decisão em grupos que dá suporte tanto a equipes locais quanto a equipes remotas. Entrevistas realizadas entre seus usuários indicam que mais de 80% das pessoas acreditam que sua utilização poupou tempo e dinheiro e que o *software* certamente ajuda a estabelecer objetivos mais claros (HO, 1999).

A ferramenta dá suporte a reuniões presenciais e demanda a existência de um facilitador para coordenar reunião e um *chauffeur*, responsável pela manipulação da ferramenta eletrônica. Na maioria dos casos, o facilitador e o *chauffeur* são a mesma pessoa, mas o papel do *chauffeur* encontra-se limitado apenas à realização da agenda e à sua execução, interagindo com o grupo apenas durante a reunião. É a ele que cabe também resolver quaisquer problemas técnicos que ocorram durante a reunião. O papel do facilitador no MW começa na fase de planejamento da reunião, começando por tentar traduzir o problema numa agenda específica. Mais tarde, o *chauffeur* irá associar as tarefas com ferramentas do MW para desenvolver uma agenda eletrônica. Durante a reunião, o facilitador apresenta a agenda, guia o grupo através de tópicos e tarefas, coloca questões e interpreta os resultados. É também responsável por manter o grupo focado, reforçar regras estabelecidas, construir um espírito de equipe, e pedir opiniões de reforço. Após a reunião, deverá assegurar que todos os resultados foram atingidos. O papel do facilitador no MW está assim associado às necessidades do grupo antes, durante e depois de uma reunião.

As idéias são enviadas pelos usuários para uma área pública comum, para serem discutidas e avaliadas em uma etapa posterior. As contribuições são enviadas anonimamente para a área comum, o que favorece o aumento da contribuição, pois evita síndromes comuns, como inibição. A participação igualitária, independente de posição

ou *expertise* dos participantes, é garantida pela ferramenta. Para a decisão final, ela dá direito de um voto para participante, tendo todos o mesmo peso.

O sistema consiste numa variedade de ferramentas que assistem o grupo a discutir, gerar, organizar, avaliar, e analisar ideias. As ferramentas fornecidas pelo MW são as seguintes:

1. Agenda Planner – Permite criar, editar, e manter agendas de reuniões (Figura 2-4).
2. Generate – Permite a geração de listas de ideias ou comentários criados anonimamente pelos participantes.
3. Organize – Permite apresentar uma lista de ideias para discussão, com o objetivo de organizá-las e estruturá-las. Simplifica o processo quando se tem que lidar com muitas informações.
4. Evaluate – Permite saber as opiniões de todos de uma forma anônima, e resumir os resultados através de gráficos.
5. Multiple Criteria Analysis – Permite ao grupo compreender o peso e a importância de vários fatores envolvidos numa decisão ou votação.
6. Cross Impact Analysis – Permite comparar como uma possível solução poderá afetar positivamente ou negativamente outros aspectos do problema, tornando fácil a identificação de conflitos.
7. File Editor – Permite gerenciar arquivos de texto com o conteúdo da reunião.
8. Timer – Ajuda a manter as tarefas dentro de um determinado período de tempo planejado.

Manual	DESCRIPTIONS/INTRODUCTIONS
External	Run slide show using Powerpoint
Organize	Open parking lot
Manual	PROBLEMS WITH TRADITIONAL MEETINGS
Generate	List problems encountered in traditional meetings
↓ Organize	Edit problem list
↓ Evaluate	Rate problems for effect on productivity
↓ File Editor	Reduce list of problems
Timer	TIMER
Manual	SOLUTIONS FOR TRADITIONAL MEETINGS
Organize	Electronic meeting benefits tutorial
↓ Generate	List ways electronically-supported meetings can help solve problems
Organize	Edit list of solutions
Manual	ELECTRONIC MEETINGS
Crash Impact	Analyze electronic meeting support for effectiveness & efficiency
Multi Crit.	Choose sites for electronic meeting rooms
Manual	CONNECT FEATURES--GROUP DRAW, BOOKSHELF
Manual	WRAP-UP
Manual	Cruise the parking lot
External	Print final report using Word
Manual	Send final HTML report

Figura 2-4 Agenda Planner, MeetingWorks

2.6.3 Ferramentas de apoio ao facilitador

Os sistemas de apoio a reunião atuais oferecem pouco ou quase nenhum suporte à facilitação. Os dois sistemas descritos na seção anterior oferecem apenas suporte em termos de planejamento da agenda. O *GroupSystems* permite organizar as sessões por pastas, de forma que o facilitador possa a qualquer momento ter acesso a todas as reuniões realizadas com o sistema. Cada sessão corresponde a uma agenda. Inicialmente a agenda encontra-se vazia, cabendo ao facilitador selecionar a sequência de atividades que melhor se adequa ao problema em questão. Ao adicionar uma nova atividade à agenda, o sistema mostra ao facilitador uma lista de todas as ferramentas disponíveis, para que ele possa escolher a mais adequada. Para cada atividade, o facilitador deve definir o seu tópico e, caso seja necessário, um limite de tempo.

O *MeetingWorks* possui uma ferramenta específica para o planejamento da agenda, designada por *Agenda Planner*. Ao iniciar uma nova agenda, o facilitador pode optar entre iniciar uma agenda vazia, onde irá adicionando atividades novas manualmente, ou escolher uma das agendas pré-definidas disponibilizadas pelo *MeetingWorks*. Se optar por planejar a agenda manualmente, o processo de seleção é semelhante ao do *GroupSystems*, tendo o facilitador que definir para cada passo da agenda qual o seu tópico, e caso seja necessário, um conjunto de instruções aos participantes. No caso de optar por escolher uma das agendas disponíveis, o facilitador poderá selecionar entre cinco tipos de agendas possíveis, que dependem do objetivo da reunião:

- Melhoria do desempenho do grupo.
- Avaliação final do projeto.
- Avaliação dos estados intermediários do projeto (checkpoints).
- Início de um novo projeto.
- Sessão de planejamento estratégico.

Existem ainda diversos sistemas que oferecem suporte ao agendamento de reuniões. Correntemente são bastante utilizados o Lotus Notes e o Microsoft Outlook. Todos estes sistemas apresentam características comuns, oferecendo aos utilizadores um calendário que permite que cada utilizador visualize a lista dos seus compromissos, convide outros utilizadores para reuniões e verifique se estes se encontram disponíveis na data pretendida.

Os sistemas descritos anteriormente oferecem um suporte limitado ao facilitador na fase de pré-reunião, através do planejamento da agenda. Sistemas de suporte a facilitação que apoiem o facilitador durante a gestão da reunião ainda são poucos, não tendo sido encontrado nenhum sistema que preencha satisfatoriamente todas as necessidades do facilitador.

3 Abordagem Proposta

Neste capítulo é apresentada a proposta de uma ferramenta para apoiar a construção e modelagem de processos colaborativos, utilizando para isso o conceito de *ThinkLets* como “blocos de construção”. Inicialmente, será contextualizada a utilização dos *ThinkLets* em reuniões e processos colaborativos (seção 3.1). Nas seções 3.2 e 3.3 é apresentado um mapeamento dos *ThinkLets* em padrões colaborativos e nos tipos de resultados possíveis de serem obtidos por eles. Esse mapeamento será utilizado pela ferramenta como forma de prover ao seu usuário uma percepção facilitada ao realizar a escolha de determinado *ThinkLet* a ser aplicado em uma atividade de seu processo colaborativo, de acordo com o padrão colaborativo que ela visa reproduzir ou tipo de resultado que visa obter. Na seção 3.4 será apresentado um modelo para formalização das *entradas/saídas* de cada *ThinkLet* e, baseado nisso, será apresentada uma tabela que visa explicitar a compatibilidade existente entre *ThinkLets*. Ou seja, quais *ThinkLets* são mais apropriados a serem executados após determinado *ThinkLet*, ou seja, qual a adequabilidade de um determinado *ThinkLet* após a execução de outro. Este mapeamento de compatibilidade é um insumo para a ferramenta auxiliar o usuário na construção do passo a passo do processo colaborativo. Por fim, na seção 3.5 são discutidas regras mais avançadas, que também podem ser aplicadas pela ferramenta. Essas regras foram baseadas em análises empíricas e documentadas por Briggs *et al* (2005). Algumas dessas regras são de difícil formalização de forma a serem aplicadas em uma ferramenta computacional e, para o protótipo desenvolvido como parte deste trabalho, foi adotada apenas uma parte delas.

3.1 *ThinkLets* como componentes do processo colaborativo

Dados os aspectos mencionados anteriormente no Capítulo 2, pode se considerar um *ThinkLet* como a menor unidade de capital intelectual requerido para criar um

padrão repetível e previsível de pensamento entre pessoas que buscam um objetivo comum. Os *ThinkLets* são peça chave na Engenharia de Colaboração, cujo objetivo é criar e implantar processos colaborativos de alto valor e recorrentes, que possam ser aplicados pelos próprios praticantes, independentemente da presença de um facilitador experiente. De acordo com Kolfshoten (2007), com a adoção dos *ThinkLets* como uma biblioteca, a tarefa de construção do processo colaborativo pode ser simplificada significativamente.

Ao adotar uma abordagem com os *ThinkLets*, o processo de *design*, que de outra forma envolveria buscar possibilidades, avaliá-las e definir todos os passos do processo, passa a ser apenas uma atividade de escolha. O processo colaborativo pode ser definido através da decomposição em uma sequência de passos ou atividades, onde cada uma será realizada através da execução de um *ThinkLet*. Diz-se que os *Thinklets* podem ser vistos como “blocos de construção”, ou seja, componentes do processo colaborativo.

O primeiro passo nessa modelagem é juntar cada um desses blocos de uma forma coerente, visando atingir os objetivos desejados. O processo pode ser construído utilizando *ThinkLets*, sendo que a saída de cada um poderá servir como entrada para o seguinte, ou seja, o fluxo de controle passará de um *ThinkLet* a outro durante a reunião. Reuniões podem ser representadas como fluxo de dados ou diagramas de atividades. Uma seleção inicial de *ThinkLets* é baseada não somente em compatibilidade de entradas e saídas, mas também em objetivos, recursos e resultados esperados. Adicionalmente, existem características particulares dos *ThinkLets* (por exemplo, número de participantes ou nível de consenso) que devem ser levados em consideração no momento da escolha.

3.2 Mapeamento dos ThinkLets por padrões colaborativos

Todos os *ThinkLets* podem ser categorizados de acordo com uma classificação de padrões colaborativos. Esses padrões representam os tipos de interações e atividades desempenhadas por um grupo durante uma atividade colaborativa. Por exemplo, uma

atividade que visa basicamente gerar idéias para resolver determinado problema está associada ao padrão colaborativo *Generate*; uma atividade que visa realizar avaliação da qualidade de um conjunto de soluções possíveis para determinado problema está associada ao padrão colaborativo *Evaluate*.

Os padrões colaborativos são (KOLFSCHOTEN, G., VREEDE, G.J., 2007):

- ***Generate***: ir de um contexto de poucos para muitos conceitos compartilhados pelo grupo. Ou seja, esse padrão visa gerar idéias, comentários ou qualquer tipo de conteúdo que o grupo necessitar.
- ***Reduce***: ir de um contexto de muitos conceitos para um com foco em poucos conceitos que o grupo considere mais relevantes. Aqui o foco é reduzir um conjunto de conteúdo eliminando redundâncias, itens sem relevância, etc.
- ***Clarify***: ir de uma situação com pouco ou nenhum entendimento comum de conceitos para uma situação com maior entendimento comum dos conceitos e dos termos e utilizados para expressá-los.
- ***Evaluate***: ir de uma situação com pouco para uma situação com maior entendimento do valor relativo dos conceitos em consideração. Esse padrão visa realizar uma avaliação de qualidade no conteúdo compartilhado pelo grupo, seguindo certos critérios.
- ***Build consensus***: ir de uma situação com menos para uma situação com mais membros do grupo dispostos a se comprometer com um propósito.

3.3 Mapeamentos dos *ThinkLets* por tipo de resultado

Os *ThinkLets* podem também ser categorizados de acordo com o tipo de resultado que obtemos a partir da sua aplicação em um processo colaborativo. De acordo com (KOLFSCHOTEN, G., VREEDE, G.J., 2007), os tipos de resultados são:

- **Input:** Podemos citar quatro tipos de *input*: *criativo*, que consistem de idéias e soluções; *informativos*, que consistem em fatos e experiências; *visionários*, que consistem em requisitos futuros, visões, cenários e tendências; e *reflexivos*, que consistem em comentários, preferências e opiniões.
- **Structure:** Os seguintes tipos de estrutura são encontrados: *cluster*, que consiste de um conjunto de conceitos relacionados; um *ranking*, que consiste em um conjunto de conceitos ordenados por algum critério; *sequência*, em que as relações temporais entre conceitos são indicadas.
- **Focus:** Incluem os seguintes tipos: *seleção*, em que apenas alguns poucos conceitos são escolhidos pelo grupo; *sumário*, em que conceitos com significado similar são integrados; *escopo*, em que as fronteiras para uma coleção de conceitos é formulada; *direção*, em que conceitos que correspondam a uma ação específica são levados em consideração.
- **Shared understanding:** Temos os seguintes tipos: *conhecimento compartilhado*, onde é alinhado o conhecimento do grupo e seus significados; *aprendizado mútuo*, onde as pessoas aprendem umas com as outras, aumentando o conhecimento individual e do grupo; *diferenças mútuas*, onde são reveladas diferenças de opiniões de forma a aumentar a compreensão sobre os conflitos do grupo.

- **Commitment:** Temos os seguintes tipos de comprometimento: *decisão*, que pode ser alcançada pela escolha da maioria ou através de meios mais sofisticados de tomada de decisão; *suporte*, onde as pessoas se comprometem com um plano ou proposta; *acordo*, onde as pessoas fecham uma determinada resolução, como por exemplo, gastar determinado recurso; *consenso*, onde todos os principais *stakeholders* se comprometem com uma proposta.
- **Empathy:** Temos os seguintes tipos: *respeito* pelos outros *stakeholders*; *shared stakes*, quando pessoas acolhem interesses de outros entre os seus próprios; *consideração* pelos interesses alheios; *team bond*, em que objetivos mútuos são perseguidos.

O mapeamento completo dos *ThinkLets* em padrões colaborativos e tipos de resultados pode ser visto na Figura 3-1.

3.4 Mapeamento dos ThinkLets através de entradas/saídas

Na tarefa de planejamento e construção do processo colaborativo, o facilitador irá definir uma sequência de atividades que serão executadas pelo grupo. Para cada atividade de interação do grupo poderá ser associado um *ThinkLet*, que guia a forma como o grupo deve interagir através do script definido por ele. Assim sendo, o processo colaborativo consistirá de uma sequência de passos, onde cada passo deverá reproduzir um *ThinkLet*, conforme especificado na definição dele.

Uma preocupação a mais a ser levada em consideração pelo facilitador na construção do processo diz respeito então ao encadeamento entre os *ThinkLets* de forma que haja coerência e compatibilidade entre suas respectivas entradas/saídas. Por exemplo, um *ThinkLet* que necessita como entrada uma lista priorizada de idéias jamais poderá ser aplicado após uma atividade de *brainstorm* livre, pois esta fornece uma lista desordenada e possivelmente redundante de ideias geradas pelo grupo.

	RESULT					PATTERN						
	Input	Structure	Focus	Shared Understanding	Commitment	Empathy	Generate	Reduce	Clarify	Organize	Evaluate	Consensus Building
FreeBrainstorm	X						X					
OnePage	X						X					
Comparative	X						X					
LeafHopper	X						X					
DealersChoice	X						X					
PlusMinus	X						X					
TopFive	X						X					
BranchBuilder	X						X					
TheLobbyist	X						X					
FastFocus			X	X				X	X			
OneUp			X	X				X	X			
BucketBriefing			X	X				X	X			
DimSum			X	X				X	X			
Pin the Tail			X					X				
BroomWagon			X					X				
GoldMiner			X					X				
ExpertChoice			X					X				
GarlicSqueezer			X					X				
ReviewReflect				X					X			
Concentration	X	X	X							X		
ThemeSeeker	X									X		
RichRelations	X									X		
Popcorn Sort	X									X		
ChauffeurSort	X									X		
Evolution	X									X		
StrawPoll	X		X								X	
BucketWalk			X								X	
BucketShuffle			X								X	
MultiCriteria			X								X	
CheckMark		X	X								X	
StakeholderPoll	X		X								X	
BucketVote	X										X	
Crowbar			X									X
MoodRing			X	X								X
PointCounterPoint			X	X								X
Red-Light-GreenLight			X	X								X
Could-Be-Should-Be	X	X	X				X	X	X	X		
SevenUp			X					X			X	X

Figura 3-1 Mapeamento de *ThinkLets*

Nesta seção apresentamos uma abordagem para a formalização e categorização dos diferentes tipos de entradas/saídas utilizadas pelos *ThinkLets*. A princípio, categorizaremos em termos de estruturas de dados. Em seguida, incluiremos alguns qualificadores que são necessários para diferenciar características quanto a organização semântica nessas estruturas (por exemplo, uma lista de ideias pode estar priorizada ou não, ou seja, a estrutura de dados é a mesma, mas sua interpretação em termos do processo é completamente diferente). A lista de estruturas que iremos utilizar consiste em:

- **Questão:** trata-se de uma simples questão ou breve descrição do tópico a ser abordado pelo grupo na discussão.
- **Lista de questões (ou tópicos):** simples extensão da estrutura anterior, para utilização em *ThinkLets* que dão suporte ao tratamento de mais de uma questão em sua execução.
- **Lista de ideias:** qualquer lista de idéias ou comentários gerados pelo grupo em uma atividade anterior.
- **Lista Categorizada de ideias:** lista de ideias separadas em categorias ou tópicos.
- **Categorias:** lista de categorias ou tópicos a serem utilizados para organizar ideias geradas pelo grupo em determinada atividade.
- **Hierarquia:** estrutura que consiste em uma árvore de tópicos ou categorias, a ser utilizada para a organização posterior das ideias.
- **Lista Estruturada de Ideias:** lista de ideias devidamente categorizada em um Outline de categorias ou tópicos.

- **Lista de Critérios:** consiste na lista de aspectos a serem considerados na avaliação da qualidade de uma ideia/solução que pode, opcionalmente, incluir uma lista de pesos que cada um representaria na avaliação final.

- **Lista de Grupos:** listagem de participantes separados por grupos. É utilizada quando os participantes da reunião são divididos em grupos menores para determinada atividade.

Para cada *ThinkLet*, poderemos então formalizar todas as suas entradas/saídas, conforme as estruturas definidas acima, da seguinte forma:

1. FreeBrainstorm:

- a. Entrada: uma questão a ser abordada pelo grupo
- b. Saída: uma lista de idéias

2. OnePage:

- a. Entrada: uma questão a ser abordada pelo grupo
- b. Saída: uma lista de idéias

3. ComparativeBrainstorm:

- a. Entrada: uma lista de questões a serem abordada pelo grupo. Aqui vale deixar claro, no entanto, que a lista de questões de entrada para este *ThinkLet* não deve ser elaborada livremente, mas baseada diretamente em um conjunto de critérios de qualidade da solução previamente definidos. Por exemplo, imagine uma equipe que deva realizar um *brainstorming* para buscar formas de melhorar seus procedimentos de produção. Os critérios de qualidade poderiam ser: reduzir tempo de produção; maior qualidade de produto; reduzir custo de produção. As questões de entrada para um *ComparativeBrainstorm* poderiam então ser formuladas da seguinte forma:

- Sugira uma solução mais provável de reduzir o tempo de produção em relação às idéias visualizadas na tela.

