FORUM: Modelo e Linguagem para Especificação de Regras em Ambientes Colaborativos

Luiz Eduardo Galvão Martins [1]

Luiz Camolesi Júnior [1, 2]

Resumo: O desenvolvimento de ambientes colaborativos vem crescendo significativamente nos últimos anos, acompanhado pelo aumento da complexidade e das exigências de melhor qualidade. No desenvolvimento de ambientes colaborativos é fundamental a definição de um conjunto preciso de regras que estabeleçam as interações permitidas entre os atores do ambiente. O objetivo deste artigo é apresentar o *Forum*, um modelo e uma linguagem para a especificação de regras em ambientes colaborativos, auxiliando os projetistas de software a definirem uma especificação precisa da interatividade.

Palavras-Chave: Colaboração, Regra, Linguagem de Especificação.

Abstract: Collaborative environments development has significantly increased in the last years, as well as the complexity of such systems and the search for better quality. On the collaborative environment development, the definition of precise rules set is important to allow correct interactions among the actors. The goal of this work is to present *Forum*, a model and a language to specify rules for collaborative environments, helping software designers to make rules specification to support a high level of control in such environments.

Keywords: Collaboration, Rule, Specification Language.

1 Universidade Metodista de Piracicaba, UNIMEP, FACEN.
Rodovia do Açúcar Km 156, Piracicaba, SP – Brasil
lgmartin@unimep.br
2 Universidade Estadual de Campinas, UNICAMP, CESET.
Rua Paschoal Marmo 1888, Limeira, SP – Brasil
camolesi@ceset.unicamp.br

1 Introdução

Muitos ambientes colaborativos de software (CSCL - *Computer-Supported Collaborative Learning*, jogos colaborativos por computador, CSCWD - *Computer-Supported Cooperative Work on Design,* entre outros) têm sido desenvolvidos nos últimos anos. A definição de um conjunto de regras de interação (compondo a Política de Colaboração do ambiente [1]) é importante no contexto em que muitos usuários estão trabalhando de maneira assíncrona ou síncrona [2]. As regras de interação em um ambiente asseguram a confiabilidade e a flexibilidade do ambiente, sendo, portanto, fatores críticos de qualidade que devem ser aplicados para melhoria do desempenho dos usuários [4][5][6].

As regras de interação em um ambiente, de modo geral, são expressas em uma grande variedade de formas e estilos, entre as quais a linguagem natural se destaca na preferência de muitos desenvolvedores [7]. Porém, apenas com a utilização de linguagem natural é possível incorrer em falhas na descrição das regras, tais como excessos ou ausências de elementos ou ambigüidades semânticas [8]. Assim, no desenvolvimento de software voltado ao trabalho colaborativo, é fundamental a definição de um conjunto de regras precisas e não ambíguas que estabeleçam as condições de interação entre os atores envolvidos no ambiente. Ambientes colaborativos tendem facilmente ao caos, na medida em que as regras de "conduta" não estejam bem estabelecidas, dificultando drasticamente a governabilidade do ambiente e diminuindo a produtividade e qualidade dos trabalhos colaborativos realizados [9]. Portanto, a definição de um conjunto correto de regras colaborativas é parte responsável pela adequação do ambiente no suporte às atividades colaborativas.

Com esta motivação, e baseado na análise de diversos ambientes colaborativos, este trabalho expõe os elementos de um modelo e uma linguagem voltada para a especificação de regras de interações (as quais também denominaremos de regras colaborativas). Para tanto, este artigo está estruturado da seguinte forma: na seção 2 são apresentados os elementos do modelo *M-Forum*: objeto, ator, atividade, tempo, espaço e associações, presentes em qualquer ambiente colaborativo; na seção 3 é apresentada a linguagem para a especificação formal das regras de interações (chamada *L-Forum)*; na seção 4 são apresentados dois estudos de caso demonstrando o uso efetivo da linguagem *L-Forum*; na seção 5 são apresentados trabalhos correlatos ao tema de políticas e regras de interação; e na seção 6 são apresentadas algumas conclusões, resultados e futuros direcionamentos.

2 O Modelo *M-Forum*

O modelo *M-Forum* apresentado neste trabalho se destina, entre outras aplicações, à análise e especificação dos requisitos para a modelagem de interações em ambientes colaborativos. Criado pelo grupo de pesquisas em ambientes colaborativos (GAC/UNIMEP), este modelo teve sua primeira versão finalizada em 2004 ao contemplar conceitos elementares da especificação de regras baseadas em eventos [10]. Atualmente a versão *M-*

Forum 1.5 (Figura 1) permite uma modelagem de semântica mais refinada, a qual originou a formalização da linguagem *L-Forum*, apresentada na seção 3.

2.1 Dimensões das Regras

As interações entre atores estão entre as principais características de um ambiente colaborativo, e desta forma, devem ser eficazmente modeladas para orientar os direitos, proibições e obrigações nas atividades dos atores. Na forma de regras, as *interações* devem estabelecer a ordem e restrições das atividades entre atores, e entre atores e objetos, em determinado espaço (físico ou virtual), intervalo ou limite de tempo [11].

Regras são normas de orientação para o envolvimento (relacionamento) entre os elementos do ambiente ou sistema. No estado de ativas, as regras estabelecem condições para a interação e as conseqüências destas interações para o usuário ou ambiente. Os elementos envolvidos nas regras, no modelo *M-Forum,* são denominados de *dimensões* e representam atores, atividades, objetos, tempo e espaço, além das associações entres estes. Uma regra deve envolver pelo menos um ator e um objeto com a utilização de operadores específicos.

Para exemplificar, considere uma regra simplificada, válida até 2010, de aprovação de um artigo em um evento científico: *um artigo é considerado aceito para publicação pelo comitê de programa se, dentro do prazo limite de 01 de setembro, o resultado conjunto da avaliação for bom ou ótimo*. Logicamente, esta regra deveria ser refinada para uma análise mais correta, mas neste estado primitivo já é possível revelar alguns elementos em suas respectivas dimensões: Atores: *comitê de programa (grupo);* Objetos: *artigo*, *avaliação;* Tempo: *01 de setembro* (aplicação) e *2010* (validade, prazo de vida).

