```
grammar org.xtext.example.lucam.Lucam with org.eclipse.xtext.common.Terminals

generate lucam "http://www.xtext.org/example/lucam:/Lucam:"

Lucam:
        useCaseHeader+=UseCaseHeader
        useCaseFlows+=UseCaseFlows
        useCaseFooter+=UseCaseFooter
;

UseCaseHeader:
        useCaseName+=UseCaseName
        useCaseBriefDescription+=UseCaseBriefDescription
        systemName+=SystemName
        primarySecondaryActors+=PrimarySecondaryActors
;

UseCaseFlows:
        mainFlowName+=MainFlowName
        mainFlowScope+=MainFlowScope
        (alternateFlows+=AlternateFlows)?
;

UseCaseFooter :
                'Key Scenarios' (keyScenario+=KeyScenario)*
                preConditions +=PreConditions
                postConditions += PostConditions
                specialRequirements +=SpecialRequirements
                extensionPoints+=ExtensionPoints
;

UseCaseName :
         'Use Case: ' ControlClassID+=ID point+=POINT
;

UseCaseBriefDescription :
        'Brief Description' descricao+=STRING  point+=POINT
;

SystemName:
        'System: ' SystemID+=ID point+=POINT
;

PrimarySecondaryActors:
                'Primary and Secondary Actors'
                primaryActorName+=PrimaryActorName
                (secondaryActorName+=SecondaryActorName)?
;
PrimaryActorName :
        'Primary Actors: ' ActorID+=ID (',' ActorID+=ID)* point+=POINT
;
```

```
SecondaryActorName:
        'Secondary Actors: ' ActorID+=ID (',' ActorID+=ID)* point+=POINT
;

MainFlowName:
        'Main Flow: ' MethodControlClassID+=ID  point+=POINT
;

MainFlowScope:
        (ActorID+=ID ' starts Use Case' point+=POINT)?
        mainFlow+=MainFlow
        (ActorID+=ID ' finishes Use Case' point+=POINT)?
;

MainFlow:
        (mainFlowElements+=MainFlowElements)+
;

MainFlowElements :
                (mainFlowCore+=MainFlowCore)
                | flowIf+=FlowIf
                | flowLoop+=FlowLoop
                | flowConcurrency+=FlowConcurrency
;

FlowIf:
        'If ' condition+=Condition mainFlow+=MainFlow ('Else' mainFlow+=MainFlow )? 'EndIf'
;

FlowLoop:
        'Loop' condition +=Condition
                mainFlow+=MainFlow
        'EndLoop'
;


FlowConcurrency:
        'StartConcurrency' mainFlow +=MainFlow 'concurrent' mainFlow +=MainFlow
=>'StartConcurrency'
;

Condition :
        '['conditionText+=ConditionText ']'
;


MainFlowCore:
        (ActorSystemID+=ID(

mainFlowTabTransVerb+=MainFlowTabTransVerb
                                                |tabIntransVerb+=TABINTRANSVERB)
        point+=POINT)
```

```
          | returnMessage+=ReturnMessage point+=POINT
          |interactionUseCase+=InteractionUseCase point+=POINT
;

InteractionUseCase:
   actor+=ID ' executes ' ('use' 'case '|'alternate flow') useCaseName+=STRING ((','| ' and' |' or')
useCaseName+=STRING)*
;



MainFlowTabTransVerb:
          (tabTransVerb+=TABTRANSVERB) ((MethodID+=STRING)? (
mainFlowAttibutes+=MainFlowAttibutes | ' on '
          mainFlowBoundaryClass+=MainFlowBoundaryClass) |
mainFlowSubstantivoSemComplemento+=MainFlowSubstantivoSemComplemento)
;

MainFlowSubstantivoSemComplemento:
          ('the ')? TABSUBSTANTIVESEMCOMPLEMENTO
;

MainFlowAttibutes:
          ((' the ')? substantive+=TABSUBSTANTIVE ('('(AttributeTypeID+=ID)? AttributeID+=ID (',
'(AttributeTypeID+=ID )? AttributeID+=ID)*')')? )?

