

Especificação de Regras de Interação em Ambientes Colaborativos

Autores

Luiz Eduardo Galvao Martins

Luiz Camolesi Junior

1. Introdução

Muitos ambientes colaborativos de software (CSCL - *Computer-Supported Cooperative Learning*, jogos colaborativos por computador, CSCWD - *Computer-Supported Cooperative Design*, entre outros) têm sido desenvolvidos e o conjunto de regras de interação (compondo a Política de Colaboração) é importante no contexto de ambientes onde muitos usuários estão trabalhando de maneira assíncrona ou síncrona (EDWARDS, 1996). 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 (KAGAL et al., 2003).

As regras de interação em um ambiente, de modo geral, são expressas em uma grande variedade de formas e estilos, considerando que a linguagem natural permite esta variabilidade na representação das regras. Desta forma, é possível incorrer em falhas na descrição das regras, como excessos ou faltas de elementos e ambigüidade semântica. 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 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. Portanto, a definição de um conjunto de regras colaborativas é em parte responsável pela adequação do ambiente no suporte às atividades colaborativas (JONES e LUCENTI Jr., 2000).

2. Objetivos

O desenvolvimento de software para ambientes de trabalho colaborativo vem crescendo significativamente nos últimos anos, acompanhado pelo aumento da complexidade de tais sistemas e da exigência de melhor qualidade para os mesmos (USZKO et. Al., 2003). Quando se desenvolve software para ambientes colaborativos é fundamental a definição de um conjunto preciso de regras que governem as interações permitidas entre os atores do ambiente. O objetivo deste artigo é apresentar um modelo e uma linguagem formal (chamada *Forum*) para a especificação de regras em ambientes colaborativos, auxiliando os projetistas de software a definirem uma boa especificação de regras que garantam um alto nível de governabilidade de tais ambientes.

3. Desenvolvimento

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 - FACEN/UNIMEP) este modelo teve sua primeira versão finalizada em 2004 ao contemplar conceitos elementares da especificação de regras baseadas em eventos (CAMOLESI Jr. e MARTINS, 2006). 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*.

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 e intervalo ou limite de tempo.

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*, representando atores, atividades, objetos, tempo, espaço, além das associações entres estes. Uma regra deve envolver pelo menos um ator e um objeto com a utilização de operadores específicos.

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 identificador, um estado corrente e um conjunto de atributos. Considerando *qh* a quantidade de atores humanos e *qs* a quantidade de atores não humanos de um ambiente, temos:

$$AchS = \{Ach_1, Ach_2, \dots, Ach_{qh}\}, AcsS = \{Acs_1, Acs_2, \dots, Acs_{qs}\}$$

AchS

$$\neq \emptyset \vee AcsS \neq \emptyset, AchS \cap AcsS = \emptyset$$

$$AcSS = AchS$$

$$Ach_i = (Ach_id_i, AchState_i, Ach_AttS_i)$$

$$Acs_i = (Acs_id_i, AcsState_i, Acs_AttS_i)$$

\cup AcsS

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. 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 enquanto ou por um ator não-humano. Considerando *qa* a quantidade de atividades de um

ambiente, temos:

$$AtSS = \{At_1, At_2, \dots, At_{qa}\}$$

$$At_i = (At_id_i, AtState_i, AtS_i, OpS_i, At_AttributeS_i)$$

AtS_i

$$\subseteq AtSS, OpS_i \subseteq OpS$$

Objetos representam elementos que constituem conceitos ou entidades do mundo real. Um objeto carrega consigo a representação das características estruturais e do comportamento do que está representando. 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. Objetos podem ser compostos por outros objetos e caracterizam-se por um estado e um conjunto de atributos. Considerando qa quantidade de objetos de um ambiente, temos:

$$ObSS = \{Ob_1, Ob_2, \dots, Ob_{qo}\}$$

Espaço virtual permitem o "armazenamento" ou localização atores e objetos, além das áreas específicas envolvidas em atividades e operações. Como os demais elementos apresentados nesta seção, os espaços virtuais são imprescindíveis para a modelagem de um ambiente colaborativo. Dependendo de sua ontologia, a modelagem de um espaço pode ou não ser geométrica (MYLON, 2002). Sendo assim, pode-se Ter desde uma *pasta de documentos* em ambientes CSCL até *elipses* e *polígonos* em jogos eletrônicos 2-D. Considerando um jogo colaborativo em que qe a quantidade de espaços, temos:

Coordinate

E_{Loc}

Fo1, Fo2, P

P_{Loc_i}

Po₁, . . . Po_n

Ellipse = {e_l | e_l

SpSS = Ellipse È Polygon

SpSS = {Sp₁, Sp₂, ..., Sp_{qt}}

Ainda sobre as mesmas aplicações de espaços geométricos, elementos da dimensão espaço devem ser expressos em interações usando o *Operador de Espaço* para a especificação de posição e tamanho de atores e objetos em ambientes colaborativos [6].

sop

A representação do tempo tem sido foco de diversas pesquisas resultando na definição de sua taxonomia (MOK et al., 2002). A formalização básica para o aspecto temporal pode ser baseada no conjunto de número naturais (N

Ty, Tm, Td, Th, Th, Ts

Tmr

Thr

Time = {Ti_{id}, (Ty, Tm, Td, Th, Tmi, Ts)}

Datetime = {Ti_{id}, (Ty, Tmr, Tdr, Thr, Tmir, Tsr)}

Tb, Te

TiSS = Time È Datetime È Interval

TiSS = {Ti₁, Ti₂, ..., Ti_{qt}}

A representação semântica do tempo permite a utilização precisa de *Operadores de tempo* para uso em expressões de interação em um ambientes. Na modelagem de tempo, *Durações, Datas e Ocorrências* 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.

tiop

) para representar anos (Ty), meses (Tm), dias (Td), horas (Th), minutos (Th) e segundos (Ts) no *Tempo e Intervalo*. Para *Datas*, conjuntos enumerados são usados para representar valores relativos ($Tmr, Tdr, Thr, Tmir, Tsr$) de um determinado calendário. Considerando *qt* a quantidade de intervalos ou momentos de tempo de um ambiente, temos: $\in \mathbf{N} \in \{1, 2, 3, 4, \dots, 11, 12\}$, $Tdr \in \{1, 2, 3, 4, 5, \dots, 31\} \in \{0, 1, 2, 3, \dots, 24\}$, $Tmir, Tsr \in \{0, 1, 2, 3, \dots, 59\} \in \text{Datetime}$, $\text{Interval} = \{Ti_id, (Tb, Te)\} \in \text{To} = \{<, <=, >, >=, =, <>, == \text{ (attribution)}\}$, precedes, succeeds, directly precedes, directly succeeds, overlaps, is overlapped by, occurs during, contains, starts, is stated with, finishes, is finished with, coincides with } = \{(x, y) | x, y \in \mathbf{N}\} = (Fo1, Fo2, P) \in \text{Coordinate} = (Sp_id_i, SpState_i, \{Po_1, Po_2, \dots, Po_n\}) \in \text{Coordinate} \in \text{Eloc} }, $\text{Polygon} = \{po | po \in \text{Ploc}\} \in \text{So} = \{<, <=, >, >=, =, <>, == \text{ (attribution)}\}$, not equal, inside, outside, intersect, meet, overlap, north, south, east, west } $\vee \text{ObSS} \neq \emptyset$

$\text{Ob}_i = (\text{Ob_id}_i, \text{CompObS}_i, \text{ObState}_i, \text{Ob_AttS}_i)$

CompObS_i

$\subseteq \text{ObSS}$

4. Resultados

A Linguagem *L-Forum* (Figura 2) está diretamente baseada nos conceitos apresentados no Modelo *M-Forum*. É uma linguagem formal para definição de regras que auxilia a governabilidade de um ambiente colaborativo, podendo trazer diversos benefícios, entre os quais podemos citar:

- Orientar na definição de regras do ambiente;
- Aumentar o grau de precisão das regras, diminuindo a ambigüidade na interpretação pelos atores do ambiente;
- Facilitar a depuração de inconsistências;
- Permitir a flexibilização do ambiente com a retirada, modificação ou inclusão de regras, permitindo que a Política de um ambiente possa ser definida ou 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;
- Regime: é um item opcional composto pelo conjunto de regras inter-relacionadas que tenham 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.