Ator

Um ator é um agente em um ambiente colaborativo e tem um papel bem definido, conforme os direitos, proibições e obrigações de suas atividades. Atores são responsáveis pela execução de atividades individuais ou sociais, podendo assim, atingir objetos, um único ator ou grupo de atores (equipes ou comunidades). Os atores de um ambiente colaborativo podem ser classificados em humanos e não-humanos. Atores humanos são representações de pessoas reais envolvidas. Atores não-humanos são "seres" virtuais envolvidos em atividades interativas com atores humanos. Todo ator tem um identificador, um estado corrente e um conjunto de atributos [11]. Considerando $\underline{qh}$ a quantidade de atores humanos e $\underline{qs}$ a quantidade de atores não humanos de um ambiente, temos:

$AchS = \{Ach_1, Ach_2, ..., Ach_{qh}\}$, $AcsS = \{Acs_1, Acs_2, ..., Acs_{qs}\}$
$AchS \neq \varnothing \ \vee \ AcsS \neq \varnothing$
$AchS \cap AcsS = \varnothing$
$AcSS = AchS \cup AcsS$
$Ach_i = (Ach\_id_i , AchState_i , Ach\_AttS_i)$
$Acs_i = (Acs\_id_i , AcsState_i , Acs\_AttS_i)$

Sendo que, *AchS* representa o conjunto de atores humanos, *AcsS* representa o conjunto de atores não-humanos, Ach_i descreve uma tupla de ator humano, composta por identificador (Ach_id_i), estado corrente (*AchState)* e um conjunto de atributos (Ach_AttS_i). Acs_i descreve uma tupla de ator não-humano (mesma composição da tupla de ator humano).

Atividade

Atividade é um elemento de execução que pode ser realizado por um ator ou grupo de atores. Atividades envolvem normalmente a manipulação ou transformação de um objeto. Atividades são compostas por um identificador, um subconjunto de atividades, um subconjunto de operações e um conjunto de atributos.

Atividades devem ser expressas em interações usando *Operadores de Atividade* que permitem a definição do direito, dever, dispensa ou proibição da atividade. Operadores de Atividade são requeridos para especificar a interação de uma atividade entre atores e objetos. Operação é uma unidade de execução executada por um ator humano ou por um ator não-humano [12]. Considerando *qa* a quantidade de atividades de um ambiente, temos:

$AtSS = \{At_1, At_2, ..., At_{qa}\}$
$At_i = (At\_id_i, AtState_i\ AtS_i, OpS_i, At\_AttributeS_i)$
$AtS_i \subseteq AtSS$
$OpS_i \subseteq OpS$

Sendo que *AtSS* representa o conjunto completo de atividades exercidas no ambiente colaborativo e At_i representa uma tupla de uma atividade.

Objeto

Objetos representam elementos que constituem conceitos ou entidades do mundo real. Um objeto carrega consigo a representação de suas características estruturais e de seu comportamento. Uma modelagem de objetos estabelece uma uniformidade de visão e de tratamento, útil tanto para projetos quanto para implementação de ambientes colaborativos [11]. Atividades e Operações podem ser realizadas sobre objetos podendo alterar suas características e estado. Objetos podem ser compostos por outros objetos e caracterizam-se por um estado e um conjunto de atributos. Considerando *qo* a quantidade de objetos de um ambiente, temos:

$ObSS = \{Ob_1, Ob_2, ..., Ob_{qo}\} \vee ObSS \neq \varnothing$
$Ob_i = (Ob\_id_i, CompObS_i, ObState_i, Ob\_AttS_i)$
$CompObS_i \subseteq ObSS$

Sendo que *ObSS* representa o conjunto completo de objetos existentes no ambiente colaborativo. Ob_i representa uma tupla de um objeto. $CompObS_i$ representa o conjunto de objetos que compõe o objeto base.

Espaço

O conceito de espaço permite representar um "compartimento" ou localização de atores e objetos no ambiente colaborativo, além das áreas específicas em que atividades e operações podem ocorrer. Como nos demais elementos apresentados nesta seção, os espaços são imprescindíveis para a modelagem de um ambiente colaborativo. Dependendo de sua ontologia, a modelagem de um espaço pode ou não ser geométrica [13]. Sendo assim, pode-se ter desde uma *pasta de documentos* em ambientes *CSCL* e até *elipses* e *polígonos* em jogos eletrônicos 2-D. Para exemplificação, considerando um jogo colaborativo em que $\underline{qe}$ é a quantidade de espaços, temos:

$Coordinate = \{(x, y) | x, y \in \mathbb{R}\}$
$ELoc = (Fo1, Fo2, P)$
$Fo1, Fo2, P \in Coordinate$
$PLoc = (Sp\_id_i, SpState_i, \{Po_1, Po_2, ..., Po_n\})$
$Po_1, .. . Po_n \in Coordinate$
$Ellipse = \{el \mid el \in Eloc\}$
$Polygon = \{po \mid po \in Ploc\}$
$SpSS = Ellipse \cup Polygon$
$SpSS = \{Sp_1, Sp_2, ..., Sp_{qe}\}$

Sendo que *Eloc* é uma tupla que descreve uma elípse e *Ploc* é uma tupla que descreve um polígono. *SpSS* é o conjunto que representa todas as elipses e polígonos de um ambiente colaborativo.

Ainda sobre as mesmas aplicações de espaços geométricos, elementos da dimensão espaço devem ser expressos em interações usando um *operador de espaço* para a especificação de posição e tamanho de atores e objetos em ambientes colaborativos [14]. Alguns dos operadores mais empregados são:

$spop \in So = \{ <, <=, >, >=, =, <>, ==$*(attribution), not equal, inside, outside, intersect, meet, overlap, north, south, east, west*$\}$

Tempo

A representação do tempo tem sido foco de diversas pesquisas resultando na definição de sua taxonomia [15]. A formalização básica para o aspecto temporal pode ser baseada no conjunto de números naturais (N), para representar anos (*Ty*), meses (*Tm*), dias (*Td*), horas (*Th*), minutos (*Tmi*) e segundos (*Ts*) no *Momento* e *Intervalo*. Para *Datas,* conjuntos enumerados são usados para representar valores relativos (*Tmr, Tdr, Thr, Tmir, Tsr*) de um determinado calendário. Considerando $\underline{gt}$ a quantidade de intervalos ou momentos de tempo em um ambiente, temos:

$Ty, Tm, Td, Th, Tmi, Ts \in N$
$Tmr \in \{1, 2, 3, 4, ..., 12\}$
$Tdr \in \{1, 2, 3, 4, 5, ..., 31\}$

$Thr \in \{0, 1, 2, 3, ..., 23\}$
$Tmir \in \{0, 1, 2, 3, ..., 59\}$
$Tsr \in \{0, 1, 2, 3, ..., 59\}$
$Time = \{Ti\_id ,(Ty , Tm , Td , Th , Tmi , Ts)\}$
$Datetime = \{Ti\_id ,(Ty , Tmr , Tdr , Thr , Tmir , Tsr)\}$
$Tb, Te \in Datetime , Interval = \{Ti\_id , (Tb, Te)\}$
$TiSS = Time \cup Datetime \cup Interval$
$TiSS = \{Ti_1, Ti_2, ..., Ti_{qt}\}$

A representação semântica do tempo permite a utilização precisa de *operadores de tempo* para uso em expressões de interação em ambientes colaborativos. Na modelagem de tempo, os conceitos de *duração*, *data* e *ocorrência* têm semânticas fundamentais para estabelecimento de referências temporais. Estas semânticas são usadas para definir a lógica temporal de operadores para tempo de duração, data de ocorrência ou intervalo de ocorrência de atividades e operações definidas em interações entre atores e objetos [3][16]. Seguindo Allen [15], temos os seguintes operadores para o tratamento temporal em ambientes colaborativos:

tiop ∈ To = { <,<=,>,>=,=,<>,== (attribution), precedes, succeeds, directly precedes, directly succeeds, overlaps, is overlapped by, occurs during, contains, starts, is stated with, finishes, is finished with, coincides with }

2.2 Associações

Em situações especificas, pode-se estabelecer ou verificar associações entre atores, atores e objetos, atores e tempo, atores e espaços, e objetos e espaços. Estas associações são *estáticas* se definidas no início das interações e permanecerem inalteradas até o final de um processo colaborativo, ou são *dinâmicas se* precisarem ser alteradas durante uma interação.

O conjunto de associações pode ser relativamente grande para atender à complexidade semântica bastante diversificada de um ambiente colaborativo, contudo, entre as associações mais comuns encontram-se os tipos: *Propriedade (possuir)* - é uma associação em que um ator ou grupo de atores tem a posse de espaços, de objetos em determinados espaços, de objetos durante certo tempo, ou espaços durante certo tempo; *Utilização (usar)* - é uma associação em que um ator ou grupo de atores tem direitos de utilização de objetos em determinados espaços, objetos durante certo tempo, ou espaços durante certo tempo; *Agrupamento* é uma abstração de semântica mais flexível que a composição/decomposição, pois permite que atores, objetos, espaços, períodos ou momentos de tempo, atividades ou operações possam ser agrupados em conjuntos homogêneos ou heterogêneos independentemente de qualquer orientação física ou lógica.

2.3 Abstrações

Em processos de modelagem de sistemas, são bastante conhecidas as práticas de abstração, em que se busca uma representação mais adequada dos elementos de um ambiente e seus relacionamentos. As abstrações de maior ocorrência em ambientes colaborativos são a classificação, generalização e composição [4].

A *abstração de classificação* está presente em qualquer ambiente colaborativo, pois permite a especificação de tipos (ou seja, atores, objetos, atividades, tempos e espaços) e instâncias na formação de expressões de regras que genericamente envolvem todos os elementos de um tipo ou um elemento específico. Por exemplo, em um jogo de futebol uma regra determina que todos os jogadores (tipo) devem usar o uniforme, enquanto que outra regra determina que um determinado jogador (instância) com cartão vermelho deve sair do jogo. A classificação é uma construção semântica que permite a elaboração de uma *hierarquia de classificação* em que tipos podem ser instanciados ao mesmo tempo em que são instâncias de supertipos.

Através da *Abstração de Generalização* é possível representar aspectos estruturais, funcionais e comportamentais, genéricos e específicos de atores, objetos, tempos, atividades e espaços. Em expressões de regras, esta abstração permite estabelecer normas de interação gerais e particularizadas, como por exemplo: nenhum jogador de futebol em campo pode tocar a bola com a mão, mas o jogador goleiro pode tocar ou segurar a bola se estive dentro de sua área de jogo. A generalização é uma construção semântica que permite a elaboração de uma hierarquia de generalização, em que tipos podem ter subtipos que ainda podem ter outros subtipos e assim por diante, em um processo de especificação até um grau de refinamento necessário para a legibilidade e consistência das regras de colaboração.

A *Abstração de Composição* é usualmente empregada para indicar que dois ou mais objetos, espaços, períodos ou momentos de tempo estão associados física ou logicamente como partes de um elemento complexo. Esta abstração é representada através de referências entre os elementos envolvidos. A composição é uma construção semântica que permite a elaboração de uma *hierarquia de composição* em que os elementos da ontologia podem ser decompostos em um processo de refinamento necessário para a legibilidade e consistência das regras de colaboração.

2.4 Políticas de Colaboração

Políticas de Colaboração definem normas para as possíveis interações em um ambiente, podendo ser estabelecidas por agrupamento de regras com metas ou objetivos bem definidos. A definição destas políticas em sistemas pode ser definida de duas formas:

- Intra-regras: em que a política é definida internamente às regras de um ambiente, ou seja, na definição de uma regra podem existir referências a outras regras que possuam algum grau de interdependência e assim, as associações entre regras que caracterizam uma política estão estabelecidas internamente ao corpo das regras;

- Extra-regras: em que a política é definida externamente às regras, e desta forma todas as inter-relações entre regras que estabelecem uma composição para normatização de um ambiente são definidas em um elemento independente, no caso, podendo ser a própria Política.

Ambas as formas de definição de uma Política apresentam suas vantagens, mas na atual versão 1.5, o modelo *M-Forum* define políticas na forma de intra-regras, como sendo o conjunto de regras ativas $P = \{r_1, r_2, ..., r_{qr}\}$. As possíveis interdependências entre as regras envolvem as priorizações de aplicação (regime) de cada regra ou as indicações de aplicação obrigatória ou opcional (operadores normativos), apresentados na linguagem *L-Forum*.