           ('on ' mainFlowBoundaryClass+=MainFlowBoundaryClass
          | (' of ') mainFlowEntityClass+=MainFlowEntityClass
          |  mainFlowsActorClass += MainFlowsActorClass
          | mainFlowProssessingClass+=MainFlowProssessingClass)
;


MainFlowsActorClass:
          (' for ' | ' to ') ('the ')? actor+=ID
;

MainFlowBoundaryClass:
          BoundaryClassID+=ID
;

MainFlowEntityClass     :
          EntityClassID+=ID (' on ' mainFlowBoundaryClass+=MainFlowBoundaryClass
                    | mainFlowProssessingClass+= MainFlowProssessingClass
                    | mainFlowsActorClass += MainFlowsActorClass
          )?
;

MainFlowProssessingClass:
          ' by ' mainFlowCommunication+=MainFlowCommunication
;
```

```
MainFlowCommunication:
        CommunicationID+=STRING
;

ReturnMessage:
    (SystemID=ID verb+=TABTRANSVERB simpleReturnMessage+=SimpleReturnMessage
('to ' | 'for ')('the')?
    ActorSystemID+=ID )
;

AlternateFlows:
    'Alternate Flows'
    (alternateFlowScope+=AlternateFlowScope)+
;

AlternateFlowScope:
    'Alternate Flow ' NUM+=INT ': ' MethodControlClassID+=ID point+=POINT
    alternateFlowCore+=AlternateFlowCore

;
AlternateFlowCore:
    mainFlow+=MainFlow
;

KeyScenario:
    'Key Scenario ' NUM+=INT ': ' MethodControlClassID+=ID point+=POINT
;

PreConditions:
    'Preconditions' (Text+=STRING)? point+=POINT
;

PostConditions:
    'Postconditions' (Text+=STRING)? point+=POINT
;

SpecialRequirements:
    'Special Requirements' (Text+=STRING)? point+=POINT
;
ExtensionPoints:
        'Extension Points' ( texto+=STRING)?  point+=POINT
;

SimpleReturnMessage:
    texto += STRING
;

ConditionText:
    texto+=STRING
;

POINT:
	'.'
;

TABTRANSVERB:
	'makes' | 'verifies' | ' deletes' | 'generates' | 'enables' | 'validates' | 'authenticates' |
	'accepts' | 'adds' | 'affords' | 'alerts' | 'allows' | 'analyses' | 'answers' | 'appears' |
	'approves' | 'arranges' | 'asks' | 'attaches' | 'backs' | 'bans' | 'begins' | 'calculates' |
	'calls' | 'changes' | 'checks' | 'chooses' | 'cleans' | 'clears' | 'closes' | 'collects' |
	'communicates' | 'compare' | 'completes' | 'connects' | 'consists' | 'contains' | 'copies'
|
	'corrects' | 'counts' | 'decides' | 'delays' | 'describes' | 'detects' | 'enters' |
	'examines' | 'fails' | 'fetches' | 'files' | 'fills' | 'finds' | 'gathers' | 'gets' |
	'gives' | 'guarantees' | 'guards' | 'identifies' | 'includes' | 'informs' | 'interrupts' |
	'introduces' | 'joins' | 'keeps' | 'lists' | 'loads' | 'loses' | 'marks' | 'matches' |
	'meets' | 'needs' | 'obtains' | 'opens' | 'pauses' | 'perexecuta mits' | 'picks' | 'posts' |
	'presents' | 'puts' | 'questions' | 'reads' | 'receives' | 'recognises' | 'records' |
	'releases' | 'removes' | 'repairs' | 'repeats' | 'replaces' | 'replies' | 'reports' |
	'reproduces' | ' requests' | 'rescues' | ' returns' | 'runs' | ' saves' | ' searches' |
	' selects' | ' sends' | 'settles' | ' shows' | 'shuts' | 'signals' | 'starts' | 'stores' |
	'subtracts' | 'supplies' | 'suspends' | 'switches' | 'takes' | ' types' | 'understands' |
	'waits' | 'warns' | 'writes' | ' retrieves' | ' displays' | 'modifies' | 'updates'| ' deducts'| '
credits' |
	'performs' | ' cancels '
;

TABINTRANSVERB:
	'fails' | 'exits' | 'closes' | 'fall' | 'pause'
;

TABSUBSTANTIVE:
	'message' | 'attributes' | 'dependencies' | 'informations' | 'request' | ' response' | 'list'
|
	'schedule' | 'prerequisites'| 'response' | 'file' | 'files' | 'notification' | 'justification' |
'record' | 'privileges' |
	'permission' | 'user interface' | 'interface'

;

TABSUBSTANTIVESEMCOMPLEMENTO:
	'operation'
;
```