5. Considerações Finais

Neste artigo são apresentados alguns aspectos do modelo e da linguagem *Forum*, voltados para a formalização de regras, contribuindo para se obter uma boa governabilidade de ambientes colaborativos. Os construtores da linguagem *L-Forum*, embasados no modelo *M-Forum*, foram especificamente projetados para definição de regras colaborativas (ou regras de interação), com potencial para facilitar a formalização das regras, tornando-as padronizadas. A linguagem proposta possui os elementos necessários 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.

Os construtores da linguagem permitem uma boa 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, como já foi comprovado em experimentação com alunos em salas de aula. A linguagem apresentada está em permanente evolução, embora o seu núcleo esteja bem estabelecido, pois são grandes a complexidade e diversidade das expressões de regras nos diversos ambientes colaborativos. *Forum* tem sido utilizado com sucesso para a formalização de regras em diversos ambientes colaborativos, desde ambientes de jogos, como também um ambiente de edição colaborativa de diagramas de classes (Engenharia Colaborativa de Software) e diversos softwares de ensino à distância (EAD).

Referências Bibliográficas

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, Berlin Heidelberg, Vol. 3865, Computer Support Cooperative Work in Design 2, W. Shen and Kuo-Ming Chao et al.(Eds.). (2006) 11-20.

EDWARDS, W. K.: Policies and Roles in Collaborative Applications. International Conference on Computer Supported Cooperative Work – CSCW. (1996) 11-20.

JONES, P. M., LUCENTI Jr., M. J.: Flexible Collaborative Support: An Architecture and Application. IEEE Int. Conf. On Systems, Man and Cybernetics. (2000) 1057-1062.

KAGAL, L., FININ, T., JOHSHI, A.: A Policy Language for Pervasive Computing Environment. IEEE Int. Workshop on Policy for Distributed Systems and Networks. (2003) 63-76.

MOK, A. K., LEE, C., WOO, H.: The Monitoring of Timing Constraints on Time Intervals. IEEE Real-Time Systems Symposium. (2002) 1-10.

MYLON, P.: On Space, Its Time, and Spatiotemporal Expressions. In: Qvortrup, L. (ed.): Virtual Space: Spatiality in Virtual Inhabited 3D. Springer-Verlag, (2002) 47-72.

USZOK, A., BRADSHAW, J., JEFFERS, R., SURI, R., et al.: 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. (2003) 93-98.