A Figura 1 apresenta um modelo conceitual dos principais elementos colaborativos segundo o *M-Forum*. No modelo, uma política é composta por um conjunto de regras, sendo que cada regra é composta por um número de elementos (ator, atividade, objeto, tempo e espaço) e os elementos podem se associar de diversas formas de abstração.

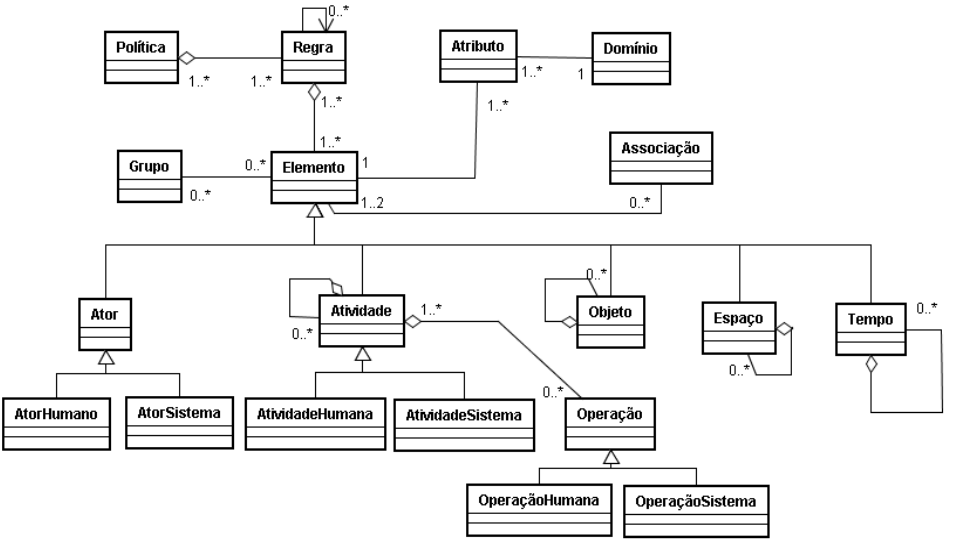

Figura 1. Classes conceituais do *M-Forum*

3 A Linguagem *L-Forum*

A Linguagem *L-Forum* está baseada nos conceitos do modelo *M-Forum*. É uma linguagem formal, orientada ao domínio da especificação de ambientes colaborativos, para definição de regras de interação que auxiliam na descrição da interatividade em ambientes colaborativos, podendo trazer diversos benefícios, entre os quais:

- Orientar na definição de regras de um ambiente;
- Aumentar o grau de precisão das regras, diminuindo a ambigüidade na interpretação pelos projetistas de um ambiente colaborativo;
- Facilitar a depuração de inconsistências ou conflitos. O *M-Forum* não apresenta soluções para estes problemas, mas permite propor técnicas viáveis com as apresentadas por Bressan [17], que propõe a inspeção intra-regras e inter-regras;
- Permitir a flexibilização do ambiente com a retirada, modificação ou inclusão de regras, permitindo que a Política de um ambiente colaborativo possa ser adaptada.

A linguagem possui três conjuntos de cláusulas com propósitos bem específicos:

- *Contexto*: composto pelos *parâmetros* para a execução ou ativação de uma regra e pelas condições de aplicabilidade que estabelecem os cenários (valores de atributos, aspectos temporais ou espaciais) em que uma regra pode ser aplicada;
- *Definição (ou corpo)*: conjunto de expressões no qual são estabelecidas as ações ou condições para as ações de interação entre os elementos, podendo envolver opcionalmente aspectos de transitoriedade no tempo e no espaço. Regras e ações podem ser invocadas no corpo de uma regra permitindo a sequencialização de expressões complexas ou simples;
- *Regime*: é um conjunto opcional composto pela colocação de regras inter-relacionadas que tenham a mesma orientação para serem executadas ou aplicadas. Também podem ser definidos os cenários (valores de atributos, aspectos temporais ou espaciais) em que uma regra deve ser ativada ou desativada para uso.

Para exemplificar a estrutura sintática (apêndice) da linguagem *L-Forum,* a tabela 1 apresenta a regra Julgar Artigo com tempo de vida até o ano de 2010 (*survivability*). No exemplo apresentado, a palavra *now* indica a data corrente e *A.possui.V.resultado* indica o atributo *resultado* da avaliação *V,* vinculada ao artigo *A* pela associação *possui*. *Set* é um operador para atribuição de estado, no caso, aceito ou rejeitado.

Tabela 1. Regra especificada em L-Forum

```
Rule  Julgar Artigo [active]   {
Parameters::   (Au: Autor, A:Artigo, V:Avaliação, C:Comitê_de_programa)
Applicability:: (now <= 01/09)
Body::            If ( A.possui.V.resultado = bom) or ( A.possui.V.resultado = ótimo)
                    then {        Action (C set  aceito A);
                                  Rule (Avisar Aceitação (C, Au, A, V), obligation);     }
                     else  {      Action (C set  rejeitado A);
                                  Rule (Avisar Rejeição (C, Au, A, V), obligation);      }
Survivability:: (now <= 2010)
}
```

4 Estudos de Caso

Com o intuito de demonstrar a pertinência e aplicabilidade, tanto do modelo *M-Forum* quanto da linguagem *L-Forum,* na especificação de regras de interação em ambientes colaborativos, nesta seção são apresentados dois estudos de casos. O primeiro estudo de caso apresenta algumas regras de um jogo de tênis, e o segundo estudo apresenta algumas as regras de um ambiente de ensino à distância.

4.1 Jogo de Tênis

Considere a utilização plena do modelo de definição de regras aplicado a um jogo de tênis, tendo a seguir a descrição de algumas normas para disputa de "simples" (entre dois oponentes), a partir das quais modelamos algumas das regras do ambiente [18]:

- *Considerar um saque inválido:* o saque é uma ação que põe a bola em jogo e para ser considerado válido o jogador tem que bater com sua raquete na bola e fazê-la ultrapassar a rede, sem tocá-la, e tocar dentro da área de recepção (polígono, Figura 2) na quadra do oponente;
- *Considerar uma devolução válida de bola:* o jogador pode rebater a bola com sua raquete após esta não ter tocado ou tocado uma única vez no seu lado da quadra, de tal forma que a bola retorne para o lado da quadra do jogador adversário;
- *Marcar um ponto ao longo da troca de bolas:* recebe nova pontuação o jogador que fizer com que o adversário não consiga devolver a bola de volta à quadra de seu oponente;
- *Para ganhar um set:* ganha o *set* aquele que ganhar 6 *games*, porém, no caso de um empate de *games* em 5 a 5, o *set* deverá terminar quando um jogador obtiver 7 *games*, se houver empate em 6 *games*.