- Sugira uma solução mais provável de gerar produtos com maior qualidade em relação às idéias visualizadas na tela.
 - Sugira uma solução mais provável de diminuir os custos de produção em relação às idéias visualizadas na tela.
- b. Saída: uma lista de idéias/soluções em potencial. Possivelmente os resultados do *ComparativeBrainstorm* apresentarão maior qualidade em relação ao *FreeBrainstorm* ou *OnePage* pois foram baseados em critérios bem definidos, fazendo o grupo dar contribuições mais focadas em qualidade.
- 4. LeafHopper:**
- a. Entrada: uma lista de tópicos/questões
 - b. Saída: uma lista de idéias/comentários categorizados por tópico
- 5. DealersChoice:**
- a. Entrada: uma lista de tópicos/questões
 - b. Saída: uma lista de idéias/comentários categorizados por tópico
- 6. Plus-Minus-Interesting:**
- a. Entrada: uma lista de idéias.
 - b. Saída: uma lista de idéias com maior aprofundamento e considerações de prós e contras.
- 7. TopFive:**
- a. Entrada: uma lista de idéias. O objetivo desse *ThinkLet* é elaborar e aprofundar as idéias mais importantes de um passo anterior antes de seguir adiante. Para tal, é necessário que a lista de idéias tenha sido priorizada previamente.
 - b. Saída: uma lista de cinco idéias com maior aprofundamento e detalhamento.
- 8. BranchBuilder:**
- a. Entrada: uma lista de tópicos que podem ser considerados como os principais nós de uma hierarquia para organizar o conhecimento do grupo.
 - b. Saída: uma hierarquia para estruturar o conhecimento do grupo.

9. TheLobbyist:

- a. Entrada: uma lista de idéias a serem avaliadas. Pode ser o resultado de um *FreeBrainstorm* ou *OnePage*, por exemplo.
- b. Saída: não há uma saída concreta para este *ThinkLet*.

10. FastFocus:

- a. Entrada: lista de idéias geradas em uma atividade *brainstorming*.
- b. Saída: lista de idéias reduzida, sem redundância, com foco nas questões mais relevantes.

11. OneUp:

- a. Entrada: lista de idéias geradas em uma atividade *brainstorming*.
- b. Saída: lista de idéias reduzida, sem redundância, com foco nas questões mais relevantes e uma lista de critérios para avaliação posterior.

12. BucketBriefing:

- a. Entrada: lista categorizada de idéias/comentários de uma atividade de *brainstorming*.
- b. Saída: lista categorizada de idéias/comentários reduzida, sem redundância, com foco nas questões mais relevantes.

13. DimSum:

- a. Entrada: nenhuma entrada
- b. Saída: uma única afirmação ou definição aceita por todos os participantes.

14. Pin the Tail on the Donkey:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.
- b. Saída: lista de idéias com os itens mais importantes destacados.

15. BroomWagon:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.
- b. Saída: lista de idéias reduzida, sem redundância, com foco nas questões mais relevantes.

16. GoldMiner:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.

- b. Saída: lista de idéias reduzida, sem redundância, com foco nas questões mais relevantes.

17. ExpertChoice:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming* e possivelmente uma estrutura para organizar as idéias.
- b. Saída: lista estruturada de idéias em uma lista de categorias ou hierarquia.

18. GarlicSqueezer:

- a. Entrada: lista categorizada de idéias/comentários de um atividade de *brainstorming*.
- b. Saída: lista categorizada de idéias/comentários reduzida, sem redundância, com foco nas questões mais relevantes.

19. ReviewReflect:

- a. Entrada: qualquer conteúdo existente, podendo ser uma lista de idéias, lista categorizada de idéias ou outro documento qualquer, possivelmente não estruturado.
- b. Saída: conteúdo revisado, validado e com foco nas questões mais importantes.

20. Concentration:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.
- b. Saída: lista de idéias reduzida, sem redundância, com foco nas questões mais relevantes.

21. ThemeSeeker:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.
- b. Saída: categorias para organizar o conteúdo.

22. RichRelations:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.
- b. Saída: categorias para organizar o conteúdo.

23. PopcornSort:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming* e lista de categorias para organizar as idéias.

- b. Saída: lista categorizada de idéias.

24. ChauffeurSort:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming* e lista de categorias para organizar as idéias.
- b. Saída: lista categorizada de idéias.

25. Evolution:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.
- b. Saída: lista categorizada de idéias.

26. StrawPoll:

- a. Entrada: lista de idéias a serem avaliadas.
- b. Saída: lista de idéias priorizada e tabela de avaliação exibindo resultado da votação.

27. BucketWalk:

- a. Entrada: lista categorizada de idéias.
- b. Saída: lista categorizada de idéias validada. Cada item é validado como pertencente à categoria.

28. BucketShuffle:

- a. Entrada: lista categorizada de idéias.
- b. Saída: lista categorizada de idéias, com priorização dentro de cada categoria.

29. MultiCriteria:

- a. Entrada: lista de idéias a serem avaliadas e lista de critérios para avaliação.
- b. Saída: lista de idéias priorizada e tabela de avaliação exibindo resultado da votação.

30. CheckMark:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.
- b. Saída: lista de idéias priorizada de acordo com as preferências dos participantes.

31. StakeholderPoll:

- a. Entrada: lista de idéias a serem avaliadas.

- b. Saída: lista de idéias priorizada, separada por grupos.

32. BucketVote:

- a. Entrada: lista de idéias a serem avaliadas.
- b. Saída: lista de idéias categorizadas em clusters de prioridades.

33. Crowbar:

- a. Entrada: lista de idéias e resultado de um *StrawPoll*.
- b. Saída: lista de idéias priorizadas.

34. MoodRing:

- a. Entrada: uma sentença sobre um tópico a ser debatida.
- b. Saída: argumentos, opiniões sobre a questão.

35. PointCounterPoint:

- a. Entrada: uma sentença sobre um tópico a ser debatida.
- b. Saída: argumentos, opiniões sobre a questão.

36. Red-Light-Green-Light:

- a. Entrada: lista de idéias e resultado de um *MultiCriteria*.
- b. Saída: uma lista priorizada de idéias.

37. Could-Be-Should-Be:

- a. Entrada: tópico ou questão complexa a ser abordada pelo grupo.
- b. Saída: lista de idéias estruturada.

38. SevenUp:

- a. Entrada: lista de idéias geradas em uma atividade de *brainstorming*.
- b. Saída: sete idéias/conceitos a serem analisados posteriormente.

Apesar de essa categorização fornecer uma boa percepção sobre como podemos encadear os *ThinkLets*, ela não captura características diferenciadas em uma mesma estrutura produzida de formas distintas. Por exemplo, uma lista de idéias obtidas a partir de um *FreeBrainstorm* é diferente da lista obtida a partir de um *FastFocus* apesar de serem estruturas idênticas. No caso do *FastFocus*, essa lista é reduzida, não contém redundâncias e itens irrelevantes foram removidos. A seguir definimos alguns

qualificadores para as estruturas de dados, de forma a identificar essas características específicas.

Detalhamento (D): uma lista de idéias irá apresentar menor detalhamento e será mais superficial em etapas iniciais de *brainstorm*, quando o foco for de alcançar maior abrangência de idéias ao invés de maior aprofundamento. Em etapas seguintes de elaboração, as idéias irão ser desenvolvidas e aprofundadas, gerando uma lista diferenciada. Para isso pode se utilizar o qualificador **(D)**.

Foco (F): uma lista de ideias pode ser gerada de forma bastante livre ou guiada e com foco em critérios bem definidos, como no caso do *ComparativeBrainstorm*.

Priorização (P): uma lista de idéias pode passar uma etapa onde serão priorizadas de acordo com uma avaliação feita pelo grupo, de forma a colocar no topo as mais relevantes e de maior qualidade.

Redução (R): uma lista de idéias que passou por uma etapa de redução, onde foram eliminadas ideias irrelevantes, redundantes e ambíguas utiliza esse qualificador.

A partir da consideração desses fatores, é possível elaborar um mapeamento de compatibilidade entre *ThinkLets*, onde podemos verificar a viabilidade de combiná-los, fazendo com que a saída de um seja utilizada como entrada do seguinte. Podemos ver o mapeamento de compatibilidade na figura abaixo (Figura 3-2). As células vermelhas indicam combinações impossíveis; as laranjas, combinações possíveis e as verdes, combinações boas.

3.5 Outras regras para escolha de ThinkLets

Até este momento foram abordadas formas de guiar a escolha do *ThinkLet* mais apropriado através da associação da atividade com um padrão colaborativo ou tipo de resultado desejado para ela. Vimos também que podemos orientar a construção iterativa

do processo, encadeando *ThinkLets* em sequência, onde a saída de um servirá como entrada para o seguinte, através de um mapeamento de compatibilidade.

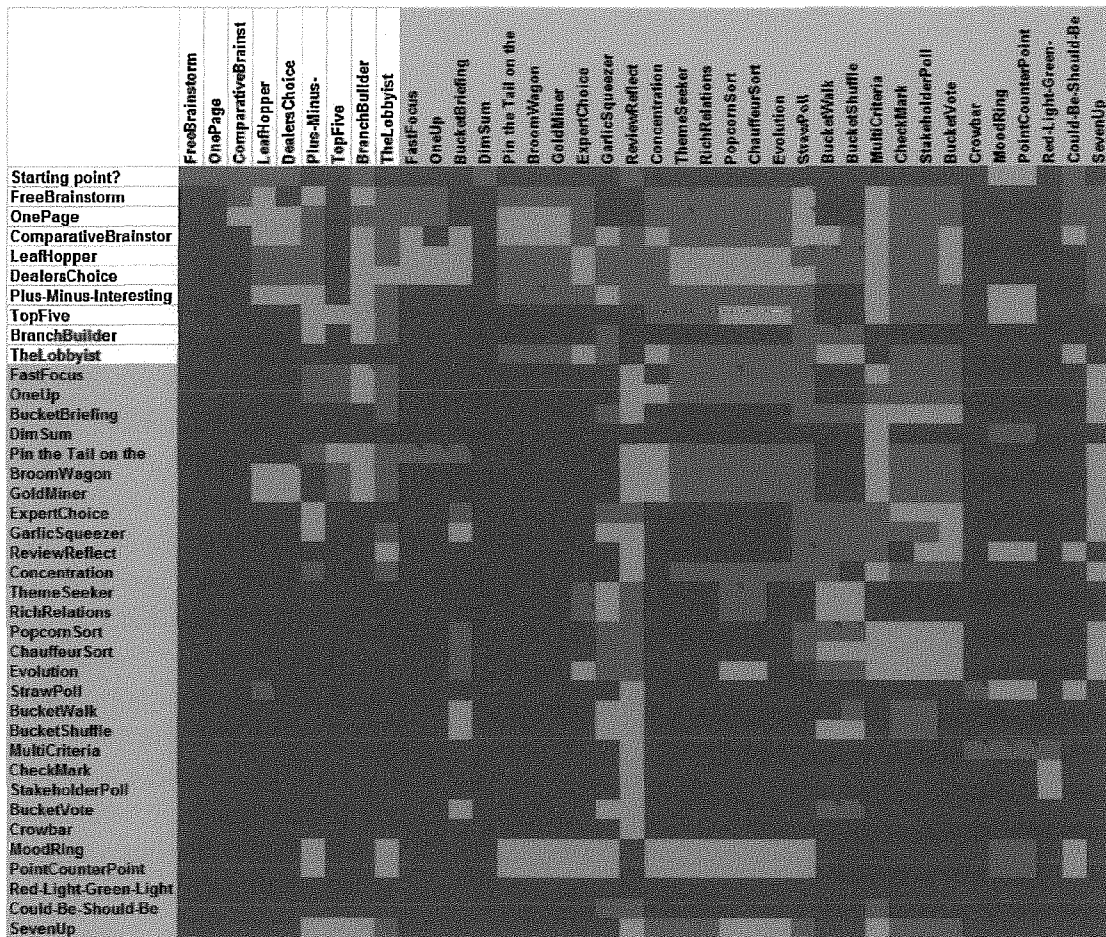


Figura 3-2 Mapeamento de compatibilidade baseado em entrada/saída

Além dos critérios já citados, existem diversos outros fatores contextuais que podem e devem ser levados em consideração no momento de escolha de um *ThinkLet*. A maioria desses fatores está relacionada a características do grupo, restrições para a reunião (como o tempo), quais insumos estão disponíveis, entre outros.

Nesse sentido, podemos ver em (BRIGGS, R.O. *et al*, 2001) um extenso catálogo de *ThinkLets* com diversas orientações de utilização baseadas nesses fatores. Essa base de conhecimento, registrada ao longo de diversas experiências em aplicações

práticas, constituem um valioso recurso para guiar decisões na construção dos processos colaborativos. Seu formato não estruturado, no entanto, dificulta essa utilização. Através da estruturação e codificação apropriada, podemos formar uma base de regras que pode ser utilizada por uma ferramenta de auxílio na construção dos processos colaborativos.

Assim, dentro da proposta deste trabalho, foram selecionadas algumas dessas regras e implementadas dentro da ferramenta proposta. Todas as regras catalogadas, implementadas na base de regras ou não, serão ainda utilizadas para orientar o facilitador no seu processo de decisão. Podemos ver em (Figura 3-3) uma pequena amostra das regras utilizadas, separadas por *ThinkLet*. A tabela completa pode ser encontrada no Apêndice A.

FreeBrainstorm	SIM	Para fazer o grupo divergir bastante em busca de novas ideias. Para grupos de 5 ou mais participantes Para grupos heterogêneos com visões bastante distintas	
	NÃO	Para grupos de menos de 5 participantes Para aprofundar em ideias	OnePage ComparativeBrainstorm
OnePage	SIM	Para gerar alguns comentários (em torno de 10 ou menos) sobre um tópico Para grupos de 5 ou menos pessoas Para grupos de 5 pessoas durante uma sessão de até 10 minutos Para reuniões onde provavelmente não haverá muitos comentários gerados Para dar melhor feedback entre os participantes em grupos distribuídos	
	NÃO	Quando for esperado mais de 10 comentários Para reuniões onde 5 ou mais pessoas não divergiem ao exporem as ideias Para reuniões onde serão abordados mais de um tópico ao mesmo tempo	FreeBrainstorm ou Comparative Brainstorm FreeBrainstorm ou Comparative Brainstorm LeafHopper ou DealersChoice
ComparativeBrainstorm	SIM	Para reuniões onde critérios de qualidade para avaliar ideias e soluções já forem conhecidos	
	NÃO	Para reuniões onde se deseja que as pessoas busquem soluções novas ou inovadoras	
LeafHopper	SIM	Para reuniões onde serão abordados mais de um tópico ao mesmo tempo Para reuniões onde os participantes tenham diferentes níveis de interesse (grupo heterogêneo)	FreeBrainstorm
	NÃO	Para reuniões onde não seja necessário que cada participante participe em todos os tópicos Se for necessário que os tópicos sejam abordados em uma ordem determinada Se for necessário que todos os tópicos sejam abordados por todos os participantes	DealersChoice DealersChoice
DealersChoice	SIM	Para reuniões onde serão abordados mais de um tópico ao mesmo tempo Se for necessário que certos participantes abordem determinados tópicos	
	NÃO	Se a ordem em que os tópicos são abordados não for relevante Se o background, posição ou expertise dos participantes não for relevante	LeafHopper LeafHopper
Plus-Minus-Interesting	SIM	Para elaborar melhor algumas ideias antes de uma avaliação formal	
	NÃO	Se a elaboração não for seguida de uma avaliação formal	
TopFive	SIM	Para uma transição entre dois outros Thinklets Para lidar com sobrecarga de informação gerada do passo anterior	
	NÃO	Se for necessário que cada conceito identificado seja exaustivamente discutido	

Figura 3-3 Regras para escolhas de ThinkLets

3.6 O Sistema *DynaMeetings*

Nesta seção, apresentaremos a proposta de implementação de um sistema que visa ser uma ferramenta de trabalho colaborativo com suporte a reuniões eletrônicas dinâmicas com ênfase no suporte ao facilitador. Esse sistema, denominado

DynaMeetings, está em fase inicial de definição e implementação. Como parte deste trabalho, foi desenvolvido um dos módulos desse sistema. Esse módulo será responsável pela construção de uma reunião eletrônica utilizando os *ThinkLets* como blocos de construção. Sua principal proposta é servir de apoio a um facilitador inexperiente de modo a produzir resultados com a qualidade que um facilitador mais experiente conseguiria.

O módulo foi desenvolvido com base nos conceitos e propostas expostas na seção anterior. Inicialmente, serão abordados os princípios de funcionamento desse módulo, seguindo de uma análise rápida da interface. No capítulo seguinte, será apresentada a arquitetura macro para, em seguida, aprofundar em detalhes das decisões tomadas para a implementação dos principais aspectos da ferramenta, como o processamento de regras. Nesse ponto, é feita também uma breve análise das tecnologias e componentes utilizados no desenvolvimento.

Vale ressaltar que aqui iremos abordar em maiores detalhes apenas o módulo proposto para a construção do processo colaborativo baseado em *ThinkLets*, pelo facilitador. O sistema *DynaMeetings* possui um escopo bem mais abrangente do que o proposto neste trabalho, incluindo os seguintes módulos:

1. *DynaMeetings AgendaBuilder*: Módulo abordado neste trabalho. Seus requisitos, funcionamento e arquitetura serão descritos adiante.
2. *DynaMeetings FacilitAid*: Esse módulo visa auxiliar o facilitador na condução da execução das fases de uma reunião, fornecendo indicadores estatísticos sobre o andamento. Através desse módulo, o facilitador pode obter uma maior percepção quanto à necessidade de intervir em determinados momentos, de acordo com a situação observada através desses indicadores.

3. Módulo (ainda sem nome) que visa auxiliar o facilitador na análise dos resultados obtidos após o término de cada fase da reunião. O objetivo dessa análise entre etapas é a verificação de necessidade de ajustes na agenda definida inicialmente. Por exemplo, se uma primeira etapa da agenda previa obter como resultado uma quantidade x de idéias, mas sua execução acabou gerando $3x$; talvez seja necessário adaptar a agenda inicial para incluir uma etapa de redução antes de seguir para a categorização. Esse módulo tem como requisitos recomendar esses ajustes, baseado em alguns indicadores, e sugerir também qual *ThinkLet* mais adequado para esse ajuste na agenda.
4. Módulo (ainda sem nome) que visa fornecer recomendações sobre participantes para uma reunião. Através de uma análise das redes sociais existentes em um contexto, este módulo se propõe a inferir os melhores participantes para uma reunião sobre determinado assunto.

Atualmente, está em desenvolvimento o módulo 4. O módulo *FacilitAid* encontra-se com um protótipo finalizado e seu experimento de validação apresentou bons resultados (MARQUES, L.C., 2009).

3.6.1 O módulo *AgendaBuilder*

A ferramenta *AgendaBuilder* foi desenvolvida e analisada como parte deste trabalho. Seu objetivo é dar suporte a um facilitador no planejamento de um processo colaborativo, especificando como este será executado através da criação de um fluxo de atividades. A sua concepção foi baseada na abordagem para a Engenharia de Colaboração apresentada no capítulo anterior (KOLFSCHOTEN, G., VREEDE, G.J., 2007).

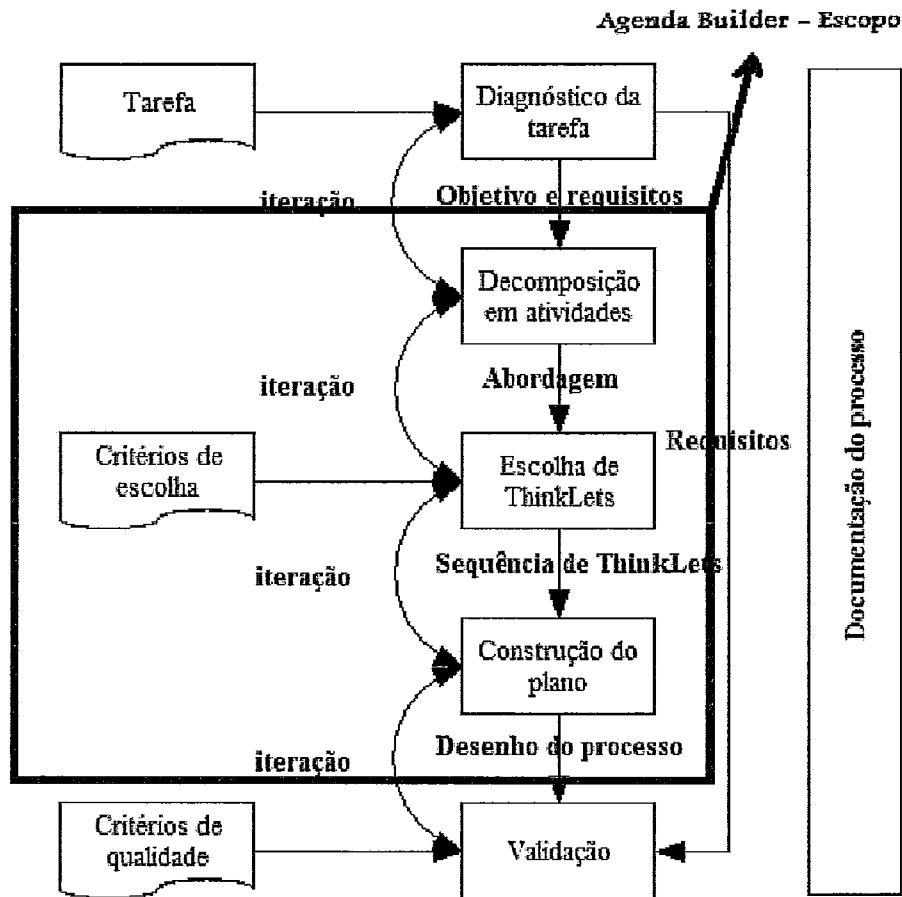


Figura 3-4 Visão geral do escopo suportado pela ferramenta *AgendaBuilder*

Dentro dessa concepção para as atividades necessárias na Engenharia de Colaboração, o escopo suportado pela ferramenta pode ser destacado da seguinte forma (Figura 3-4):

1. **Diagnóstico da tarefa:** A ferramenta **não** oferece suporte direto ao facilitador nessa atividade, mas as informações levantadas nessa etapa serão registradas na ferramenta, como o objetivo do processo e entregas. As informações registradas aqui serão utilizadas pela ferramenta na etapa de escolha de *ThinkLets*.