Anexos

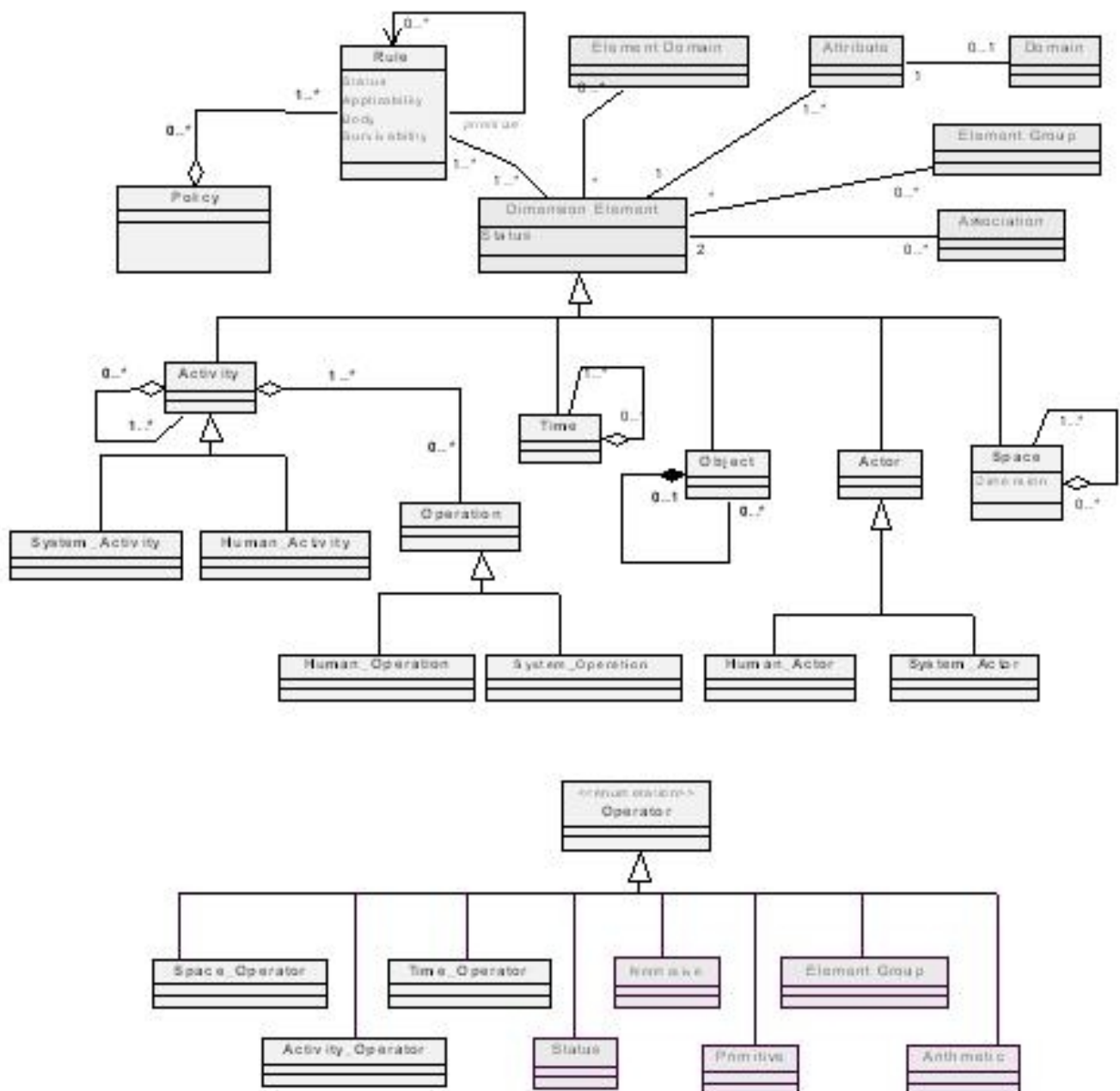


Figura 1. Meta-Modelo *M-Forum*

```

<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> ::=
'Execute 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>)
<definition action> ::=
<actor> 'set' <status>
<attribution action> ::=
<actor> 'attribute' (<value> | <formula> | ((next | prior) (<value domain> | <domain name>)) <attribute>
'(' (<attribute> <attribute condition operator> (<value> | (( 'all' | 'any' ) (<value domain> | <domain name>))) |
(( <element> | <rule name> ) <status condition operator> <value> ) |
(<element> ) <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> ::=
[( <identifier> ; ) | 'any' | 'all' | <integer value> ] [ <association> ] <name> [ ':Activity' ]
<operation> ::=
[( <identifier> ; ) | 'any' | 'all' | <integer value> ] [ <association> ] <activity> ' ' <name> [ ':Operation' ]
<object> ::=
[( <identifier> ; ) | 'any' | 'all' | <integer value> ] [ <association> ] <name> [ ':Object' ]
<space> ::=
[( <identifier> ; ) | 'any' | 'all' | <integer value> ] [ <association> ] <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 operator> ::=
'create' | 'destroy'
<group operator> ::=
'e' | 'g' | 's' | 'a' | 'c'
<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> ::=
'<' | '<=' | '>' | '>=' | '=' | '<>'
'<' | '<=' | '>' | '>=' | '=' | '<>' | '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> ::=
[ 'not' ] 'is'
<status condition operator> ::=
<string>
<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>

```

Figura 2 – Linguagem *L-Forum*