Considerando as normas do jogo descritas anteriormente, é possível identificar os elementos e as associações (tabela 2) e especificar algumas de suas regras (tabela 3).

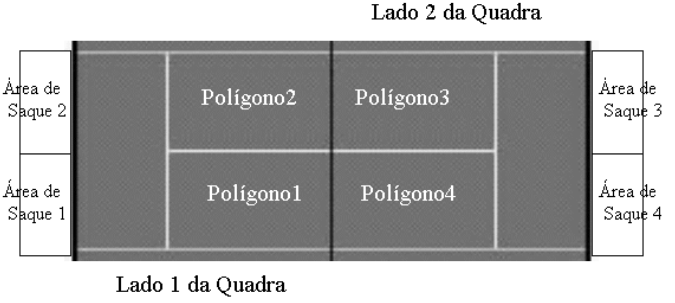

Figura 2. Espaços da quadra de tênis

Tabela 2. Alguns elementos e associações do jogo de tênis

Atores (atributo)	Jogador [1,2] (saques), Juiz, Sistema
Objetos (atributo)	Bola (toques, quique), Rede, Raquete [1,2], Saque, Devolução, Placar [1,2] (pontos, games, sets, tiebreak), Jogo (numerosets)
Atividades	Iniciar, Encerrar, Bater, Rebater, Atribuir, Invalidar
Tempo	Intervalo entre games, Intervalo entre sets
Espaço	Lado [1,2], Polígono [1,2,3,4], Área de Saque [1,2,3,4]
Abstrações	Jogador.**possui**.Raquete *(Associação - Propriedade)* Jogador.**saca em**.Área de Saque *(Assoiação. - Utilização)* Jogador.**está com**.Bola *(Associação - Propriedade)* Jogador.**pontua em**.Placar Lado.**define**.Área de Saque *(Assoiação. - Agrupamento)* Lado.**composto de**.Polígono *(Composição)* Área de Saque.**orienta**.Polígono

Na tabela 3, são apresentadas algumas regras colaborativas previstas para o correto andamento do jogo de tênis. Para cada regra, logo após seu nome identificador, é definido o seu contexto, utilizando-se as cláusulas *Parameters* e *Applicability*.

Em *Parameters* pode-se definir qualquer elemento estrutural de um ambiente colaborativo, os elementos estruturais reconhecidos pela linguagem são: *actor, group, object, activity, operation, space, time e association* (apêndice).

Em *Applicability*, define-se em que condição a regra pode ser aplicada, portanto nesta cláusula, uma expressão condicional sempre deverá ser avaliada. Por exemplo, na regra "*Pontuar Set",* a condição de aplicabilidade da regra é que o jogo tenha sido iniciado. Portanto, a cláusula *Applicability* expressa a pré-condição da regra. Na cláusula *Body* define-se o corpo da regra, especificando-se as condições de funcionamento interno da regra, as ações embutidas na regra e possíveis invocações de outras regras (encadeamento de regras).

Tabela 3. Especificação de regras do jogo de tênis usando *L-Forum*

```
Comentário: Sistema pode iniciar um jogo.
Rule  Iniciar Jogo [active]
{
Parameters::       (i:Jogador, j:Jogador, s:Sistema, jo:Jogo, ar:Área de Saque)
Applicability::    (i inside i.sacaem.ar) and
                   (j inside j.sacaem.ar ) and (jo is não_iniciado)
Body::             Action (s right set  iniciado jo);
}
```

```
Comentário: Jogador pode rebater um saque do oponente.
Rule Rebater Saque [active]
{
Parameters::      (i,j:Jogador, b:Bola, s:Sistema, jo:Jogo, ar:Área de Saque, po:Polígono)
Applicability::   (jo is iniciado) and (b.quiques = 1 ) and
          (b.prim_quique inside j.esta em.ar.orienta.po) and (b.toque_rede = false) and (i is sacou)
Body::            Action (j right bater b);
}
```

```
Comentário: Sistema define o vencedor de um jogo
Rule Proclamar Vencedor [active]
{
Parameters::      (i:Jogador, jo:Jogo, s:Sistema, jo:Jogo, pl:Placar)
Applicability::   (jo is iniciado) and ( (i.pontuaem.pl.sets  =  2 ) and ( jo.numerosets  = 3 ) ) or
                  ( (i.pontuaem.pl.sets = 3) and (jo.numerosets  = 5 ) )
Body::            Action (s obligation encerrar jo);
                  Action (s obligation set i vencedor);
}
```

```
Comentário: Sistema incrementa pontuação de jogador.
Rule Pontuar Set [active]
{
Parameters::      (i:Jogador, j:Jogador, jo:Jogo, s:Sistema, pl:Placar)
Applicability::   (jo is iniciado)
Body::        If  ( (i:.pontua em.pl.games = 6) and (j.pontua em.pl.games < 5)) or
                    (i.pontua em.pl.games = 7)
              Then { Action (s obligation attribute next (1,2,3) i.pontua em.pl.sets);   }
}
```

4.2 TelEduc

O TelEduc é um ambiente de criação, participação e administração de cursos à distância na *Web* que está sendo desenvolvido e aperfeiçoado a mais de dez anos pela Universidade Estadual de Campinas (UNICAMP), envolvendo uma parceria entre o Núcleo de Informática aplicada à Educação (NIED) e o Instituto de Computação (IC) [19][20]. A partir da análise das normas do TelEduc, foi possível identificar os elementos e as associações (tabela 4) e especificar algumas de suas regras (tabela 5).