2. **Decomposição da tarefa em atividades:** Essa atividade é totalmente suportada pela ferramenta, que permite que o facilitador realize a decomposição criando as atividades através de uma interface gráfica. Também permite que o facilitador associe as atividades a padrões colaborativos e resultados e registre outras informações relevantes para cada atividade do processo. Essas informações serão parte do insumo utilizado pela ferramenta para auxiliar a tomada de decisão na etapa seguinte.
3. **Escolha de ThinkLet:** O suporte a essa atividade é o principal foco da ferramenta e onde reside a maior parte da inteligência implementada. Nesta atividade, a ferramenta oferece uma lista de sugestões dos *ThinkLets* mais adequados para cada atividade, com base nas informações fornecidas. Além de sugerir, a ferramenta também fornece pequenas informações sobre aplicabilidade de cada *ThinkLet*, permitindo uma maior percepção para a tomada de decisão do facilitador.
4. **Definição da agenda:** A ferramenta permite que o facilitador especifique todas as atividades do processo colaborativo, sejam elas atividades mapeadas para *ThinkLets* ou atividades de suporte, como as aberturas, aquecimentos, introdução na tecnologia etc. Para cada atividade, pode ser definida sua descrição detalhada e o mapeamento para padrões colaborativos, resultados e *ThinkLets* é totalmente opcional.

O princípio de funcionamento da ferramenta pode ser visto na Figura 3-5. O usuário, ao elaborar um processo colaborativo e definir as suas atividades com o devido registro de todas as informações relevantes, ele gera uma massa de dados que será considerada como o contexto do processo colaborativo. A partir desse contexto, a ferramenta realiza um processamento, fazendo um cruzamento com um repositório onde estão contidos todos os *ThinkLets* e seus dados de definição (ou seja, visão geral, *inputs*, *outputs*, configuração e passo a passo). Nesse repositório, a ferramenta dispõe também das seguintes informações: mapeamento de cada *ThinkLet* para os padrões colaborativos e tipos de resultados existentes; mapeamento de compatibilidade, que fornece o *insight*

sobre, para cada *ThinkLet*, quais os que podem sucedê-lo e quais não podem; um conjunto de regras que determinam quando determinado *ThinkLet* é apropriado e quando é inapropriado. Após realizar esse cruzamento e avaliação das regras, a ferramenta retorna ao usuário uma lista de *ThinkLets* que são recomendados para cada atividade definida. Para cada recomendação de *ThinkLet*, a ferramenta exhibe ainda as descrições textuais das situações onde cada um é apropriado e onde não, permitindo uma análise adicional para o facilitador realizar a sua escolha.

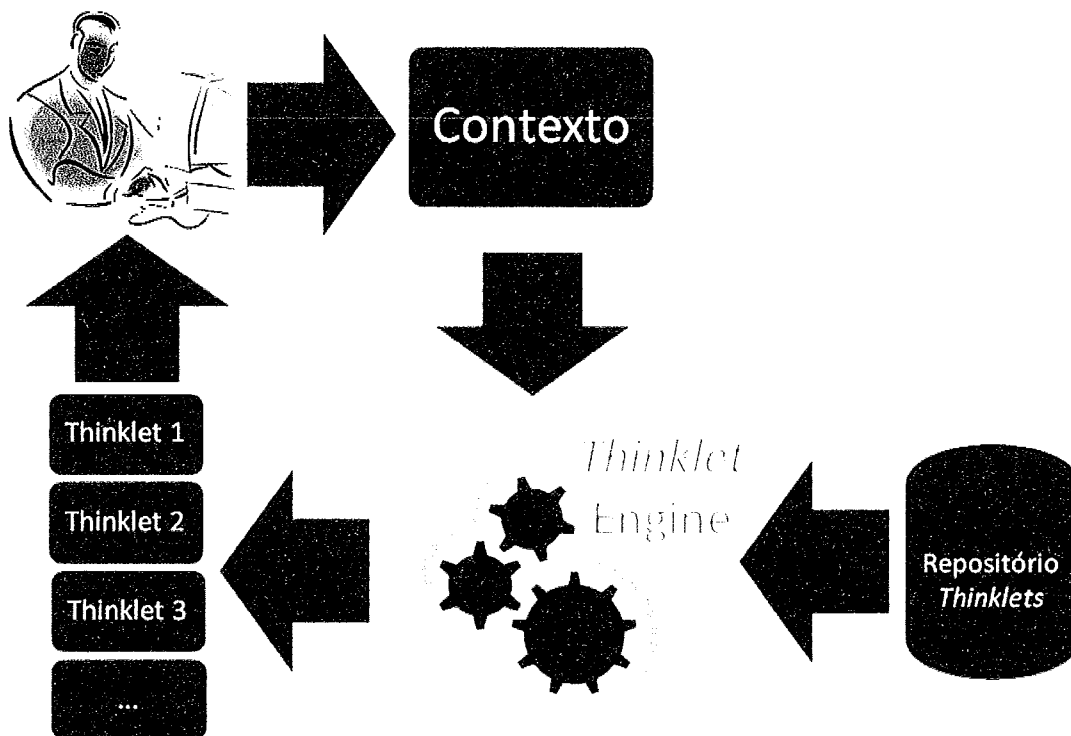


Figura 3-5 Princípio de funcionamento do *AgendaBuilder*

As informações que compõem o contexto do processo colaborativo podem ser analisadas na Figura 3-6. Por simplificação, adotamos apenas algumas informações referentes aos participantes, ao tempo, *input* e atividade. No entanto, outras informações poderiam ser consideradas também.

Para as informações relacionadas aos participantes, temos a quantidade e o perfil. Ambas as informações são registradas pelo facilitador no escopo do processo.

Para o perfil, o facilitador poderá selecionar uma de duas opções: *homogêneo*, quando os participantes possuírem *expertises*, interesses, áreas de atuação semelhantes etc; *heterogêneo*, quando o grupo for bastante diversificado. A ferramenta não disponibiliza uma metodologia específica para quantificar a heterogeneidade do grupo, de forma que ela deverá ser baseada na percepção do facilitador durante a análise dos participantes, na etapa de diagnóstico da tarefa.

O facilitador define também, no escopo do processo, o tempo máximo desejado para a realização do processo colaborativo e o tipo de *input* que será utilizado nas primeiras etapas, geralmente de *brainstorm*. Os tipos de *inputs* considerados são: um único tópico a ser desenvolvido; um conjunto de tópicos que deverão ser abordados pelo grupo; ou um conjunto de critérios de avaliação da solução, se houver.

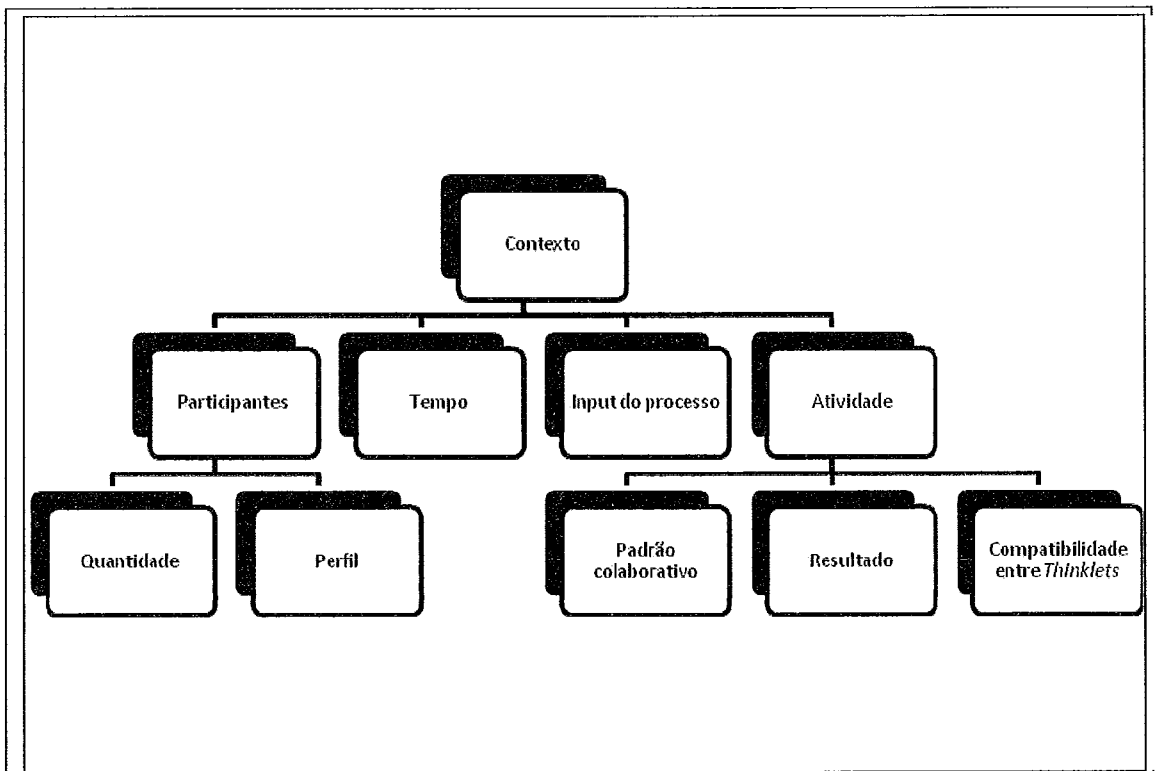


Figura 3-6 Contexto do processo colaborativo

As atividades de escopo de atividade serão definidas repetidamente para cada atividade do processo. As informações que constituem o contexto considerado pelo *ThinkLet Engine* são: o padrão colaborativo associado à atividade; o tipo de resultado gerado por ela; e a compatibilidade entre *ThinkLets*. As duas primeiras são determinadas pelo facilitador através da escolha na lista de opções disponíveis (*Generate, Reduce, Clarify, Organize, Evaluate, Build consensus*, para padrões; e *Input, Structure, Focus, Shared understanding, Commitment e Empathy*, para tipo de resultado). A informação de compatibilidade será considerada da seguinte forma: ao mapear a atividade para um *ThinkLet*, este será considerado na etapa de escolha do *ThinkLet* da atividade subsequente, através do cruzamento com o mapeamento de compatibilidade definido no repositório.

3.6.1.1 Visão geral da ferramenta e sua interface

Nesta seção será detalhado como se dá o funcionamento da ferramenta, do ponto de vista do usuário. Os detalhes de arquitetura e de implementação serão descritos no capítulo seguinte. Uma visão geral da interface da ferramenta pode ser vista na Figura 3-7.

A interface pode ser dividida nas seguintes partes:

1. Menu: Onde podem ser acessadas as funções gerais da ferramenta, como criar novo, salvar e abrir processos, impressão, funções de seleção, copiar, colar e diversas opções para modificar a visualização do desenho do processo e da interface.
2. Barra de ferramentas: As principais funções acessíveis por menu estão disponíveis nessa barra para acesso rápido. As funções disponíveis são, da esquerda para a direita (Figura 3-8): criar novo processo; abrir processo salvo previamente pelo usuário; salvar processo atual; imprimir; recortar objeto; copiar objeto; colar objeto; excluir objeto; desfazer; refazer; alterar fonte; alterar tamanho da fonte; aplicar negrito, aplicar itálico; alinhar texto à esquerda; centralizar; alinhar texto à direita; alterar cor de fonte; alterar cor de

borda; alterar cor de preenchimento; zoom. Todas as funções (excetuando-se as três primeiras) afetam a visualização dos objetos na área de construção e permitem uma maior customização gráfica do desenho do processo.

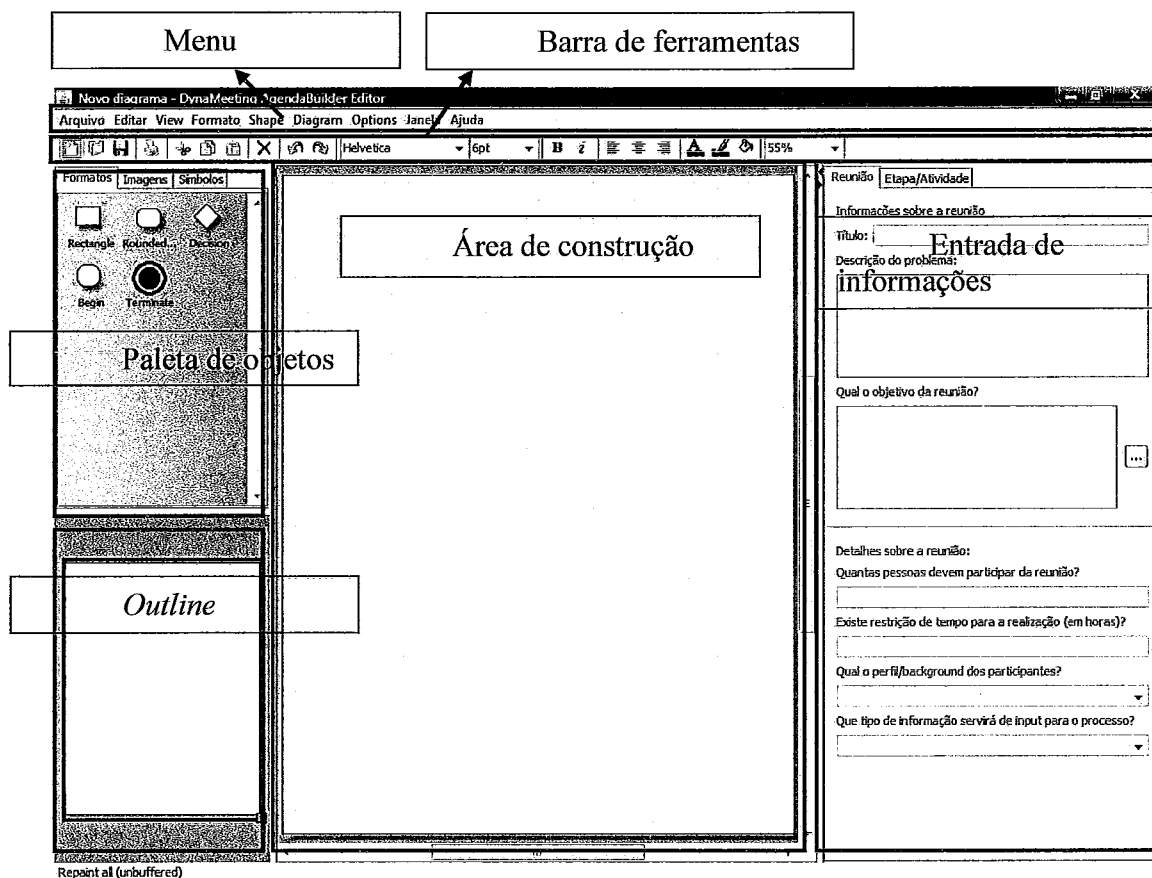


Figura 3-7 Visão geral da ferramenta

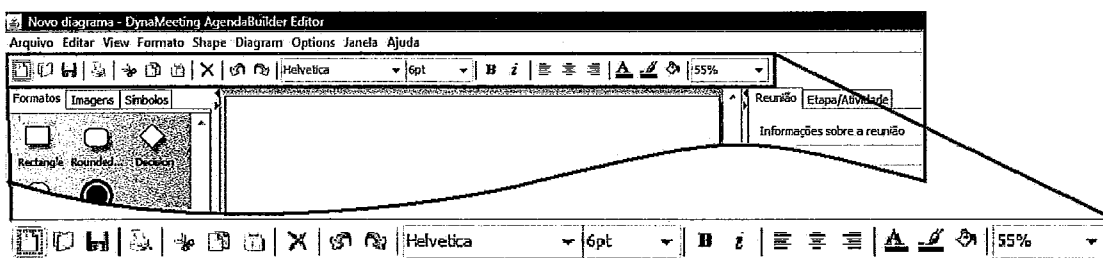


Figura 3-8 Barra de ferramentas

3. Paleta de objetos: Aqui estão disponíveis todos os objetos, símbolos e imagens que o facilitador pode utilizar para a construção gráfica do processo.

Os tipos básicos e mais importantes para estão na aba “Formatos”, onde estão os objetos que estão diretamente associados com a confecção do processo colaborativo (retângulos, que representam as atividades; losango, para representar pontos de decisão; e os marcadores de início e término do fluxo). Para adicionar esses objetos ao fluxograma, basta clicar no ícone correspondente e arrastar para a área de construção.

4. Outline: Utilizado para manipular a visualização na área de construção, permite deslocar a janela visível da página e redefinir o zoom.
5. Área de construção: Nessa área, o usuário irá construir graficamente todo o fluxo do processo colaborativo, utilizando os objetos disponibilizados na paleta. A interface e funcionamento são bastante intuitivos. Por exemplo, ao incluir uma atividade, para definir a atividade no fluxo seguinte basta posicionar o cursor no meio da atividade origem (o cursor muda para o formato mão) e arrastar para uma área vazia na página. Ao soltar, é criada uma nova atividade, com a ligação indicando o sequenciamento (Figura 3-9). Também pode ser criada a ligação entre duas atividades já existentes no desenho. A ferramenta oferece ainda diversas outras facilidades visuais para a construção do fluxo, que não abordaremos aqui para manter a objetividade.

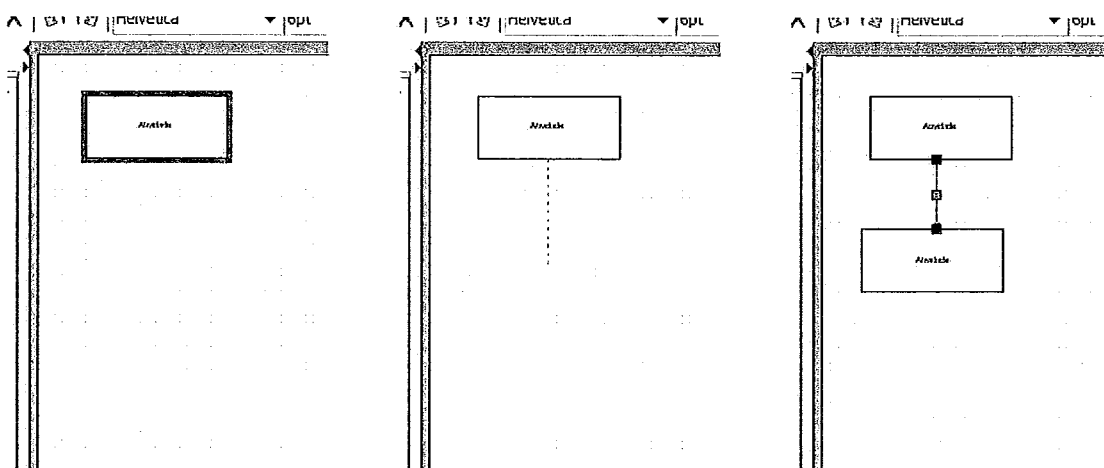


Figura 3-9 Construção fluxo – exemplo

6. Entrada de informações: Nessa área da ferramenta, dividida em duas abas, o usuário irá cadastrar e manipular todas as informações referentes ao processo

colaborativo sendo construído. Na primeira aba irão ser manipuladas todas as informações em nível do processo. Na segunda aba serão manipuladas todas as informações em nível de atividade. Para manipular as informações na segunda aba, o usuário deverá selecionar na área de construção o objeto correspondente da atividade desejada.

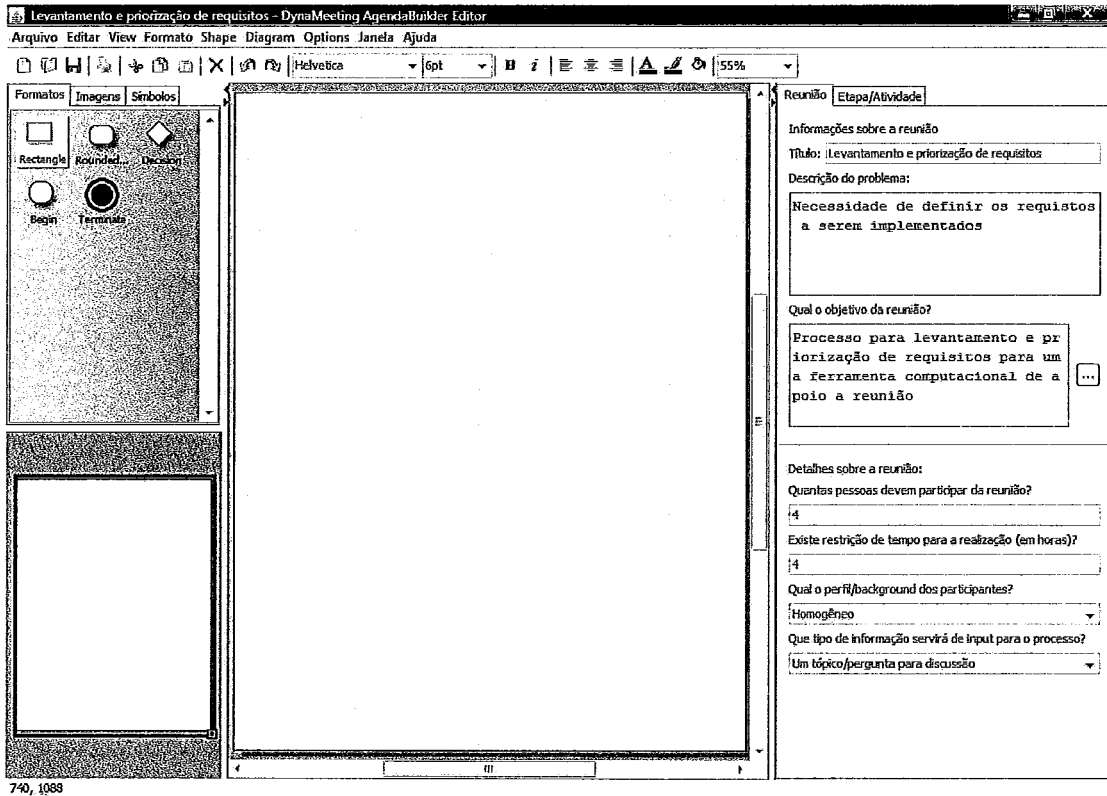


Figura 3-10 Registrando informações iniciais sobre o processo

Ao criar um novo processo colaborativo, o facilitador deverá preencher a primeira aba com as informações iniciais que descrevem o que será feito, qual o propósito da reunião e algumas informações contextuais, como quantidade de participantes e o perfil dos participantes. Essas informações são completamente opcionais. No entanto, essas informações representam um contexto no qual a ferramenta irá se basear para oferecer recomendações na etapa de escolha dos *ThinkLets*. Portanto, quanto maior a qualidade do registro realizado pelo facilitador, melhor e mais precisas serão as recomendações fornecidas pela ferramenta. Podemos ver um exemplo de

preenchimento sobre um processo colaborativo que visa levantar requisitos a serem atendidos por uma nova ferramenta de *CSCW* na Figura 3-10. Após ter registrado todas as informações de escopo da reunião, o facilitador irá proceder com a criação das atividades propriamente ditas. Ao criar uma atividade na área de construção, ele poderá registrar na segunda aba as informações do escopo da atividade, conforme Figura 3-11.

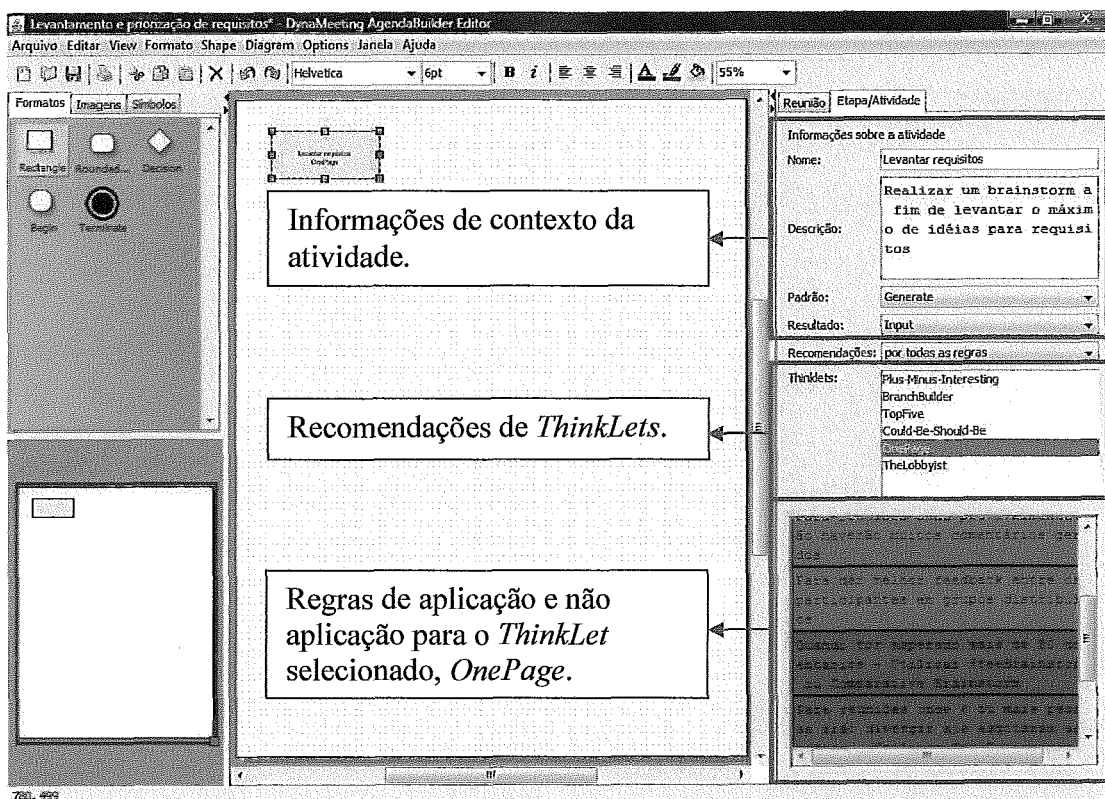


Figura 3-11 Registrando informações sobre atividades

As seguintes informações serão registradas pelo facilitador para cada atividade:

- a. Nome: Título para identificar a atividade (o nome é exibido na área de construção, dentro do objeto que representa a atividade, visando facilitar a identificação delas na visão geral do fluxo).

- b. Descrição: Texto explicativo para identificar os objetivos da atividade, o que será feito, seus pré-requisitos, resultados ou qualquer outra informação relevante.
- c. Padrão colaborativo: O facilitador irá selecionar em uma lista o padrão colaborativo que corresponde ao objetivo da atividade. Essa informação é cruzada com o repositório para guiar a escolha do *ThinkLet* apropriado.
- d. Resultado: O facilitador irá selecionar em uma lista o tipo de resultado correspondente da atividade. Essa informação é cruzada com o repositório para guiar a escolha do *ThinkLet* apropriado.
- e. *ThinkLet*: Aqui finalmente o usuário poderá determinar o *ThinkLet* a ser utilizado na atividade. A escolha será feita em uma lista de opções disponíveis. Essa lista é gerada pela ferramenta após o processamento de todas as informações de contexto fornecidas pelo usuário e irá disponibilizar apenas os *ThinkLets* considerados apropriados para a atividade. A forma como a ferramenta irá processar o contexto e gerar a lista de recomendações pode ser modificada pelo facilitador, conforme ele julgar necessário. Para isso, a ferramenta disponibiliza a lista de opções do comportamento a ser adotado (na Figura 3-11, logo acima da lista de recomendações). Temos as seguintes opções:
 - i. por Padrão: considera-se apenas o padrão colaborativo da atividade. A ferramenta simplesmente irá recomendar todos os *ThinkLets* associados ao padrão selecionado.
 - ii. por Resultado: considera-se apenas o tipo de resultado da atividade. A ferramenta simplesmente irá recomendar todos os *ThinkLets* associados ao tipo de resultado selecionado.
 - iii. por Compatibilidade: considera-se apenas a compatibilidade entre os *ThinkLets* ao longo do sequenciamento do fluxo. Ou seja, a ferramenta irá recomendar apenas os *ThinkLets* de aplicação recomendada após a execução do *ThinkLet*

mapeado na atividade anterior, conforme apresentado na Figura 3-2.

- iv. por Contexto: consideram-se apenas as informações contextuais no escopo do processo. Essas são as informações registradas na aba “Reunião”.
- v. por todas as regras: considera-se todo o contexto colaborativo, conforme Figura 3-6. Ou seja, a ferramenta irá recomendar apenas os *Thinklets* que atenderem a todos os requisitos e sejam inteiramente aderentes ao contexto. Essa é opção padrão da ferramenta.
- vi. Nenhuma: nesta opção, o facilitador poderá desabilitar completamente o mecanismo de recomendação e visualizar todos os *ThinkLets* existentes no repositório, podendo designar qualquer um deles para a atividade, ignorando a aderência ao contexto. A sua utilização é desaconselhada, pois pode gerar processos inconsistentes. Sua utilização pode ser necessária em situações excepcionais, onde o facilitador identificar que nenhuma das recomendações seja a mais apropriada. Assim, apenas facilitadores mais experientes devem utilizar esta opção.

Ao selecionar um *ThinkLet* na lista de recomendações, a ferramenta exibe um painel logo abaixo, contendo uma lista de algumas situações onde ele é aplicável e onde não é. Assim, o facilitador pode realizar uma análise detalhada com maiores informações para tomar a decisão definitiva.

3.6.1.2 Interface de Administração

A ferramenta disponibiliza ainda uma interface de administração onde podem ser realizados configurações e ajustes no comportamento da ferramenta (Figura 3-12).

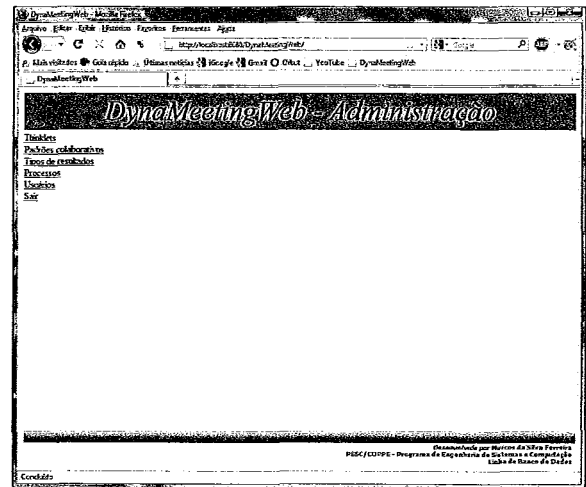
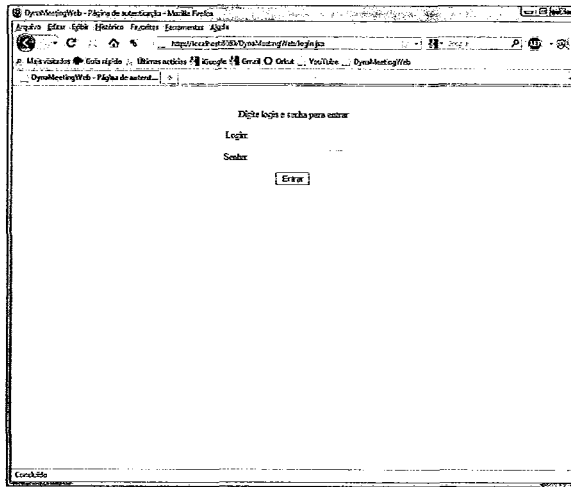


Figura 3-12 Interface de administração

Através dessa interface é possível que (mediante uma autenticação) qualquer usuário autorizado realize alterações nas informações cadastradas de *ThinkLets*, padrões colaborativos, tipos de resultados, bem como fazer novos cadastros. Usuários administradores podem ainda alterar informações de processos construídos na ferramenta, como o seu título, sua descrição e o usuário proprietário.

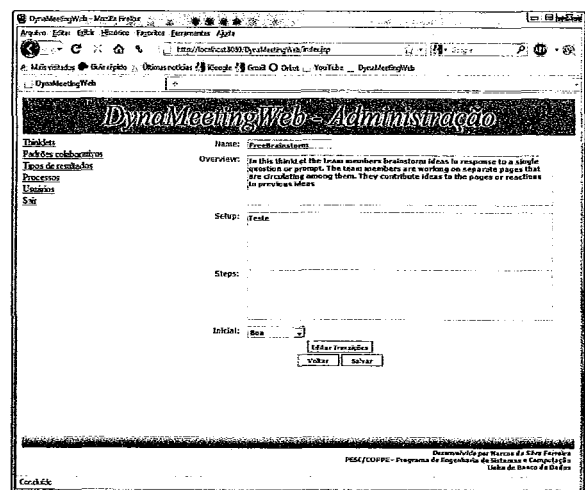
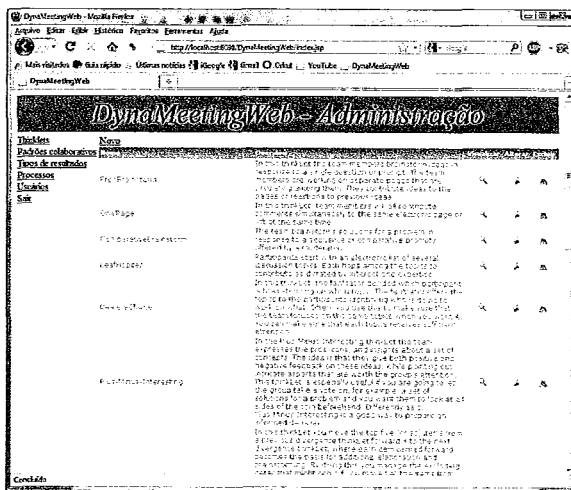


Figura 3-13 Cadastro de *ThinkLets*

Através do cadastro de *ThinkLets*, é possível incluir qualquer novo *ThinkLet* que não era conhecido no momento de implementação, de forma que a ferramenta é

facilmente estendida. Para o cadastro de *ThinkLets* (Figura 3-13) é possível incluir as seguintes informações: nome, visão geral, configuração, descrição dos passos para execução e um grau de aplicabilidade como a primeira atividade em um processo colaborativo (Impossível, Possível ou Boa).

Através do cadastro de padrões colaborativos e tipos de resultados (Figura 3-14), o administrador pode determinar o relacionamento entre essas entidades e os *ThinkLets*, selecionado para cada um os *ThinkLets* que correspondem a ele. Através do ajuste desse mapeamento o comportamento da ferramenta será modificado.

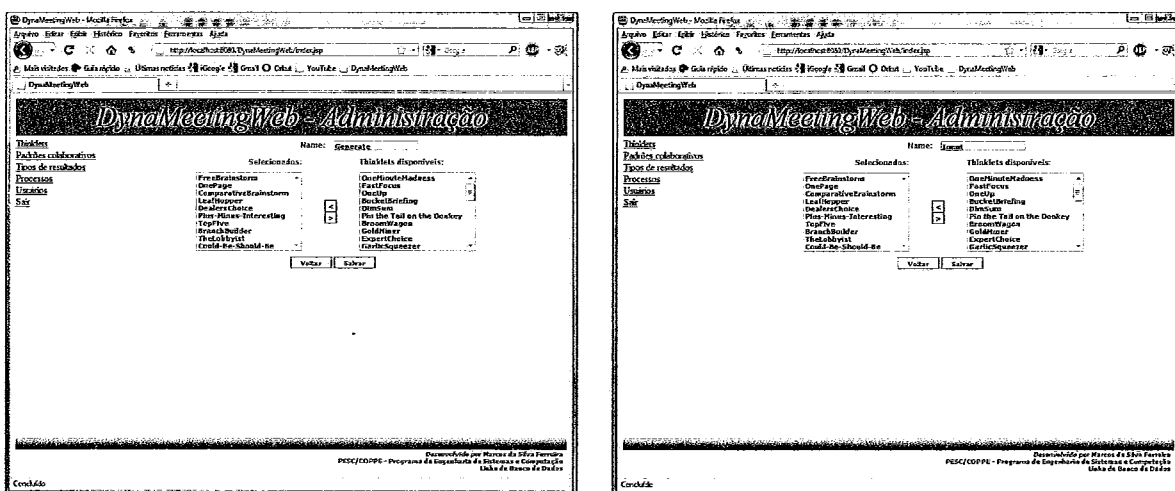


Figura 3-14 Cadastro de padrões colaborativos e tipos de resultado

Além disso, é possível editar através desse cadastro informações que irão determinar parte do comportamento da ferramenta com respeito às recomendações. Como podemos ver na Figura 3-15, é possível definir a adequabilidade da utilização de um determinado *ThinkLet* após a execução de cada *ThinkLet*, ou seja, a compatibilidade entre eles. A disponibilização dessa funcionalidade na interface de administração visa a tornar a ferramenta o mais flexível e adaptável possível. Facilitadores mais experientes podem configurar a ferramenta da forma mais apropriada de forma a permitir que ela seja usada por facilitadores menos experientes.



Figura 3-15 Cadastro de compatibilidade entre *ThinkLets*

4 Implementação

Este capítulo apresenta a arquitetura técnica da solução desenvolvida, bem como apresenta as principais tecnologias utilizadas na construção. Este capítulo aborda ainda o planejamento de um experimento, visando a validação da ferramenta em relação a sua proposta.

4.1 A arquitetura do *AgendaBuilder*

Nesta seção iremos apresentar uma visão geral da arquitetura proposta para a ferramenta, explicitando seus principais módulos e suas finalidades, bem como as principais tecnologias escolhidas para sua implementação. Na Figura 4-1 podemos ver uma visão geral da arquitetura da ferramenta.

A arquitetura da ferramenta desenvolvida se apóia em uma divisão de três camadas, seguindo o padrão clássico MVC (Modelo-Visão-Controle). A seguir, serão analisadas cada camada e detalhes sobre sua implementação e tecnologias utilizadas.

Na camada mais inferior (Modelo), temos o repositório onde ficam armazenadas tanto as informações relacionadas aos *ThinkLets*, quanto as informações referentes a cada processo de reunião criada por um usuário. O modelo de dados do repositório pode ser encontrado no Apêndice E.

Para a implementação do repositório foi utilizado como banco de dados o MySQL 5.0, em sua versão gratuita. O MySQL se tornou um dos mais populares bancos de dados do mundo, tendo sido adotado largamente em soluções de pequeno porte e até mesmo em grandes corporações. Entre as razões para sua adoção neste trabalho, podemos citar o fato de ter disponibilidade gratuita; simplicidade de configuração,

utilização e administração; facilidade de conexão através de JDBC; e não necessita de investimentos em *hardware*, funcionando a contento em máquinas de pequeno porte (MySQL, 2009).