Tabela 4. Alguns elementos e associações do TelEduc

Atores	Formador, Aluno, Colaborador Externo, Monitor, Participante (generalização dos demais atores humanos), Sistema.
Grupo	Grupo de alunos
Objetos	Email, Disciplina, Material de Apoio, Avaliação, Atividade, Agenda, Dinâmica do Curso, Fórum de Discussão, Perfil, Portfólio de Grupo, Portfólio Individual, Relatório de Acesso, Relatório de Freqüência, Intermap, Idioma, Mural, Leitura, Parada Obrigatória, Sessão de Bate-papo, Pergunta Freqüente, Anotação, Exercício, Mensagem, Arquivo, Endereço URL.
Atividades	Enviar, Retirar, Publicar, Anexar, Atribuir, Ler, Comentar.
Tempo	Data da avaliação, Prazo de atividade.
Espaço	Pasta, Sala de bate-papo.
Abstrações	Aluno.**compõe**.Grupo *(Associação - Composição)* Formador.**publicou**.Atividade Atividade.**possui**.Avaliação *(Associação - Propriedade)* Participante.**publicou**.Mensagem Mural.**dispõe de**.Mensagem

Tabela 5. Especificação de regras do Teleduc usando *L-Forum*

```
Comentário: Participante pode colocar mensagem em Mural.
   Rule  Publicar em Mural [active]
   {
   Parameters::      (p:Participante, s:Sistema, mu:Mural, me:Mensagem)
   Body::            Action  (p obligation redige me);
                     Action  (p obligation envia me);
                     Action  (s obligation create p.publicou.me );
                     Action  (s obligation create mu.dispõe de.me );
}
Comentário: Formador pode criar uma atividade de apoio à metodologia de ensino.
   Rule  Criar Atividade [active]
   {
   Parameters::      (f:formador, s:Sistema, at:Atividade, ar:Arquivo, e:EndereçoURL, av:Avaliação)
   Body::            Action  (f obligation redige at);
                     Action  (f right create at.possui.av);
                     Action  (f obligation envia at);
                     Action  (s obligation create f.publicou.at );
                     Rule  (Anexar Arquivo (f, at, ar) right );
                     Rule  (Anexar EndereçoURL (f, at, e) right );
}
Comentário: Aluno pode criar um grupo de alunos.
   Rule  Criar Grupo de Alunos [active]
   {
   Parameters::      (a:Aluno, am:Aluno, g:Grupodealunos)
   Body::            Action  (a right create g);
                     Action  (s obligation create a.compõe.g );
                     Rule  (Incluir Membro (a, g, am) right );
}
Comentário: Aluno pode incluir item de conteúdo no portfólio.
   Rule  Incluir Item em Portfólio de Grupo [active]
   {
   Parameters::      (a:Aluno, g:Grupodealunos, i: item, ar:Arquivo,
                      e:EndereçoURL, av:Avaliação, p:Portfólio de grupo)
   Applicability::   (exist a.compõe.g) and (g is aberto)
   Body::            Action  (a right redige i);
                     Rule  (Anexar Arquivo (a, p, ar) right );
                     Rule  (Anexar EndereçoURL (a, p, e) right );
                     Rule  (Vincular Avaliação (a, i, av) right );
}
```

5 Trabalhos Correlatos

Regras são úteis para descrever normas e restrições em ambientes colaborativos e, consequentemente, podem ser úteis no desenvolvimento de modelos que orientam tais ambientes. Na literatura encontram-se definições de várias aplicações de regras, por exemplo: regras de produção, regras de transição, regras de interação, regras de restrição etc.

Nas últimas décadas, muitas pesquisas têm enfocado aspectos comportamentais em ambientes colaborativos, especialmente para a definição de políticas que estabeleçam regras de conduta para a formalização de interações humanas [7]. Neste contexto, algumas linguagens para especificação de políticas têm sido propostas, entre as quais podemos citar: Rei [11], KaoS [21] e Ponder [14]. Embora estas linguagens tratem da definição de políticas em ambientes colaborativos, elas não são totalmente apropriadas para apoiar vários aspectos importantes do trabalho colaborativo, dos quais podemos citar:

- Definição explícita da semântica de espaço e tempo para o refinamento de regras de interação;
- Definição de interdependência entre regras, para a definição de direitos, obrigações e proibições, baseada em *Deontic Logic* [22];
- Distinção entre atores humanos e não-humanos para permitir diferentes tipos de controles sobre regras, ações, atividades e operações, baseada em *Role Theory* [3][23];
- Decomposição de atividades em sub-atividades e unidades operacionais suportando o fluxo de trabalho entre atores, conforme abordado pela Teoria da Atividade [8];
- Hierarquização de objetos para construir uma abstração de composição suportando fluxo de trabalho entre atores e colaboração orientada a objetos;
- Relacionamento semântico entre regras, ação, atividade e operação, suportando diferentes níveis de interação entre atores e objetos;
- Abstração de agrupamento para estabelecer associações e classificação de qualquer elemento para controle e modelagem de otimização;
- Representação de interdependências entre as regras envolvendo as priorizações de aplicação de cada regra ou as indicações de aplicação obrigatória ou opcional;
- Distinção entre as condições que estabelecem a aplicação de regras *(applicability),* condições de exceção *(if-then-else)* e as condições de validade *(survivability).*

Prezando pelas demandas reais dos recursos citados acima para a especificação de regras com semântica mais precisa, o modelo *M-Forum,* reformulado na sua versão 1.5, atende aos requisitos citados para os ambientes colaborativos e permite a flexibilização para sua extensão.

6 Conclusões

Neste artigo foram apresentados alguns aspectos do modelo *M-Forum* e da linguagem *L-Forum* voltados para a formalização de regras, contribuindo para se obter uma especificação da interatividade em ambientes colaborativos. Os construtores da linguagem *L-*

Forum, embasados no modelo *M-Forum,* foram especificamente projetados para definição de regras colaborativas com potencial para facilitar a padronização das regras.

A linguagem proposta possui os elementos necessários e são extensíveis para a formalização de regras colaborativas, com ênfase para o tratamento dos elementos que normalmente constituem um ambiente colaborativo: atores, objetos, atividades e operações, espaço, tempo, grupos e associações. Com o exemplo do jogo de tênis baseado nas regras profissionais (ITF – *International Tennis Federation*, www.itftennis.com) e do ambiente de aprendizagem TelEduc, pôde-se mostrar a pertinência e os benefícios da utilização do modelo e linguagem *L-Forum* na descrição de regras colaborativas.