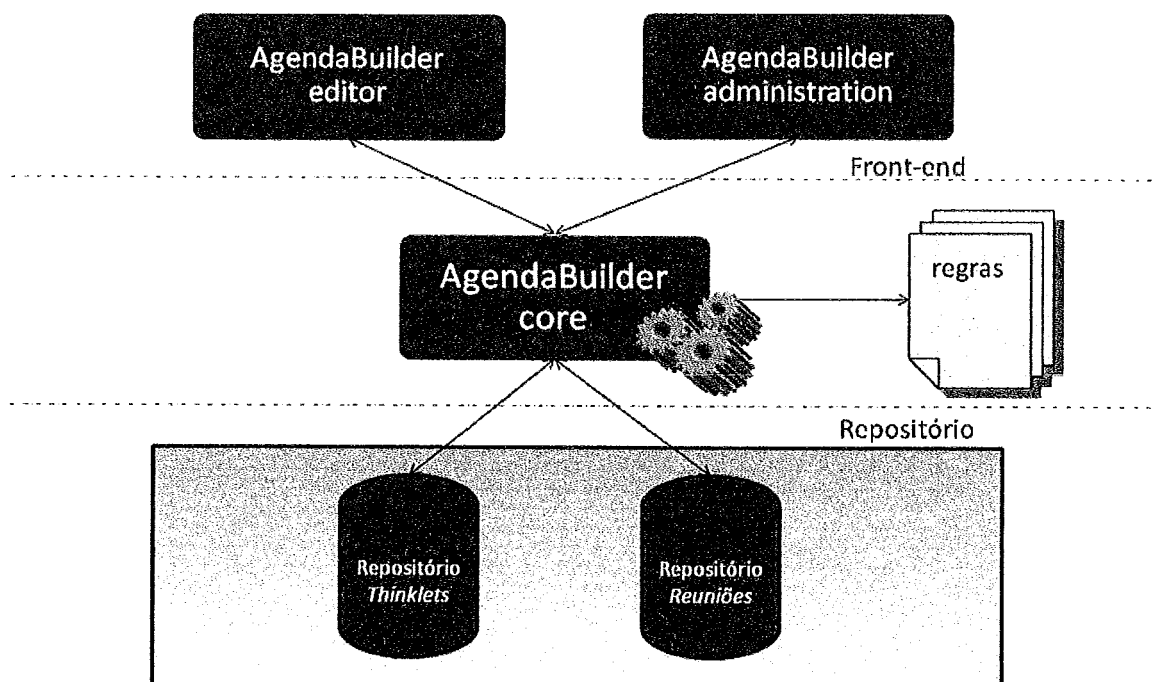


Figura 4-1 Arquitetura do *AgendaBuilder*

Na camada central, temos toda a lógica de processamento da ferramenta. Essa é a camada responsável por orquestrar todo o seu funcionamento, além de dispor os recursos de inteligência para suporte ao facilitador. Seu componente central é o *AgendaBuilder core*, que fornece as recomendações de *ThinkLets* através do processamento dos mapeamentos disponíveis no repositório e as regras pré-definidas, que são parte também da camada central.

O componente *AgendaBuilder core* possui os seguintes recursos:

- Classes e interfaces para acesso a camada de banco de dados. Para o acesso ao banco de dados foi utilizado JDBC juntamente com o

framework Hibernate. Os componentes de front-end utilizam essas interfaces para fazer toda a persistência necessária em banco de dados.

- Classes do tipo JavaBeans que espelham e representam as entidades do modelo de dados. Os componentes de front-end manipulam essas classes para fazer todo o processamento de dados necessário em memória.
- Implementa o mecanismo chamado de *ThinkLet Engine*, visto na Figura 3-5, responsável por processar as regras e mapeamentos e resolver as recomendações de acordo com o contexto.

Para processar as recomendações, o mecanismo *Thinklet Engine* se vale dos seguintes insumos: as informações contextuais obtidas a partir da interface do *AgendaBuilder editor* com o usuário, os mapeamentos mantidos em banco de dados e as regras de recomendação (representada na Figura 4-1 por um conjunto de arquivos na camada central). O processamento realizado pelo mecanismo se dá sequencialmente para cada atividade do processo, da seguinte forma:

1. Processa uma lista de *ThinkLets* associados ao padrão colaborativo selecionado (se algum foi selecionado) pelo usuário para a atividade, através do mapeamento mantido no repositório.
2. Processa uma lista de *ThinkLets* associados ao tipo de resultado selecionado (se algum foi selecionado) pelo usuário para a atividade, através do mapeamento mantido no repositório.
3. Processa uma lista de *ThinkLets* apropriados de acordo com a lista de *ThinkLets* precedentes dentro do processo (caso haja algum), através do mapeamento mantido no repositório.
4. Processa a base de regras utilizando todas as informações contextuais definidas pelo usuário (quantidade de participantes, tipo de entrada do processo, etc), obtendo uma lista de *ThinkLets* aderentes de acordo com as regras.

5. Processa todas as listas criadas de forma a realizar uma mesclagem em uma lista final que deverá conter apenas os *ThinkLets* que atendam todos os critérios acima trabalhados.

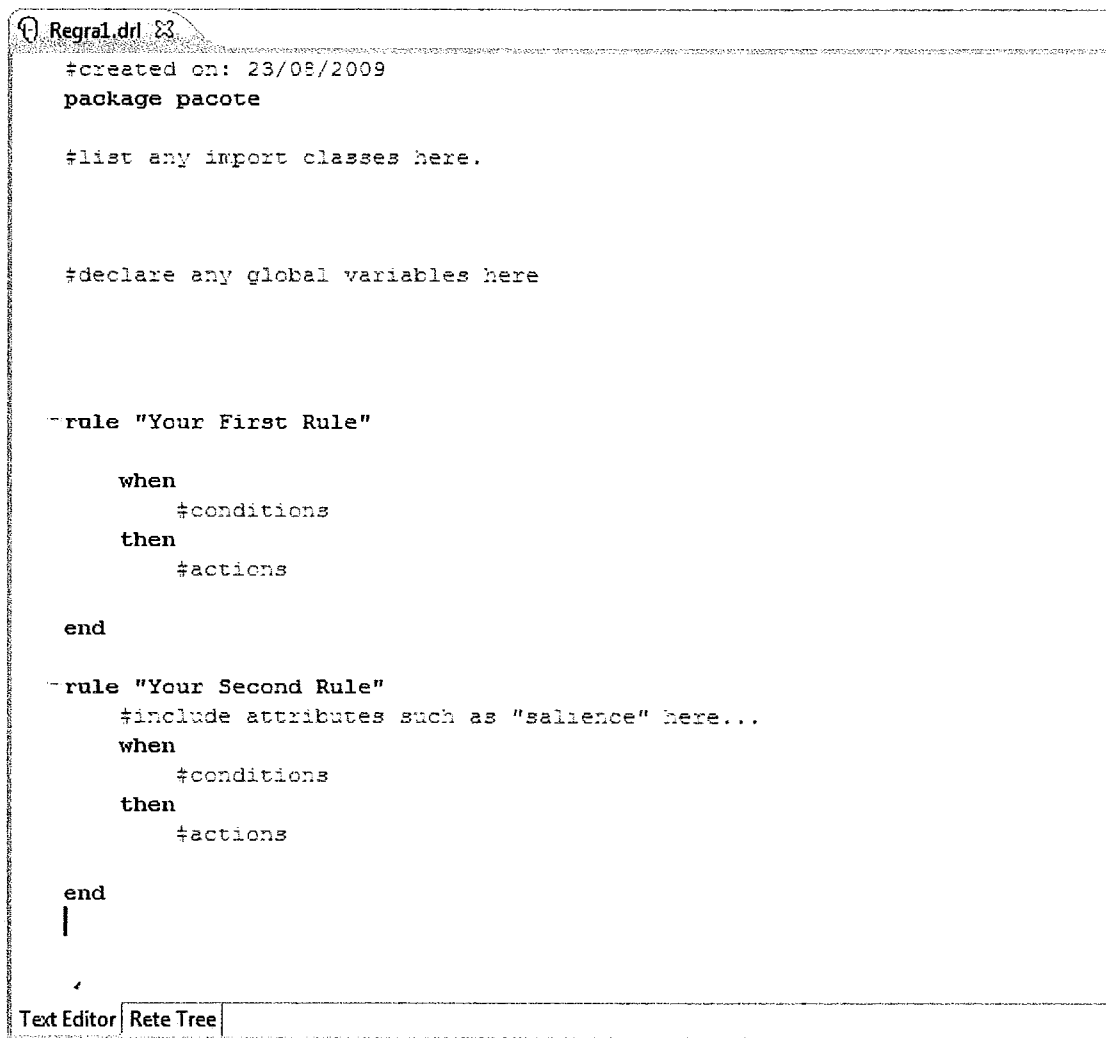
Após esse processamento, a lista resultante e as intermediárias são devolvidas para a camada de front-end. Nela o usuário irá visualizar a lista resultante mesclada e, caso deseje, poderá também visualizar separadamente cada uma das listas de recomendações intermediárias.

A base de regras foi desenvolvida utilizando o framework *Drools*, desenvolvido pela IBM, que provê uma interface integrada para regras, *workflow* e processamento de eventos. A adoção desse framework permite desenvolver, manter, versionar, testar, organizar e implantar regras de negócio, de forma desacoplada, permitindo que o comportamento de um sistema seja facilmente estendido simplesmente através da inclusão de uma nova regra.

O desenvolvimento com o *Drools* é bem simples e consiste em escrever arquivos de regras, como podemos ver na Figura 4-2. Para cada regra, é definida uma condição lógica que, quando satisfeita, irá disparar as ações associadas. Para a implementação das ações foi utilizada a própria linguagem Java. Para este trabalho foram criados os seguintes arquivos de regras (os códigos dos arquivos de regras podem ser vistos em detalhes em Apêndice F, Apêndice G e Apêndice H):

1. ***GroupProfile.drl***: Avalia o tipo de perfil do grupo de participantes definido pelo facilitador e, de acordo com a situação, define a recomendação para os *Thinklets LeafHopper* e *DealersChoice*.
2. ***GroupSize.drl***: De acordo com a quantidade de participantes definida pelo facilitador, define a recomendação relacionada aos *Thinklets FreeBrainstorm* e *OnePage*.
3. ***ProcessInput.drl***: De acordo com a entrada do processo colaborativo, define a recomendação relacionada aos *ThinkLets*

ComparativeBrainstorm, FreeBrainstorm, OnePage, LeafHopper e DealersChoice.



```
Regra1.drl 83
#created on: 23/09/2009
package pacote

#list any import classes here.

#declare any global variables here

--rule "Your First Rule"

    when
        #conditions
    then
        #actions
    end

--rule "Your Second Rule"
    #include attributes such as "saliency" here...
    when
        #conditions
    then
        #actions
    end

end
|
/

Text Editor | Rete Tree
```

Figura 4-2 Template de um arquivo de regras no Drools

Na camada de *front-end* temos dois componentes: *AgendaBuilder editor* e *AgendaBuilder administration*. O *AgendaBuilder editor* é a interface gráfica responsável por toda a construção do processo colaborativo. Todas as telas analisadas na seção 3.6.1.1 são implementadas através deste componente. O componente *AgendaBuilder administration* implementa toda a interface de administração da ferramenta, analisada na seção 3.6.1.2.

Para o componente *AgendaBuilder editor* foi utilizada a linguagem de programação Java e a interface foi totalmente construída através dos frameworks SWING e SWT (Java para desktop). O desenvolvimento foi feito através da IDE Eclipse, com parte desenvolvida em NetBeans, que conta com suporte para desenvolvimento de interfaces desktop. Para a edição e visualização gráfica dos fluxogramas do processo colaborativo foi utilizada a biblioteca JGraph (*Java Graph Visualization and Layout*) (JGRAPH, 2009), uma biblioteca Java gráfica, gratuita, para manipulação e visualização de grafos. Entre as vantagens e características oferecidas por essa biblioteca, pode-se destacar:

- Desenvolvida 100% em Java, integrada com a hierarquia de componentes Swing;
- Completamente compatível com padrões de codificação Java (Java Code Conventions);
- Completamente documentada com diversos exemplos que demonstram suas características;
- Provê funcionalidades de zoom, desfazer modificações, drag and drop, entre outras.

O desenho do processo elaborado na ferramenta é convertido em um arquivo xml, que codifica toda a disposição gráfica dos componentes, bem como as referências para os objetos de negócio associado às atividades. Esse conteúdo é persistido em banco de dados juntamente com as demais informações referentes ao processo.

O componente *AgendaBuilder administration* foi desenvolvido como uma interface web, e pode ser utilizada em qualquer *browser*. A implementação foi feita em Java J2EE, na IDE Eclipse, utilizando o framework Struts (versão 1.3) para processar requisições do usuário e para definir o layout do conteúdo (através dos recursos Tiles e TagLib). O servidor de aplicação utilizado foi o Apache Tomcat (em sua versão 6.0).

4.2 Plano de experimento

Nesta seção será apresentado um planejamento e metodologia para a realização de um experimento de validação da ferramenta.

4.2.1 Planejamento

A metodologia adotada para o experimento visa dois pontos básicos: demonstrar a vantagem da utilização de uma ferramenta de modelagem de fluxos para a atividade de construção de um processo colaborativo; a importância do suporte da ferramenta nas escolhas relacionadas à construção do processo, de tal forma que o facilitador não precise ter conhecimentos muito avançados sobre *ThinkLets* e processos colaborativos.

Para o experimento foram preparadas algumas tarefas de construção de um processo colaborativo para resolver um determinado problema. Essas tarefas devem ser distribuídas entre três grupos distintos de pessoas, da seguinte forma:

1. **Grupo 1:** Deve realizar a tarefa sem nenhum suporte da ferramenta computacional. Para realizar a atividade, deve ser fornecido ao participante apenas a documentação disponível sobre os *ThinkLets*. Com esse material, o participante pode realizar consultas sobre cada *ThinkLet* e sua aplicabilidade, de tal forma a poder fazer as seleções apropriadas para o processo sendo construído.
2. **Grupo 2:** Deve realizar a tarefa utilizando suporte da ferramenta computacional. No entanto, a ferramenta deve ser configurada para fornecer apenas as funcionalidades gráficas para montagem do processo e de seleção dos *ThinkLets* a partir da base de dados. A ferramenta, neste cenário, não irá realizar nenhuma recomendação de *ThinkLets*, deixando o processo de escolha totalmente a cargo do participante.
3. **Grupo 3:** Deve realizar a tarefa utilizando suporte completo fornecido pela ferramenta computacional. Neste cenário, todas as funcionalidades

de inteligência da ferramenta são ativadas para auxiliar o participante na escolha apropriada do *ThinkLet* de acordo com as informações contextuais do processo.

Cada participante deve responder a dois questionários, um previamente à realização do experimento e outro posterior. O primeiro questionário visa coletar informações gerais e o perfil do participante (profissional, experiências prévias com processos colaborativos e facilitação).

Após a realização do experimento, é respondido um questionário que visa a realizar uma avaliação do processo, nas seguintes dimensões: Processos colaborativos, *ThinkLets*, Facilitação, Suporte a facilitação, Ferramenta. Cada dimensão tem os seguintes objetivos:

1. Processos colaborativos: Identificar o ponto de vista do participante sobre a importância dos processos colaborativos para tomada de decisão e resolução de problemas;
2. Thinklets: Obter uma avaliação sobre a importância dos Thinklets como ferramentas para construção de processos colaborativos, na medida em que fornecem padrões colaborativos bem definidos, conhecidos e reproduzíveis;
3. Facilitação: Identificar o ponto de vista do participante quanto à importância do papel do facilitador na preparação e condução dos processos colaborativos;
4. Suporte a facilitação: Identificar a importância de mecanismos que forneçam um suporte adequado na atividade do facilitador na preparação de um processo colaborativo, independente do seu nível de experiência.
5. Ferramenta: Avaliação aplicada apenas para o Grupo 2 e Grupo 3, visando coletar um *feedback* sobre a utilização da ferramenta, focando nos seguintes aspectos:

- a. Aderência aos requisitos propostos
- b. Usabilidade
- c. Relevância para o facilitador

4.2.2 Análise dos Resultados

A análise dos resultados deverá focar em dois aspectos: quantitativo e qualitativo. No aspecto quantitativo serão analisados indicadores colhidos a partir da realização, bem como serão tabulados e analisados os resultados dos questionários preenchidos pelos participantes. Para o aspecto qualitativo, devem ser analisados os comentários livres dos questionários e, mais importante, devem ser analisados os processos colaborativos obtidos pelos participantes de cada grupo, quanto a sua qualidade.

4.2.2.1 Análise quantitativa

Na análise quantitativa será medido o desempenho dos participantes de cada grupo, com o objetivo de compararmos em relação a utilização da ferramenta. Para isso, serão colhidos durante a realização do experimento, tais como tempo gasto na conclusão da tarefa e quantidade de manifestação de dúvidas por parte dos participantes.

Os resultados serão tabulados por grupo e por participante, conforme Tabela 4-1, abaixo. Os resultados serão também traçados em um gráfico.

4.2.2.1 Análise qualitativa

Para a análise qualitativa deverá ser avaliada a qualidade do processo colaborativo desenhado pelos participantes de cada grupo. Essa análise deverá ser conduzida por um especialista em facilitação com experiência razoável no planejamento e execução de processos colaborativos.

Tabela 4-1 Template para tabela de análise quantitativa

	Tempo para conclusão	Manifestações
Grupo 1	Tempo médio (min.)	Total (soma)
Participante 1	Duração em minutos	Nº de manifestações
...		
Grupo 2	Tempo médio (min.)	Total (soma)
Participante 4	Duração em minutos	Nº de manifestações
...		
Grupo 3	Tempo médio (min.)	Total (soma)
Participante 7	Duração em minutos	Nº de manifestações
...		

Para essa análise, o especialista deverá levar em consideração a adequação do processo desenhado de acordo com os objetivos pretendidos e com o contexto determinado.

Além da análise realizada pelo especialista, deverão ser analisados os comentários livres feitos pelos participantes, de forma a identificar possíveis pontos de melhoria e os pontos fortes da ferramenta.

4.3 Análise crítica de par

Nesta seção é abordada uma análise realizada através da apresentação da ferramenta e seu funcionamento para uma pessoa envolvida com a área de Engenharia da Colaboração. Essa pessoa realizou uma rápida avaliação crítica através de um questionário que será apresentado aqui.

A pessoa escolhida para realizar a análise crítica possui atuação prévia na área de Engenharia da Colaboração. Mais especificamente, a pessoa atuou desenvolvendo um dos módulos da ferramenta *DynaMeetings*. Após a apresentação da ferramenta e das

funcionalidades de recomendação, foi solicitado que fossem respondidas três perguntas abordando os principais aspectos da proposta, que seguem:

1. Você considera que o sistema de recomendação implementado auxilia de fato a atividade de escolhas nos planejamento de uma reunião com thinkLets?

R.: Sim, pois com a recomendação, os facilitadores terão maior facilidade durante o planejamento da reunião. Isto porque a ferramenta mostra todas os prós e os contras de cada thinkLet que pode ser usado a cada fase da reunião.

2. O que você considera que poderia ser melhorado ou incluído na ferramenta para um maior suporte no planejamento da reunião com thinkLets?

R.: A ferramenta poderia dar um suporte ao replanejamento da reunião durante sua execução; ou até mesmo deixar que o planejamento do próximo thinkLet a ser usado seja escolhido ao final da execução da fase anterior. Uma melhora na interface, tornando-a mais intuitiva a usuários menos experientes também é importante.

3. De uma forma geral, você avaliaria positivamente a ferramenta e a adotaria no planejamento de reuniões com thinkLets?

R.: Sim. A ferramenta possui um grande potencial se associada aos outros módulos que estão sendo e já foram implementados do DynaMeetings, cujo objetivo é o suporte à reuniões facilitadas.

5 Conclusão

No processo de tomada de decisão, faz-se cada vez mais necessária a participação de uma pessoa que auxilie os participantes, conduzindo a reunião de forma que os melhores resultados possíveis sejam alcançados. Este é o papel do facilitador, uma pessoa especializada em auxiliar os grupos neste processo e oferecer todo o suporte necessário durante todas as fases da reunião.

Neste trabalho, foram analisadas lacunas que ferramentas atuais possuem no que tange o suporte às atividades desempenhadas por um facilitador, principalmente na etapa de planejamento de um processo colaborativo. Nesse contexto, foi proposta uma abordagem, baseada na utilização de *ThinkLets*, de uma ferramenta que se destina a auxiliar o facilitador fornecendo sugestões a partir das informações contextuais e de regras pré-definidas.

A ferramenta foi desenvolvida com uma interface gráfica que permite ao usuário construir o desenho de um processo de forma bastante intuitiva. A ferramenta permite também a inserção das informações contextuais que precisam ser previamente identificadas pelo facilitador.

5.1 Trabalhos futuros

Inicialmente, este trabalho visava fornecer suporte tanto para a fase de planejamento quanto para a fase de acompanhamento. No entanto, ao longo de seu desenvolvimento esse escopo foi reduzido, ficando o acompanhamento para ser abordado através de outro módulo do projeto *DynaMeetings*. Apesar disso, algumas possibilidades para a abordagem necessária para o suporte a execução e acompanhamento foram levantadas e serão discutidas a seguir.

Ao permitir ao facilitador construir um processo colaborativo baseado em *ThinkLets*, a ferramenta *AgendaBuilder* agrega valor na etapa de planejamento e reduz o tempo necessário para esta etapa, na medida que traz uma redução de esforço cognitivo através de sugestões direcionadas a partir de informações contextuais e regras pré-definidas. A ferramenta *FacilitAid*, outro módulo do sistema *DynaMeeting* (MARQUES, L.C., 2009), fornece suporte ao facilitador (e também aos participantes) durante a realização de cada atividade do processo, fornecendo indicadores do andamento de forma a dar subsídios para que o facilitador reconheça quando é necessário intervenção. No entanto, a condução do processo como um todo representa ainda uma lacuna. Não há um suporte ao facilitador para as transições entre uma atividade e a seguinte (entre a execução de cada *ThinkLet*). Como identificar necessidades de ajustes no processo após a realização de cada atividade?

O processo colaborativo desenhado inicialmente representa um guia inicial para a realização da reunião, mas ele é baseado apenas nas informações contextuais conhecidas pelo facilitador **previamente** à realização da reunião. A partir de uma ou mais rodadas de atividades, a situação conhecida inicialmente pode apresentar alterações substanciais, demandando que o planejamento inicial seja revisto.

Para ilustrar essa possibilidade, suponhamos a seguinte situação: um facilitador prepara um processo colaborativo visando guiar um grupo de 5 profissionais de uma empresa de TI a buscar idéias para redução de custos no seu processo de desenvolvimento de *software*, ao mesmo tempo entregando soluções de maior qualidade. A partir da análise da necessidade e do perfil dos participantes, o facilitador inicia a elaboração do processo, considerando uma primeira etapa de geração de idéias que utilizará o *ThinkLet OnePage*. A utilização deste *ThinkLet* é viável, pois estamos lidando com um grupo pequeno e homogêneo, então não é esperada uma quantidade muito grande de idéias ou muita divergência. Assim, imediatamente após essa primeira etapa, o facilitador estabelece que já será realizada uma etapa de organização, com um *RichRelations* seguido de um *PopcornSort* e o processo seguirá então para as etapas de avaliação e construção de consenso, para gerar o resultado final.

Após a execução da etapa de geração, diferentemente do esperado inicialmente, é constatado que foram levantadas algumas idéias polêmicas e que geraram uma enxurrada de comentários relacionados e muita divergência no grupo. Nesse novo cenário, é altamente recomendável que não se siga diretamente para uma etapa de organização, pois o conjunto a ser trabalhado é grande e diverso, sendo mais recomendável realizar uma etapa intermediária de convergência e redução. Facilitadores experientes possuem maior capacidade para identificar alterações de cenário e corrigir rapidamente o rumo do processo. Facilitadores menos experientes podem não ser muito bem sucedidos nesse quesito, comprometendo o resultado final.

A partir do estudo feito, poderia ser dado um suporte ao facilitador durante as transições entre atividades, com uma análise quantitativa feita através da ferramenta após cada etapa finalizada. Para isso, pode-se utilizar como insumos os próprios indicadores sugeridos por Marques, L.C. (2009), bem como novos indicadores que se mostrarem relevantes. Algumas possibilidades são:

- Quantidade de idéias/comentários: a análise da quantidade de idéias pode fornecer uma percepção direta do volume de informação que será tratada em uma etapa posterior, podendo indicar necessidade de incluir etapas de redução caso elas não tenham sido previstas inicialmente.
- Nível de discussão: identificando idéias que geram mais polêmica, pode se selecionar essas idéias para a realização de uma etapa intermediária de elaboração ou entendimento, visando facilitar a convergência do grupo posteriormente.
- Conflito interpessoal: indentificando um elevado conflito interpessoal em alguma etapa da reunião, a ferramenta pode recomendar ao facilitador a realização de um *CheckMark*, caso essa atividade não estivesse prevista no planejamento inicial. Através desse *ThinkLet*, é possível buscar e focar nos pontos em comum entre os membros de um grupo polarizado, tirando o foco das discordâncias.

A ferramenta também poderia ser aprimorada a partir da implementação e inclusão de novas regras de recomendação, que podem ser desenvolvidas utilizando o *framework Drools*, conforme detalhado no capítulo 4. Para isso, podem ser selecionados, formalizados e codificados alguns dos critérios definidos por Briggs et al (2001). Esse conjunto de critérios pode ser encontrado também na seção de anexos deste trabalho. Para citar alguns exemplos, podemos ter as seguintes situações:

- Escolha de um *ThinkLet* para uma etapa de categorização das idéias geradas e com restrições de tempo, sendo necessário que as atividades sejam rápidas sempre que possível. Para categorização, tem-se como sugestões os *ThinkLets Concentration, ThemeSeeker, RichRelations, PopcornSort, ChauffeurSort, Evolution* e *Could-Be-Should-Be*. Analisando a tabela abaixo (Tabela 5-1), nota-se que *ChauffeurSort* e *Evolution* são contra-indicados em situações com restrições de tempo, pois requerem uma análise mais cuidadosa. Assim, é possível codificar facilmente uma regra através da utilização do *Drools* para restringir a utilização desses dois *ThinkLets*, dado um determinado *threshold* para o tempo disponível, removendo-os da lista de recomendações. Os demais *ThinkLets* de categorização não apresentam contra-indicações explícitas relacionadas a tempo, então podem continuar a ser considerados como sugestões válidas pela ferramenta.

Tabela 5-1 Critérios para SevenUp

ChauffeurSort	SIM	Quando for necessário garantir que a categorização foi analisada cuidadosamente pelo grupo
		Quando criar um entendimento comum for tão importante quanto a categorização em si
	NÃO	Quando o tempo for crítico. Uma sequência de um PopcornSort e BucketWalk é muito mais eficiente.
		Quando a categorização for simples e não requerer discussões
Evolution	SIM	Para achar possíveis categorias para organizar as idéias e elas não forem óbvias
	NÃO	Para encontrar categorias óbvias Se for necessário fazer uma organização rápida.

- Escolha de um *ThinkLet* para uma etapa de avaliação e votação com restrições de tempo. Para avaliação, tem-se como sugestões os *ThinkLets StrawPoll, BucketWalk, BucketShuffle, MultiCriteria, CheckMark, StakeholderPoll, BucketVote, SevenUp*. Como consta na tabela (Tabela 5-2), *SevenUp* é explicitamente contra-indicado em circunstâncias de tempo restrito, cabendo da mesma forma uma regra baseada em um *hreshold* para removê-lo da lista de recomendação.

Tabela 5-2 Critérios para SevenUp

SevenUp	SIM	Quando uma atividade de brainstorming gerar centenas de idéias e for desejável extrair o conjunto mais promissor.
		Quando desejar tomar nota e talvez discutir as causas das discordâncias, mas após focar nas concordâncias para seguir para os próximos passos.
		Quando desejar ter certeza de convergir em até dois passos.
	NÃO	Quando quiser que o grupo foque apenas nos seus pontos comuns. Quando o tempo for curto.

Pode se fazer necessário também modificar a implementação em nível de código caso seja necessário que a ferramenta manipule novas informações contextuais não contempladas nessa primeira versão. A partir da inclusão do tratamento de novas informações contextuais, torna-se possível a inclusão de novas regras e critérios que não são possíveis de serem tratados atualmente.

Partindo do nível funcional da ferramenta para o nível da arquitetura técnica, uma lacuna significativa é quanto à escolha da plataforma Java para *desktop* na implementação do *AgendaBuilder*. A escolha se deu devido a maior facilidade para construção de interfaces gráficas mais complexas em tal plataforma, através da IDE do NetBeans. Como o propósito do sistema *DynaMeetings* é ser uma plataforma *web* totalmente acessível sem necessidade de instalações e configurações, faz-se necessário uma migração da interface *AgendaBuilder* para a *web*. Para isso, podem ser utilizadas a tecnologia de *applets* ou, talvez o mais apropriado por apresentar mais flexibilidade, bibliotecas gráficas para construir interfaces ricas na *web*, como por exemplo, AJAX ou YUI Library (Yahoo User Interface Library).

Ainda resta realizar uma integração efetiva com o módulo *FacilitAid*, que não ocorreu por dificuldades de alinhamento. No entanto, a integração dos dois módulos não representa grandes dificuldades técnicas, até mesmo porque ambos representam momentos bem distintos no ciclo de vida de uma reunião. Para fazer a ponte entre os dois módulos, precisará ser desenvolvido o módulo de apoio e ajuste da execução do processo. Para tal, bastaria a princípio integrar no repositório de *ThinkLets* do *AgendaBuilder* as referências para as URLs responsáveis pela execução de cada um na ferramenta *FacilitAid*.

Espera-se que o desenvolvimento desta ferramenta traga ganhos significativos para o planejamento de reuniões eletrônicas para tomada de decisão e, inserida no contexto do projeto *DynaMeetings*, constitua um importante sistema para a comunidade de CSCW.

Referências Bibliográficas

- ANDRIESSEN, J.H.E., 2003. "Working with Groupware". London: Springer-Verlag.
- ANTUNES, P., HO, T. and L. CARRIÇO, L., 1999. "A GDSS Agenda Builder for Inexperienced Facilitators", Proceedings of the 10th EuroGDSS Workshop. G. Vreede and F. Ackermann, (eds.) Copenhagen, Denmark: Delft University of Technology, 1999, pp. 1-15.
- ANTUNES, P., HO, T., 1999. "Developing a Tool to Assist Electronic Facilitation of Decision-Making Groups", Fifth International Workshop on Groupware, CRIWG '99. IEEE.
- ANTUNES, P., HO, T., 2001. "The Design of a GDSS Meeting Preparation Tool", Group Decision and Negotiation, vol. 10, no. 1, pp. 5-25, January 2001.
- BASILI, V., CALDIERA, G., ROMBACH, D.H., 1994. "The goal question metric approach", In Encyclopedia of Software Engineering, J. Marciniak, Ed. Wiley, 1994.
- BEAUDOUIN-LAFON, M., 1999. *Computer Supported Co-operative Work*. 1 ed. Paris, Wiley.
- BRIGGS, R.O., DE VREEDE, G.J., 2005. "Thinklets: Building Blocks for Concerted Collaboration".
- BRIGGS, R.O., de VREEDE, G.J., NUNAMAKER Jr., J.F., DAVID, T.H., 2001. "Thinklets: Achieving Predictable Patterns of Group Interaction with Group Support Systems". Hawaii International Conference on System Sciences. IEEE Computer Society Press, Los Altos, pp. 9, 2001.
- BRIGGS, R.O., DE VREEDE, G.J., NUNAMAKER JR., J.F., 2003. "Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems", Journal of Management Information Systems 19 (4), 2003, pp 31- 63.
- BRIGGS, R.O., de VREEDE, G.J., KOLFSCHOTEN, G.L., DEAN, D.L., 2006. "Defining Key Concepts for Collaboration Engineering". Americas Conference on Information Systems, Acapulco, México: AIS, 2006.

- BURSTEIN, F., HOLSAPPLE, C.W., 2008. "Handbook on Decision Support Systems", editora Springer, pp. 347, 2008.
- CATTUTO C., BALDASSARRI A., SERVEDIO VDP, LORETO V., 2007. "Vocabulary Growth in Collaborative Tagging Systems". In European Conference on Complex Systems", Dresden, 2007.
- CLAWSON, V. K., BOSTROM, R. P., 1993. "The facilitation role in group support systems environments", Proceedings of the 1993 conference on Computer personnel research, St Louis, Missouri, United States, pp. 323-335, 1993.
- DAVISON, R.M., 2001. "Socio-Psychological Aspects of Group Processes" City University of Hong Kong, China. Sprouts: Working Papers on Information Systems, vol. 1, no.13, 2001.
- DAVISON, R., QURESHI, S., VREEDE, G.J. DE, VOGEL, D.R., JONES, N., 1998. "Group Support Systems through the Lens of Action Research: Cases in Organisations", Working Paper Series, City University Hong Kong, 1998.
- DESANCTIS, G., GALLUPE, B., 1987, "A Foundation for the Study of Group Decision Support Systems", Management Science, vol. 33, no.5, pp.589-609, 1987.
- DOYLE, M., STRAUS, D., 1985. "How to make meetings work". Berkeley Publishing Group, New York, 1985.
- DROOLS, 2009. Disponível para *download* em <http://jboss.org/drools/>, último acesso em 23 de agosto de 2009.
- Eclipse IDE, 2009. Disponível para *download* em <http://www.eclipse.org/>, último acesso em 10 de janeiro de 2009.
- EHRlich, K, 1999, "Designing Groupware Applications: A Work-Centered Design Approach". In: Beaudouin-Lafon, M. (eds), *Computer Supported Co-operative Work*, 1 ed., cap. 1, Paris, Wiley.
- ELLIS, C., WAINER, J., 1994. "A conceptual model of groupware", Proceedings of the ACM conference on Computer Supported Cooperative Work, pp.79-88, Chapel Hill, North Carolina, United States, 1994.

- ENGELBART, D.C, 1962. "Augmenting human intellect: A conceptual framework" (Summary Report, AFOSR-3233) Menlo Park, CA: SRI International, 1962.
- FEENBERG, A., 1986. "Network design: an operation manual for computer conferencing". Proceedings of the Conference on Computer-Supported Cooperative Work, ACM, Austin, Texas, 1986.
- GOSLING, J., JOY, B., STEELE, G., et al., 2005. "The Java Language Specification". 3^a ed., Prentice Hall PTR.
- GALLUPE, R.B., DENNIS, A.R., COOPER, W.H., VALACICH, J.S., BASTIANUTTI, L.M., NUNAMAKER Jr., J.F. Electronic brainstorming and group size. Academy of Management Journal, 35(2), 1992, 350-369.
- GROHOWSKI, R., MCGOFF, C., VOGEL, D.R., MARTZ, B., & NUNAMAKER, J.F., 1990. "Implementing Electronic Meeting Systems at IBM: lessons learned and success factors". MIS Quarterly, vol.14, no. 4, pp. 368-383, 1990.
- GROUPSYSTEMS, 2009. Disponível em <http://www.groupsystems.com/>, último acesso em: 14 de fevereiro de 2009.
- GRUDIN, J., 1994, "Computer-Supported Cooperative Work: History and Focus", *IEEE Computer*, v. 27, n. 5 (May), pp. 19 – 26.
- HO, T., 1999. "Ferramentas de Suporte à Facilitação em Processos de Decisão em grupo", Dissertação de mestrado em Engenharia Electrotécnica e de Computadores, Universidade Técnica de Lisboa, 1999.
- JAVA, 2009. Disponível para *download* em <http://java.sun.com>, último acesso em: 15 de fevereiro de 2009.
- JGRAPH, 2009. Disponível para *download* em <http://www.jgraph.com/>, último acesso em: 23 de agosto de 2009.
- JOHANSEN, R., 1988, *GroupWare: Computer Support for Business Teams*. 1 ed. New York, The Free Press.

KOLFSCHOTEN, G.L., BRIGGS, R.O., APPELMAN, J.H., de VREEDE, G.J., 2004. "ThinkLets as Building Blocks for Collaboration Processes: A Further Conceptualization", 2004.

KOLFSCHOTEN, G., VREEDE, G.J., 2007. "The Collaboration Engineering Approach for Designing Collaboration Processes". In JM Haake, SF Ochoa, A. Cechich (Eds.) 13th Int. Workshop in Groupware (CRIWG 2007), LNCS 4715 pp.95-110, Berlin: Springer Verlag, 2007.

KOLFSCHOTEN, G.L., ROUETTE, E.A.J.A., 2006. "Choice Criteria for Facilitation Techniques". Hawaii International Conference on System Sciences (HICSS-39). IEEE Press.

KOLFSCHOTEN, G.L., VEEN, W., 2005. "Tool Support for GSS session design". Hawaii International Conference on System Sciences. IEEE Press, Los Alamitos.

MACAULAY, L.A., ALABDULKARIM, A., 2005. "Facilitation of e-Meetings: State-of-the-Art Review". Proc. 2005 IEEE Int. Conf. e-Technology, e-Commerce and e-Service (EEE'05), pp 728-735.

MANAGING MEETINGS, 2009. Disponível em http://managementhelp.org/grp_skill/meetings/meetings.htm, último acesso em 20 de janeiro de 2009.

MARQUES, L.C., 2009, *Uma ferramenta de auxílio ao facilitador de uma reunião eletrônica síncrona*. Tese de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.

MCGOFF, C., HUNT, A., VOGEL, D., NUNAMAKER, J., 1990. "IBM's experiences with GroupSystems". Interfaces, vol. 20, no.6, pp. 39-52, 1990.

MEETINGS, 2009. Guide to Facilitator. Disponível em http://www.cob.sjsu.edu/turner_m/meetings.htm, último acesso em: 20 de janeiro de 2009.

MEETINGWORKS, 2009 - Disponível em <http://www.meetingworks.com>, último acesso em 20 de janeiro de 2009.

- MOECKEL, A. "CSCW: conceitos e aplicações para cooperação". CEFET-PR, Curitiba, Jun. 2003.
- MySQL, 2008. Disponível para *download* em: <http://mysql.com/>, último acesso em: 22 de novembro de 2008.
- MySQL, 2009. <http://www.mysql.com/why-mysql/topreasons.html>, último acesso em: 22 de agosto de 2009.
- NetBeans IDE, 2009. Disponível para *download* em <http://www.netbeans.org/>, último acesso em 13 de junho de 2009.
- NUNAMAKER, J.F., DENNIS, A.R., VALACICH, J.S., VOGEL, D., GEORGE, J.F., 1991. "Electronic Meeting Systems", In Communications of the ACM, vol.34, no.7, p.40-61, 1991.
- NUNAMAKER, J., BRIGGS, R., MITTLEMAN, D., VOGEL, D., BALTHAZARD, P., 1996, "Lessons from a dozen years of Group Support systems research: a discussion of lab and field findings", In Journal of Management Information Systems, vol. 13, no. 3, pp. 163-207, 1996.
- NUNAMAKER, J. BRIGGS, R. VREEDE, G.J., 2001. "From Information Technology to Value Creation Technology". In Dickinson G.W., DeSanctis, G. (eds.) Information Technology and the Future Enterprise: new Models for Managers. New York: Prentice Hall, 2001.
- POST, B.Q., 1993. "A business framework for group support technology". Journal of Management Information Systems, vol. 9, no. 3, pp. 7-26, 1993.
- PRANTE , T., MAGERKURTH, C., STREITZ, N., 2002. "Developing CSCW tools for idea finding: empirical results and implications for design", Proceedings of the 2002 ACM conference on Computer supported cooperative work, New Orleans, Louisiana, USA, pp. 106-115, 2002.
- SCHMIDT, K., BANNON, L., 1992. "Taking CSCW Seriously: Supporting Articulation Work". Computer Supported Cooperative Work, vol. 1, no. 1, pp. 7-40, 1992.