Os construtores da linguagem permitem a organização da especificação das regras, de tal forma que estas se tornem precisas, não ambíguas e de fácil entendimento. No projeto da linguagem procurou-se privilegiar a legibilidade das regras formalizadas, de tal forma que a curva de aprendizado da linguagem e o posterior esforço de interpretação das regras sejam baixos. A linguagem apresentada está em evolução, embora o seu núcleo esteja bem estabelecido, pois é grande a complexidade e diversidade das expressões de regras nos diversos ambientes colaborativos. O *M-Forum* tem sido utilizado com sucesso para a formalização de regras em diversos ambientes colaborativos, desde ambientes de jogos [18], ferramentas colaborativas de análise de sistemas [10], ambientes de ensino à distância [24], além da modelagem da interatividade com robôs [25].

REFERÊNCIAS

[1] Adomavicius, G., Tuzhilin, A.: User Profiling in Personalization Applications through Rule Discovery and Validation. ACM Int. Conf. on Knowledge Discovery and Data Mining. p. 377-381, 1999.

[2] Egenhofer, M. J., Mark, D.: Modeling Conceptual Neighborhoods of Topological Line-Region Relationships, Int. Journal Geographical Inf. Systems. 9:5, p. 555-565, 1995.

[3] Edwards, W. K.: Policies and Roles in Collaborative Applications. International Conference on Computer Supported Cooperative Work – CSCW. p. 11-20, 1996

[4] Jixin, M., Knight, B.: A General Temporal Theory. The Computer Journal, Oxford University Press, 31:2. p. 114-123, 1999.

[5] Takada, H., Shimakawa, H., Horiike, S.: The Time/Place /Object Model for Tracking and History Management in Manufacturing Line Control. IEEE Int. Symposium on Database Applications in Non-Traditional Environments. p. 385-394, 1999.

[6] Yokota, Y., Sugiyama, K., Tarumi, H., Kambayashi, Y.: A Workspace Management Mechanism for Cooperative Work. IEEE Int. Symposium on Database Applications in Non-traditional Environments. p. 333-340, 1999.

[7] Chang, C. K., Vorontsov, A., Zhang, J., Quek, F.:Rule-Mitigated Collaboration Technology. IEEE Workshop on Future Trends of Distributed Computing Systems. p. 137-142, 1999.

[8] Kuuti K.: Activity Theory as a Potential Framework for Human-computer Interaction Research. In Context and Consciousness: Activity Theory and Human-Computer Interaction, MIT Press. Cambridge. p. 17-44, 1995.
[9] Raposo, A. B., Cruz, A. J., et all.: Coordination Components for Collaborative Virtual Environment. Computer & Graphics Journal, Elsevier. v. 25. p. 1025-1039, 2001.
[10] Camolesi Jr., L., Martins, L. E. G.: A Model for Interaction Rules to Define Governance Policies in Collaborative Environments, Lecture Notes in Computer Science - LNCS, Springer-Verlag, v. 3865, p. 11-20, 2006.
[11] Kagal, L., Finin, T., Johshi, A.: A Policy Language for Pervasive Computing Environment. Workshop on Policy for Distributed Syst. and Networks. p. 63-76, 2003.
[12] Jones, P. M., Lucenti Jr., M. J.: Flexible Collaborative Support: An Architecture and Application. IEEE Int. Conf. On Systems, Man and Cybernetics. p. 1057-1062, 2000.
[13] Mylon, P.: On Space, Its Time, and Spatiotemporal Expressions. In: Qvortrup, L. (ed.): Virtual Space: Spatiality in Virtual Inhabited 3D. Springer-Verlag, p. 47-72, 2002.
[14] Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The Ponder Policy Specification Language. IEEE International Workshop on Policy for Distributed Systems and Networks - Policy. LNCS, Springer-Verlag, v. 1995.. p. 18-38, 2001.
[15] Allen, J. F.: Maintaining Knowledge about Temporal Intervals. Communications of ACM. 26:1. p. 832-843, 1983.
[16] Mok, A. K., Lee, C., Woo, H.: The Monitoring of Timing Constraints on Time Intervals. IEEE Real-Time Systems Symposium. p. 1-10, 2002.
[17] Bressan, R. L.: Verificação de Conflitos em Regras de Interação de Ambientes Colaborativos. Dissertação. Ciência da Computação. UNIMEP. 2008.
[18] Camolesi Jr., L., Martins, L. E. G.: Um Modelo de Interações para Definição de Regras de Jogos", II Simp. Bras. de Jogos p/ Computador e Entretenimento Digital, v. 1, 2005.
[19] Otsuka, J. L. et all.: Suporte à Avaliação Formativa no Ambiente de Educação à Distância TelEduc, VI Congr Iberoamericano de Informática Educativa (IE2002), 2002.
[20] Otsuka, J. L., Rocha, H. V.: Um modelo de suporte à avaliação formativa para ambientes de EaD. Relatório Técnico IC-05-11. IC-Unicamp, 2005.
[21] Uszok, A., Bradshaw, J., et all.: KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. IEEE Int. Workshop on Policy for Distributed Systems and Networks – Policy. p. 93-98, 2003.
[22] Meyer, J. J., Wieringa, R. J.: Deontic Logic: A Concise Overview. J. Wiley and Sons Publishing, Chichester, 1994.
[23] Robbins S.: Organizational Behavior: Concepts, Controversies and Applications, Prentice Hall, 1991.
[24] Camolesi Jr., L.; Forbeck, V. L. A.: Uso de Métrica Baseada em Regras para a Análise da Interatividade dos Ambientes de Educação à Distância nas Atividades de Ensino e Aprendizagem. Simpósio Brasileiro de Informática na Educação -SBIE, 2007.
[25] Martins Jr., J.; Camolesi Jr., L.; Caurin, G. A. P.: Scara3D: 3-Dimensional HRI Integrated to a Distributed Control Architecture for Remote and Cooperative Actuation. ACM Symposium on Applied Computing – SAC, v. 2, 2008.