- SCHWARZ, R., 1994. "The Skilled Facilitator". Jossey-Bass Publisher.
- SCHUTT, R. "Getting Unstuck: Common Problems in Meetings and Some Solutions", <http://www.vernalproject.org>, último acesso em 14 de Fevereiro de 2009.
- SIMON, H.A., 1955. "A behavioral model of rational choice". In The Quarterly Journal of Economics, vol 69, no 1 pp. 99-118.
- SIMON, H.A., 1959. "Theories of Decision Making in Economics and Behavioral Science". In The American Economic Review, vol 49, No 3 pp.253-283.
- TOMCAT, web applications container, 2009. Disponível para *download* em <http://tomcat.apache.org/>, ultimo acesso em: 15 de agosto de 2009.
- TALBOTT, S. L., 1995. "Thoughts on a Group Support System", chapter 10 - The Future Does Not Compute: Transcending the Machines in Our Midst. Copyright 1995 O'Reilly & Associates, 1995. Disponível em <http://www.netfuture.org/fdnc/ch10.html>, último acesso em 19 de março de 2009.
- TURBAN, E., ARONSON, J.E., 1998. "Decision Support Systems and Intelligent Systems". Prentice Hall.
- VILLER, S., 1991. "The Group Facilitator: A CSCW Perspective". Proceedings of the Second European Conference on Computer-Supported Cooperative Work.
- VIVACQUA, A.S., 2008, *Information Dissemination in Collaborative Environments*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- VIVACQUA, A.S., MARQUES, L.C., SOUZA, J.M. "Assisting Meeting Facilitation through Automated Analysis of Group Dynamics" In: International Conference on Computer Supported Cooperative Work in Design (CSCWD 2008), 2008, Xian.
- VIVACQUA, A.S.; MARQUES, L.C.; FERREIRA, M.S.; SOUZA, J.M.. "Information Needs for Meeting Facilitation". In: 14th Collaboration Researchers International Workshop In Groupware, 2008, Omaha, NE, USA. Proceedings CRIWG 2008, 2008.
- VREEDE, G.J., de, BRIGGS, R.O., 2001. "Thinklets: Five Examples of Creating Patterns of Group Interaction", In: Ackermann, F., Vreede, G.J. de (eds): Proceedings Of Group Decision & Negotiation 2001, La Rochelle France, pp. 199-208, 2001.

VREEDE, G.J., BOONSTRA, J., NIEDERMAN, F., 2002. “What Is Effective GSS Facilitation? A Qualitative Inquiry Into Participants’ Perceptions”. In Proceedings of the Hawaiian Conference on System Sciences (HICSS), IEEE Computer Society Press, 2002.

VREEDE, G.J., VOGEL, D., KOLFSCHOTEN, G., WIEN, J. “Fifteen Years of GSS in the Field: A Comparison Across Time and National Boundaries”. In Proceedings of the Hawaii International Conference on System Sciences, 2003.

VREEDE, G.J., de, DAVIDSON, R.M., BRIGGS, R.O., 2003. “How a Silver Bullet May Lose Its Shine”. Communications of the ACM, vol. 46, no. 8, pp. 96-101, 2003.

VREEDE, G.J., de, BRIGGS, R.O., 2005. “Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks”. In Hawaii International Conference on System Sciences (HICSS), Hawaii. Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

TALBOTT, S.L., 1995. “Thoughts on a Group Support System, chapter 10 - The Future Does Not Compute: Transcending the Machines in Our Midst”. Copyright 1995 O'Reilly & Associates.

Yahoo User Interface Library, 2009. Disponível para *download* em <http://developer.yahoo.com/yui/>, último acesso em 15 de agosto de 2009

WESTLEY F., WATERS, J.A., 1988. “Group Facilitation skills for managers”, Management Education and Development, vol. 19, no.2, pp. 134-143, 1998.

Apêndice A - Tabela de recomendações para escolha de ThinkLet

FreeBrainstorm	SIM	Para fazer o grupo divergir bastante em busca de novas idéias
		Para grupos de 6 ou mais participantes
		Para grupos heterogêneos, com visões bastante distintas
	NÃO	Para grupos de menos de 6 participantes
OnePage	SIM	Para gerar alguns comentários (em torno de 80 ou menos) sobre um tópico
		Para grupos de 5 ou menos pessoas
		Para grupos de 6 pessoas durante uma sessão de até 10 minutos
		Para reuniões onde provavelmente não haverá muitos comentários gerados
	NÃO	Para dar melhor feedback entre os participantes em grupos distribuídos
		Quando for esperado mais de 80 comentários
ComparativeBrainstorm	SIM	Para reuniões onde critérios de qualidade para avaliar idéias e soluções já forem conhecidos
		Para reuniões onde o tempo de realização for restrito
	NÃO	Para reuniões onde se deseja que as pessoas busquem soluções novas ou inovadoras
LeafHopper	SIM	Para reuniões onde serão abordados mais de um tópico ao mesmo tempo
		Para reuniões onde os participantes tenham diferentes níveis de interesse (grupo heterogêneo)
		Para reuniões onde não seja necessário que cada participante contribua em todos os tópicos
	NÃO	Se for necessário que os tópicos sejam abordados em uma ordem determinada
DealersChoice	SIM	Se for necessário que todos os tópicos sejam abordados por todos os participantes
		Para reuniões onde serão abordados mais de um tópico ao mesmo tempo
	NÃO	Se for necessário que certos participantes abordem determinados tópicos
Se a ordem em que os tópicos são abordados não for relevante		
Plus-Minus-Interesting	SIM	Se o background, posição ou expertise dos participantes não for relevante
		Para elaborar melhor algumas idéias antes de uma avaliação formal
TopFive	SIM	Se a elaboração não for seguida de uma avaliação formal
		Para uma transição entre dois outros Thinklets
	NÃO	Para lidar com sobrecarga de informação gerada do passo anterior
BranchBuilder	SIM	Se for necessário que cada conceito identificado seja exaustivamente discutido
		Para capturar e organizar sub-aspectos e sub-tópicos de um ou mais questões bem conhecidas
	NÃO	Para construir uma estrutura hierárquica que descreva uma questão ou tópico
TheLobbyist	SIM	Para compreender questões complexas, com pouco entendimento e mal estruturadas. Se for esperado que um tópico tenha diversos sub-tópicos e perspectivas, utilizar Could-Be-Should-Be.
		Se o grupo tiver gerado uma certa quantidade de itens complexos que necessitam de maior elaboração antes de seguir para um ThinkLet de avaliação.
		Se as pessoas do grupo tiverem interesses nos resultados da avaliação e for necessário deixar cada um manifestar explicitamente esses interesses.

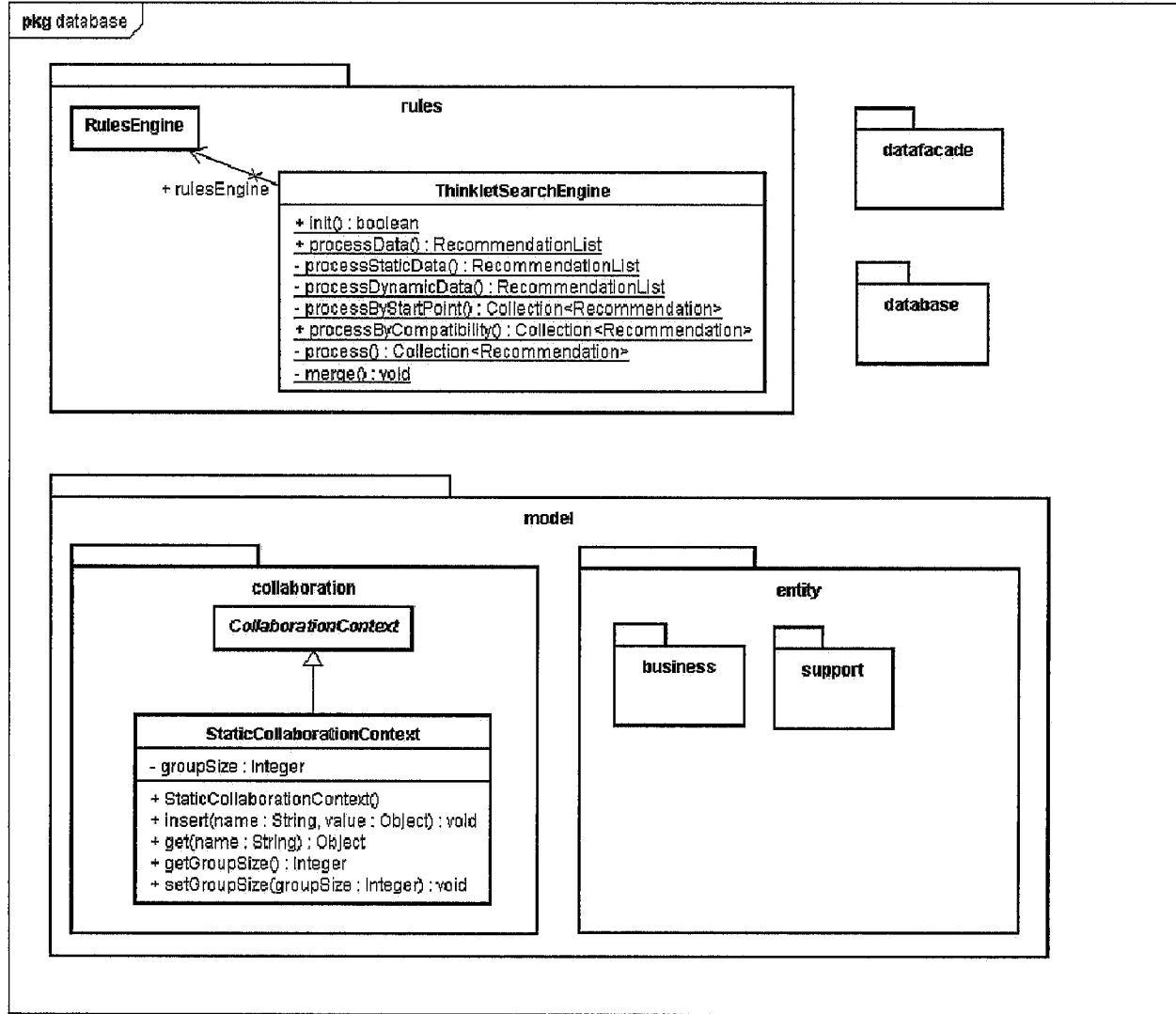
	NÃO	Se o conjunto de itens que o grupo gerou é relativamente simples de compreender, tal que não necessitem ser explicados explicitamente.
		Se os interesses individuais relacionados aos itens não são relevantes. Nesse caso, utilizar simplesmente um ThinkLet de avaliação seguido de um Crowbar.
OneMinuteMadness	SIM	Se o grupo tiver divergido entusiasticamente mas suas contribuições estão em diferentes níveis de abstração. Se o grupo estiver respondendo a uma questão de brainstorm, mas não corretamente. Por exemplo, em uma análise SWOT da organização, as pessoas frequentemente esquecem que as forças devem ter foco interno e oportunidades, externo.
	NÃO	Se o grupo precisar de tempo para explorar o espaço de brainstorm. Por exemplo, quando estiverem trabalhando em uma questão vaga e o objetivo do processo for coletar quantas perspectivas forem possíveis.
FastFocus	SIM	Para extrair rapidamente uma lista de questões chave em um nível apropriado de abstração a partir do resultado de uma atividade de divergência. Quando for importante garantir que os membros do grupo concordem sobre o significado e o fraseamento dos itens da lista resultante.
	NÃO	Para alcançar consenso nas questões chave que resultam de uma atividade de divergência. FastFocus auxilia apenas na listagem das questões chave, não na discussão de seus méritos. Considere seguir o FastFocus com um StrawPoll e Crowbar, para tal. Se os resultados da atividade de divergência apresentam respostas predominantemente bem formuladas e objetivas, tal que possam ser categorizadas facilmente. Nesse caso, utilize um PopcornSort seguindo de um BucketWalk e BucketBriefing.
OneUp	SIM	Para convergir em resultados de alta qualidade sob pressão de tempo.
		Para revelar os critérios para avaliação de qualidade das idéias enquanto se converge.
		Após uma atividade de brainstorm e antes de organização e avaliação.
	NÃO	Quando o problema é complexo e não claramente compreendido. Se a uma discussão exaustiva de cada idéia é necessária. Neste caso, utilize FastFocus ou Evolution. Se não for necessário conhecer critérios de avaliação. Utilize FastFocus, neste caso.
BucketBriefing	SIM	Para remover redundância e ambiguidade dos comentários em categorias.
		Para converter categorias dos comentários do brainstorm em categorias contendo uma lista de conceitos concisa, não redundante e sem ambiguidades.
	NÃO	Após um PopcornSort. Se uma organização simples dos comentários e idéias é suficiente para a tarefa.
DimSum	SIM	Para integrar os pensamentos de várias pessoas em uma única afirmação ou uma definição simples para um termo chave, de forma que todos os participantes estejam de acordo. Para superar um impasse causado por interesses polarizados sobre o fraseamento de uma afirmação conjunta.
	NÃO	Para desenvolver uma coleção diversa de conceitos. Quando já existir uma definição ou afirmação aceitável.
Pin the Tail on the Donkey	SIM	Quando um grupo tiver gerado vários comentários (100 a 400 ou mais) em um conjunto de idéias, propostas etc.
		Para construir um entendimento comum no grupo sobre comentários chave e questões. Para evitar repassar todos os comentários, focando apenas nos itens de destaque.

	NÃO	Para alcançar um nível de sumarização aceito por todos. Este ThinkLet não é destinado a obter um sumário do que o grupo considere importante. Utilize FastFocus para este propósito. Este ThinkLet pode ser utilizado para se preparar para um FastFocus.
	NÃO	Para determinar os elementos mais importantes de um comentário. Este ThinkLet é destinado a aprimorar o entendimento comum sobre questões que indivíduos do grupo considerem importante. Se for necessário obter uma percepção geral do grupo sobre os melhores elementos, utilize BroomWagon, possivelmente após o Pin the Tail.
BroomWagon	SIM	Se o grupo tiver gerado um grande número de itens e for necessário focar rapidamente apenas nos itens chave.
		Quando for necessário evitar que o grupo analise cada item em detalhe..
	NÃO	Quando as escolhas forem uma questão de preferência.
		Quando for necessário obter uma lista final em que cada item é avaliado cuidadosamente. Para isso, é necessário utilizar um ThinkLet de avaliação. BroomWagon apenas propicia um primeiro passo para ajudar o grupo convergir em um certo número de questões.
		Quando for necessário que o grupo tome uma decisão. BroomWagon não é adequado para tomada de decisão, apenas para separar questões chave de outras questões.
		Quando uma análise racional deve ser conduzida.
GoldMiner	SIM	Para explorar muitas contribuições de uma sessão de brainstorm e separar aquelas merecedoras de maior atenção.
		Quando é importante dar a todos os membros do grupo a oportunidade de levantar questões para maiores discussões.
	NÃO	Quando é importante considerar idéias no contexto da discussão onde elas foram geradas.
ExpertChoice	SIM	Quando o grupo se sentir desconfortável ou não for qualificado para organizar um conjunto de idéias em categorias.
		Quando o grupo não tiver tempo suficiente para organizar um conjunto de idéias.
		Se houver uma pessoa qualificada disponível para organizar as idéias.
	NÃO	Se for esperado que as escolhas do especialista possam ser altamente questionadas pelo grupo.
GarlicSqueezer	SIM	Se o grupo tiver gerado um grande número de reações (150 ou mais) em uma lista estruturada de idéias ou tópicos, e o cerne das questões precisar ser filtrado rapidamente.
		Quando um grupo não tiver tempo suficiente ou competência para filtrar as reações.
		Se houver um par com conhecimento apropriado para ajudar.
	NÃO	Se for esperado que os resultados da filtragem possam ser altamente questionados pelo grupo.
Se a atividade seguinte necessitar de uma filtragem cuidadosa e discussão de cada item.		
ReviewReflect	SIM	Quando for necessário revisar, validar e modificar o conteúdo de uma estrutura de informação existente.
	NÃO	Quando for necessário gerar uma estrutura de informação.
Concentration	SIM	Quando um grupo gerar uma ou mais lista idéias com redundâncias e ambiguidades
	NÃO	Quando for necessário limpar e remover redundância da lista de idéias
ThemeSeeker	SIM	Quando a próxima atividade não necessitar de uma lista limpa e reduzida. Por exemplo, um PopcornSort pode ser executado sem um Concentration.
		Para sumarizar os tópicos de discussão de uma atividade de brainstorming
	NÃO	Após qualquer brainstorming e antes de um PopcornSort
	NÃO	Se as categorias para organização já forem conhecidas

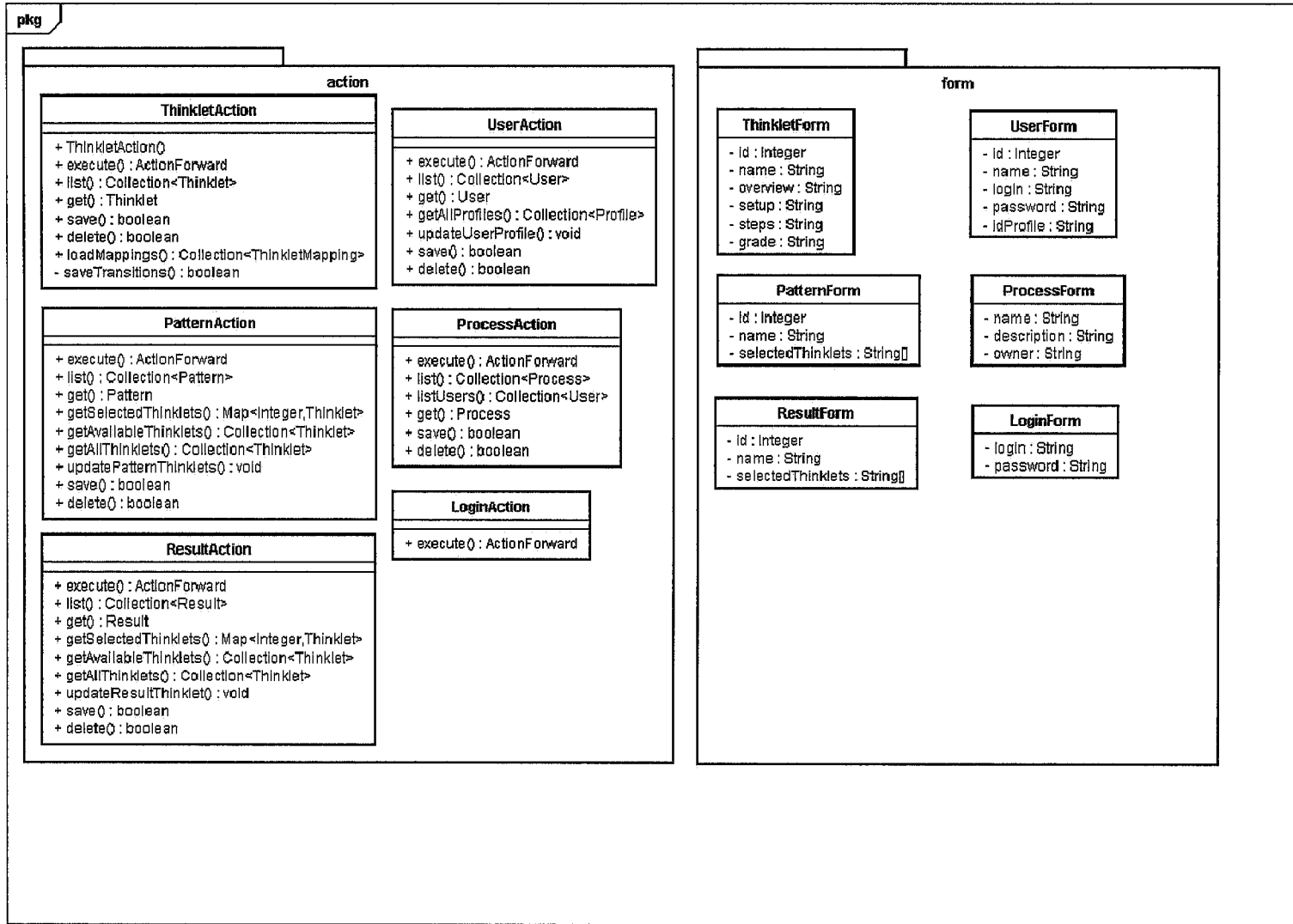
		Se ainda não foi realizado um brainstorming de ideias
RichRelations	SIM	Para criar um conjunto de categorias para organizar as ideias de um brainstorming Após qualquer brainstorming e antes de um PopcornSort
	NÃO	Para convergir de muitas ideias em poucas ideias Se as categorias para organização já forem conhecidas
PopcornSort	SIM	Após uma atividade de brainstorming e uma atividade de sumarização, como FastFocus, ThemeFinder ou RichRelations. Para organizar rapidamente um conjunto não estruturado de ideias de um brainstorming (50 a 1000 comentários) nas categorias apropriadas Para validar uma redução
	NÃO	Para convergir nas questões relevantes. Este thinklet é para categorizar e não reduzir
ChauffeurSort	SIM	Quando for necessário garantir que a categorização foi analisada cuidadosamente pelo grupo Quando criar um entendimento comum for tão importante quanto a categorização em si
	NÃO	Quando o tempo for crítico. Uma sequência de um PopcornSort e BucketWalk é muito mais eficiente Quando a categorização for simples e não requerer discussões
Evolution	SIM	Para achar possíveis categorias para organizar as ideias e elas não forem óbvias
	NÃO	Para encontrar categorias óbvias Se for necessário fazer uma organização rápida.
StrawPoll	SIM	Para medir o consenso dentro do grupo Para revelar padrões de acordo e desacordo dentro do grupo Para avaliar um conjunto de conceitos
	NÃO	Para tomar uma decisão
BucketWalk	SIM	Para validar e discutir os resultados de um PopcornSort ou LeafHopper
	NÃO	Se não for importante que cada item seja categorizado apropriadamente e uma aproximação for suficiente
BucketShuffle	SIM	Para colocar ideias uma ordenadas em uma categoria
	NÃO	Se a ordem das ideias não for importante Se a ordem precisar ser determinada de forma anônima onde cada participante poderá dar sua contribuição
MultiCriteria	SIM	Para avaliar uma lista de ideias por múltiplos critérios Quando o grupo quiser discutir questões complexas relacionadas a uma decisão Para provocar uma discussão útil e focada de um conjunto de opções/ideias
	NÃO	Como um processo final de tomada de decisão
CheckMark	SIM	Quando quiser que o grupo foque no seus pontos de concordância em vez de discordância. Para grupos altamente polarizados. Quando o grupo precisar selecionar muitos itens tal que atribuir notas a cada um for causar sobrecarga
	NÃO	Quando for importante explorar as premissas e interesses por trás das principais discordâncias do grupo. Quando quiser utilizar a atividade de votação para provocar discussão.
StakeholderPoll	SIM	Para confrontar as opiniões de diferentes grupos de interesses. Para compreender e integrar as opiniões de diferentes grupos de experts.
	NÃO	Se quiser destacar padrões de concordância em grupos altamente polarizados.
BucketVote	SIM	Quando o grupo precisar separar os itens em categorias baseadas em valor Quando velocidade for mais importante que precisão Quando os membros do grupo confiam uns nos outros para julgar as questões

	NÃO	Quando for necessário comparar, contrastar e explorar as opiniões de vários participantes.
		Quando for necessário obter uma lista rankeada de itens.
Crowbar	SIM	Para expor e examinar premissas.
		Para compartilhar informações não compartilhadas.
		Para revelar planos secretos
	NÃO	Para provocar discussão sobre questões que o grupo possuir baixo consenso.
		Se você estiver tentando fazer o grupo tratar algumas questões para maiores discussões.
MoodRing	SIM	Para rastrear padrões de consenso em uma única questão.
		Para descobrir quando é o momento para parar tomar uma decisão.
	NÃO	Se for necessário tratar diversas questões ao mesmo tempo.
PointCounterPoint	SIM	Para ajudar a quebrar um impasse em um grupo altamente polarizado.
		Para ajudar a encontrar pontos comuns entre facções polarizadas.
	NÃO	Para reforçar consenso em uma questão em particular
Red-Light-Green-Light	SIM	Para manter controle de padrões de consenso em mudança no grupo.
		Para manter a discussão focada em apenas as questões onde o grupo tem maior nível de discordância.
		Após um MultiCriteria
	NÃO	Se consenso sobre as questões não for importante.
Could-Be-Should-Be	SIM	Para evoluir de um baixo entendimento para um melhor entendimento sobre questões complexas
		Para criar uma estrutura hierárquica de todos os aspectos e sub-aspectos a serem considerados sobre uma questão complexa.
	NÃO	Para documentar conhecimento já bem compreendido pelo grupo.
SevenUp	SIM	Quando uma atividade de brainstorming gerar centenas de idéias e for desejável extrair o conjunto mais promissor.
		Quando desejar tomar nota e talvez discutir as causas das discordâncias, mas após focar nas concordâncias para seguir para os próximos passos.
		Quando desejar ter certeza de convergir em até dois passos.
	NÃO	Quando quiser que o grupo foque apenas nos seus pontos comuns.
		Quando o tempo for curto.

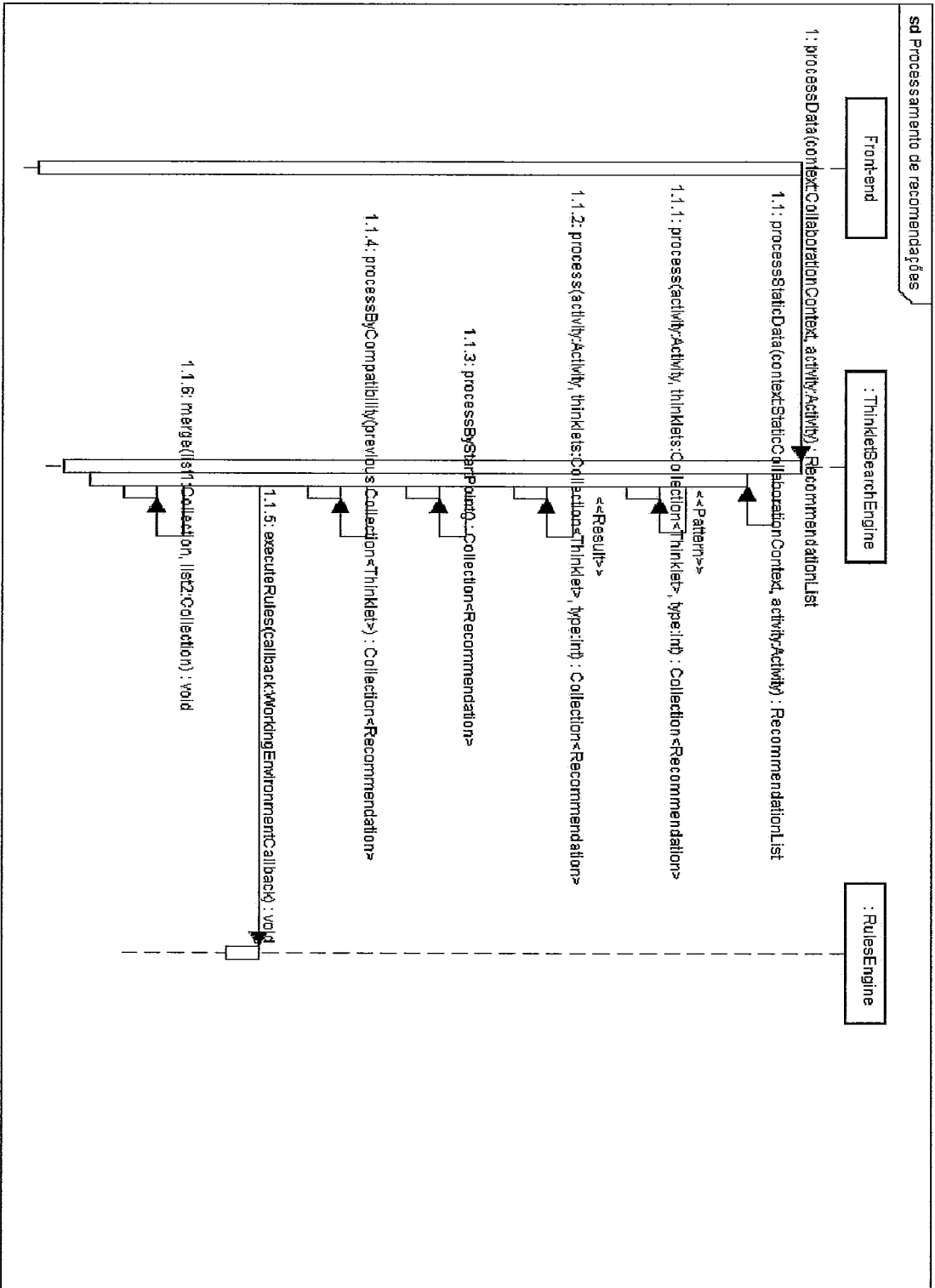
Apêndice B - Módulo core – Diagrama de Pacotes



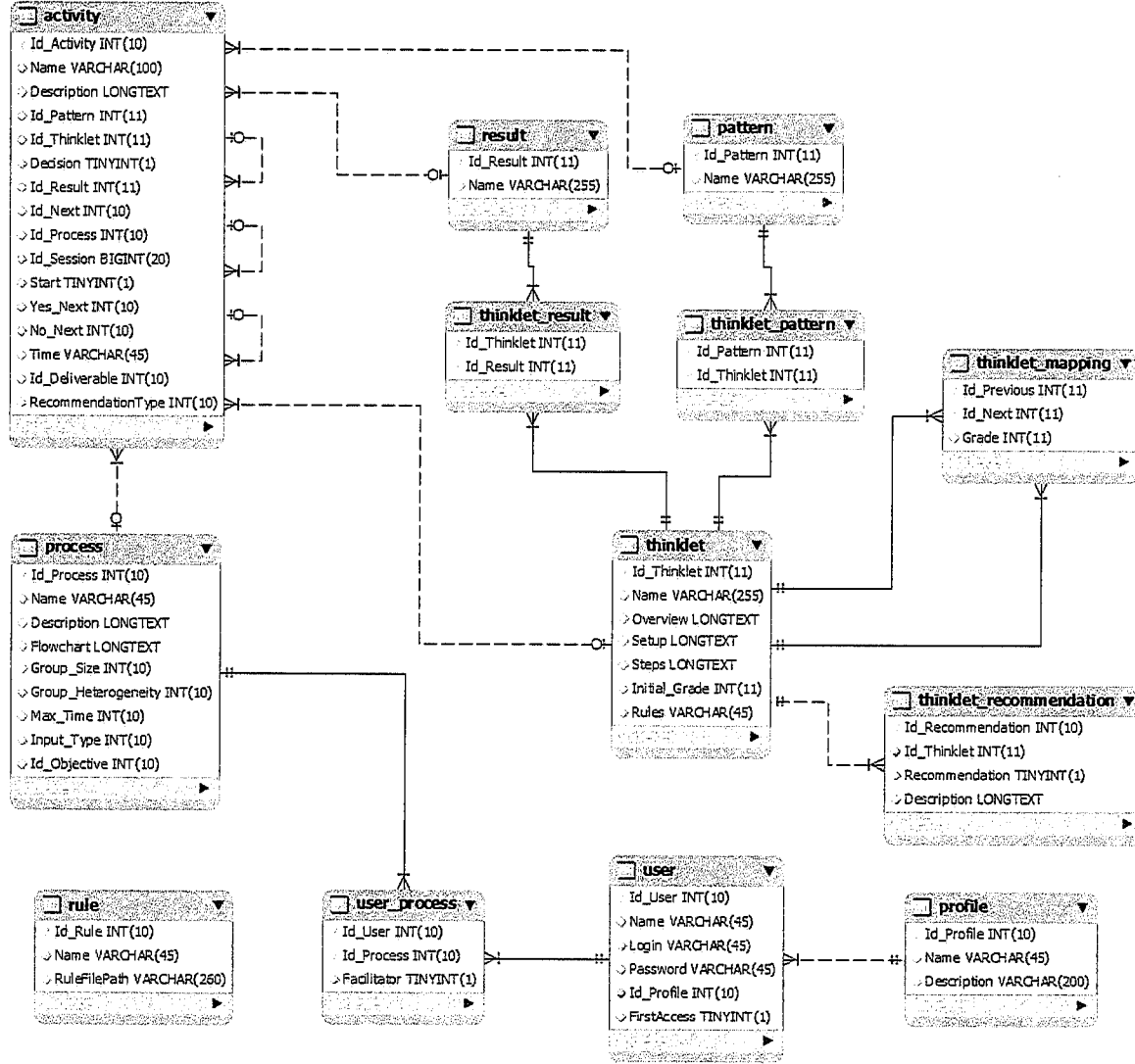
Apêndice C - Módulo de Administração – Diagrama de Pacotes



Apêndice D - Diagrama de sequência



Apêndice E - Modelo de Dados



Apêndice F - Fonte de regras (Drools) – Tamanho do grupo

```
rule "FreeBrainstorm GroupSize"
  #include attributes such as "saliency" here...
  when
    #conditions
    process : Process ( groupSize >= 6)
  then
    #actions
    DataHandler handler = DefaultDataHandler.getInstance(Constants.THINKLET_ENTITY);
    Object obj = handler.findByName("FreeBrainstorm");
    if (obj instanceof Thinklet) {
      Thinklet thinklet = (Thinklet) obj;
      Recommendation rec = new Recommendation();
      rec.setThinklet(thinklet);
      rec.setType(RecommendationType.ADVANCED_REC_TYPE);
      rec.setGrade(0);
      rec.setReason("FreeBrainstorm é recomendado para este tamanho de grupo: " +
process.getGroupSize());
      recList.add(rec);
    }

    Object obj2 = handler.findByName("OnePage");
    if (obj2 instanceof Thinklet) {
      Thinklet thinklet = (Thinklet) obj2;
      Recommendation rec = new Recommendation();
      rec.setThinklet(thinklet);
      rec.setType(RecommendationType.ADVANCED_REC_TYPE);
      rec.setGrade(-1);
      rec.setReason("OnePage não é recomendado para este tamanho de grupo: " +
process.getGroupSize());
      recList.add(rec);
    }
  end

rule "OnePage GroupSize"
  #include attributes such as "saliency" here...
  when
    #conditions
    process : Process ( groupSize < 6)
  then
    #actions
    DataHandler handler = DefaultDataHandler.getInstance(Constants.THINKLET_ENTITY);
    Object obj = handler.findByName("OnePage");
    if (obj instanceof Thinklet) {
      Thinklet thinklet = (Thinklet) obj;
      Recommendation rec = new Recommendation();
      rec.setThinklet(thinklet);
      rec.setType(RecommendationType.ADVANCED_REC_TYPE);
      rec.setGrade(0);
      rec.setReason("OnePage é recomendado para este tamanho de grupo");
      recList.add(rec);
    }

    Object obj2 = handler.findByName("FreeBrainstorm");
    if (obj2 instanceof Thinklet) {
      Thinklet thinklet = (Thinklet) obj2;
      Recommendation rec = new Recommendation();
      rec.setThinklet(thinklet);
      rec.setType(RecommendationType.ADVANCED_REC_TYPE);
      rec.setGrade(-1);
      rec.setReason("FreeBrainstorm não é recomendado para este tamanho de grupo: " +
process.getGroupSize());
      recList.add(rec);
    }
  end
```

Apêndice G - Fonte de regras (Drools) – Perfil do grupo

```
function Recommendation recommend(DataHandler handler, String thinkletName, String reason) {
    Object obj = handler.findByName(thinkletName);
    if (obj instanceof Thinklet) {
        Thinklet thinklet = (Thinklet) obj;
        Recommendation rec = new Recommendation();
        rec.setThinklet(thinklet);
        rec.setType(RecommendationType.ADVANCED_REC_TYPE);
        rec.setGrade(0);
        rec.setReason(reason);
        return rec;
    } else {
        return null;
    }
}

function Recommendation condemn(DataHandler handler, String thinkletName, String reason) {
    Object obj = handler.findByName(thinkletName);
    if (obj instanceof Thinklet) {
        Thinklet thinklet = (Thinklet) obj;
        Recommendation rec = new Recommendation();
        rec.setThinklet(thinklet);
        rec.setType(RecommendationType.ADVANCED_REC_TYPE);
        rec.setGrade(-1);
        rec.setReason(reason);
        return rec;
    } else {
        return null;
    }
}

rule "Comparative ProcessInput"
    #include attributes such as "salience" here...
    when
        #conditions
        process : Process ( groupProfile == 2 ) # heterogeneous group
    then
        #actions
        DataHandler handler = DefaultDataHandler.getInstance(Constants.THINKLET_ENTITY);

        recList.add(recommend(handler, "LeafHopper",
            "LeafHopper é recomendado quando o grupo apresentar pessoas de diferentes interesses e
backgrounds"));

        recList.add(condemn(handler, "DealersChoice",
            "DealersChoice não é recomendado quando o grupo apresentar pessoas de diferentes
interesses e backgrounds"));
    end

rule "Brainstorm Several Topics"
    #include attributes such as "salience" here...
    when
        #conditions
        process : Process ( groupProfile == 1 ) # homogeneous group
    then
        #actions
        DataHandler handler = DefaultDataHandler.getInstance(Constants.THINKLET_ENTITY);

        recList.add(recommend(handler, "DealersChoice",
            "DealersChoice é recomendado quando o grupo apresentar pessoas de diferentes interesses e
backgrounds"));

        recList.add(condemn(handler, "LeafHopper",
            "LeafHopper não é recomendado quando o grupo apresentar pessoas de diferentes
interesses e backgrounds"));
    end
```


Apêndice H - Fonte de regras (Drools) – Entrada do processo

```
function Recommendation recommend(DataHandler handler, String thinkletName, String reason) {
    Object obj = handler.findByName(thinkletName);
    if (obj instanceof Thinklet) {
        Thinklet thinklet = (Thinklet) obj;
        Recommendation rec = new Recommendation();
        rec.setThinklet(thinklet);
        rec.setType(RecommendationType.ADVANCED_REC_TYPE);
        rec.setGrade(0);
        rec.setReason(reason);
        return rec;
    } else {
        return null;
    }
}

function Recommendation condemn(DataHandler handler, String thinkletName, String reason) {
    Object obj = handler.findByName(thinkletName);
    if (obj instanceof Thinklet) {
        Thinklet thinklet = (Thinklet) obj;
        Recommendation rec = new Recommendation();
        rec.setThinklet(thinklet);
        rec.setType(RecommendationType.ADVANCED_REC_TYPE);
        rec.setGrade(-1);
        rec.setReason(reason);
        return rec;
    } else {
        return null;
    }
}

rule "Comparative ProcessInput"
    #include attributes such as "salience" here...
    when
        #conditions
        process : Process ( inputType == 3 )
    then
        #actions
        DataHandler handler = DefaultDataHandler.getInstance(Constants.THINKLET_ENTITY);

        recList.add(recommend(handler, "ComparativeBrainstorm",
            "ComparativeBrainstorm é recomendado quando um conjunto de critérios de qualidade for
conhecida"));

        recList.add(condemn(handler, "FreeBrainstorm",
            "FreeBrainstorm não pode ser usado quando o input for um conjunto de tópicos"));
        recList.add(condemn(handler, "OnePage",
            "OnePage não pode ser usado quando o input for um conjunto de tópicos"));
        recList.add(condemn(handler, "LeafHopper",
            "LeafHopper não pode ser usado quando o input for um conjunto de tópicos"));
        recList.add(condemn(handler, "DealersChoice",
            "DealersChoice não pode ser usado quando o input for um conjunto de tópicos"));

    end

rule "Brainstorm Several Topics"
    #include attributes such as "salience" here...
    when
        #conditions
        process : Process ( inputType == 2 )
    then
        #actions
        DataHandler handler = DefaultDataHandler.getInstance(Constants.THINKLET_ENTITY);

        recList.add(recommend(handler, "LeafHopper",
```

```

        "LeafHopper é recomendado quando um conjunto de tópicos for utilizado");
recList.add(recommend(handler, "DealersChoice",
        "DealersChoice é recomendado quando um conjunto de tópicos for utilizado"));

recList.add(condemn(handler, "FreeBrainstorm",
        "FreeBrainstorm não é recomendado quando um conjunto de tópicos for utilizado"));
recList.add(condemn(handler, "OnePage",
        "OnePage não é recomendado quando um conjunto de tópicos for utilizado"));
recList.add(condemn(handler, "ComparativeBrainstorm",
        "ComparativeBrainstorm não é recomendado quando um conjunto de tópicos for
utilizado"));
end

rule "Brainstorm Single Topic"
    #include attributes such as "salience" here...
    when
        #conditions
        process : Process ( inputType == 1 )
    then
        #actions
        DataHandler handler = DefaultDataHandler.getInstance(Constants.THINKLET_ENTITY);

        recList.add(recommend(handler, "FreeBrainstorm",
            "FreeBrainstorm é recomendado quando um tópico for utilizado"));
        recList.add(recommend(handler, "OnePage",
            "OnePage é recomendado quando um tópico for utilizado"));

        recList.add(condemn(handler, "LeafHopper",
            "LeafHopper não pode ser utilizado quando um tópico for utilizado"));
        recList.add(condemn(handler, "DealersChoice",
            "DealersChoice não pode ser utilizado quando um tópico for utilizado"));
        recList.add(condemn(handler, "ComparativeBrainstorm",
            "ComparativeBrainstorm não pode ser utilizado quando um tópico for utilizado"));
    end
end

```