Apêndice - Sintaxe Completa da Linguagem *L-Forum*

<rule> ::=	´Rule´ <rule name> ´[´<status>´]´ ´{´ [<context>] ´Body ::´<definition> [<regime>] ´}´
<context>::=	[´Parameters: (´<parameters>´)´] [<applicability>]
<definition> ::=	<condition> | <action> | <rule call> [<definition>]
<regime> ::=	<survivability> [´Priorities:´ <priority>]
<parameters> ::=	<identifier>´:´<element> [´,´ <parameters>]
<element> ::=	<actor> | <group> | <object> | <space> | <time> | <association> | <activity> | <operation>
<type> ::=	´actor´ | ´group´ | ´object´ | ´space´ | ´time´ | ´association´ | ´activity´ | ´operation´
<applicability> ::=	´Applicability::´ <condition expression>
<survivability> ::=	´Survivability::´ <condition expression>
<condition> ::=	´If ´ <condition expression> ´then {´ <definition> ´}´ [´else {´ <definition>´}´]
<action> ::=	´Action: (´ <supreme action> | <definition action> | <attribution action> | (<actor> <normative operator> {<activity> (<actor> | <object>)} [<space attribution operation> <space>] [<time attribution operation> <time>]) [´);´ <action>] ´);´
<supreme action> ::=	<actor> <normative operator> <primitive operator> (<element>|<domain>|(´is part of´|´is a´) <element>) | <actor> <normative operator> <primitive group operator> <group> <element>)
<definition action> ::=	<actor> ´set´ <status>
<attribution action> ::=	<actor> ´attribute´ (<value>|<formula>| ((next | prior) (<value domain>|<domain name>)) <attribute>
<condition expression> ::=	´(´ (<attribute><attribute condition operator>(<value>|([´all´|´any´] (<value domain>|<domain name>)) | ((<element>|<rule name>) <status condition operator> <value>) | (<element>) <element group operator> <group>) | (<element>) <group group operator> <group>) | ([not] ´exist´<association>) | (<element> <space condition operator> (<space> | (<group>|<domain name>))) | (<attribute><space condition operator>(<value>| ´here´|([´all´|´any´] (<value domain>|<domain name>)))) | (<attribute><time condition operator>(<value>| ´now´|([´all´|´any´] (<value domain>|<domain name>)))) | (<actor> <activity condition operator> <activity>) | (<condition expression> (and | or)) ´)´
<rule call> ::=	´Rule (´ <rule name> ´ (´<parameters>´)´ <normative operator> [´);´ <rule call>] ´);´
<priority> ::=	<name> [´,´ <priority>]
<group> ::=	[(<identifier>) |´any´ | ´all´| <integer value>] [<association>] [´:´<name>] [´:Group´]
<actor> ::=	[(<identifier>) |´any´ | ´all´| <integer value>] [<association>] [<group>´.´] [´:´<name>] [´:Actor´]
<activity> ::=	[<activity>´.´] <name> [´:Activity´]
<operation> ::=	<activity>´.´<name> [´:Operation´]
<object> ::=	[(<identifier>) |´any´ | ´all´| <integer value>] [<association>] [<object>´.´] [´:´<name>] [´:Object´]

<space> ::=	[(<identifier>) |´any´	´all´| <integer value>] [<association>] [<space>´.´] [´:´<name>] [´:Space´]
<time> ::=	[(<identifier>) |´any´	´all´| <integer value>] [<association>] [´:´<name>] [´:Time´]
<association> ::=	<element>´.´<name> [´.´<association>] [´:Association´]
<attribute> ::=	<element>´.´<name> [´:Attribute´]
<domain> ::=	<name> (<value domain> | <grouping>)
<value domain> ::=	´(´ (<numeric value> {´,´ <numeric value>}) | (<string> {´,´ <string>}) ´)´
<grouping> ::=	(<type> <name> <attribute condition operator> (<value>|([´all´| ´any´](<value domain>|<domain name>))) {(and| or) <grouping>}) | (<element> {´,´ <element>})
<status> ::=	<element> <status attribution operator> <value>
<primitive group operator> ::=	´insert´ | ´delete´ | ´update´
<primitive operator> ::=	´create´ | ´destroy´
<group element operator> ::=	´∈´ | ´∉´ |
<group group operator> ::=	´⊆´ | ´⊄´ | ´⊂´
<activity condition operator> ::=	[´not´] ´has´
<normative operator> ::=	´right´ | ´prohibition´ | ´obligation´ | ´dispensation´
<activity attribution operator> ::=	´receive´
<status attribution operator> ::=	´put on´ | ´move to´
<space attribution operator> ::=	´==´ | ´inside´ | ´outside´ | ´north´ | ´south´ | ´east´ | ´west´
<time attribution operator> ::=	´in´ | ´on´ | ´at´
<attribute condition operator> ::=	´<´ | ´<=´ | ´>´ | ´>=´ | ´=´ | ´<>´
<time condition operator> ::=	´<´ | ´<=´ | ´>´ | ´>=´ | ´=´ | ´<>´ | ´precedes´ | ´succeeds´ | ´directly precedes´ | ´directly succeeds´| ´overlaps´| ´is overlapped by´| ´occurs during´| ´contains´| ´starts´ | ´is stated with´| ´finishes´| ´is finished with´| ´coincides with´
<space condition operator> ::=	´<´ | ´<=´ | ´>´ | ´>=´ | ´=´ | ´<>´ | ·´not equal´ | ´inside´ | ´outside´|´intersect´|´meet´|´overlap´|´north´ |´south´ | ´east´ | ´west´
<arithmetic operator> ::=	´+´ | ´-´ | ´/´ | ´*´
<status condition operator> ::=	[´not´] ´is´
<rule name> ::=	<string>
<name> ::=	<string>
<domain name>::=	<string>
<string> ::=	<letter>,{<letter>}
<integer value> ::=	<digit>,{<digit>}
<real value> ::=	<digit>,{<digit>},´.´,<digit>,{<digit>}
<numeric value> ::=	<integer value> | <real value>
<digit> ::=	´1´ | ´2´ | ´3´ | ´4´ | ´5´ |´6´ | ´7´ | ´8´ | ´9´ | ´0´
<letter> ::=	´a´ | ´b´ | ´c´ | ´d´ | ´e´ | ´f´ | ´g´ | ´h´ | ´I´ | ´j´ | ´k´ | ´l´ | ´m´ | ´n´ | ´o´ | ´p´ | ´q´ | ´r´ | ´s´ | ´t´ | ´u´ | ´v´ | ´x´ | ´w´ | ´y´ | ´z´
<identifier> ::=	<letter>|<identifier> [,<letter> | <integer value>]
<formula>::=	´(´(<attribute>|<numeric value>|<formula>) [<arithmetic operator> <formula> ´)´
<value>::=	<string> | <numeric value>