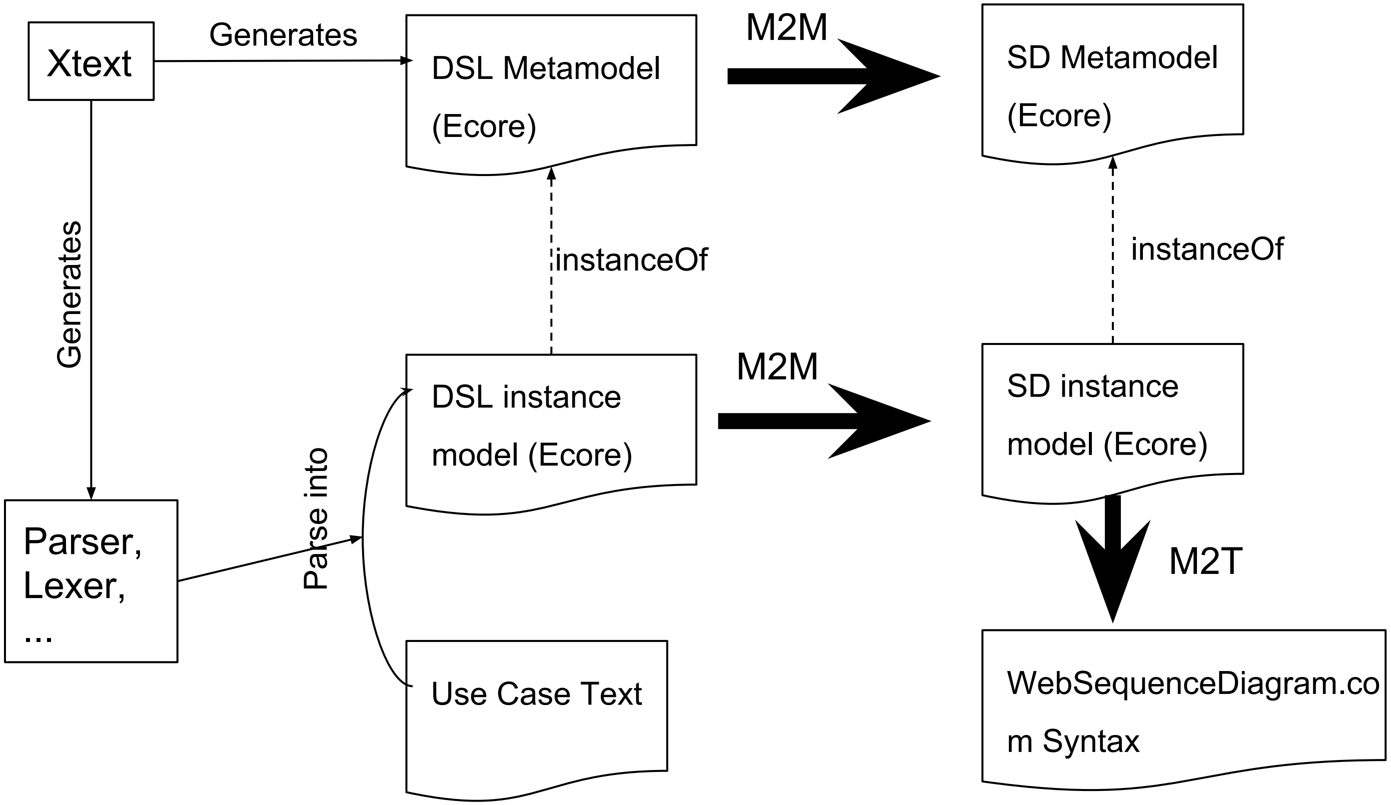
Textual Use Case DSL with Sequence Diagram Transformation

Andrés Carrasco

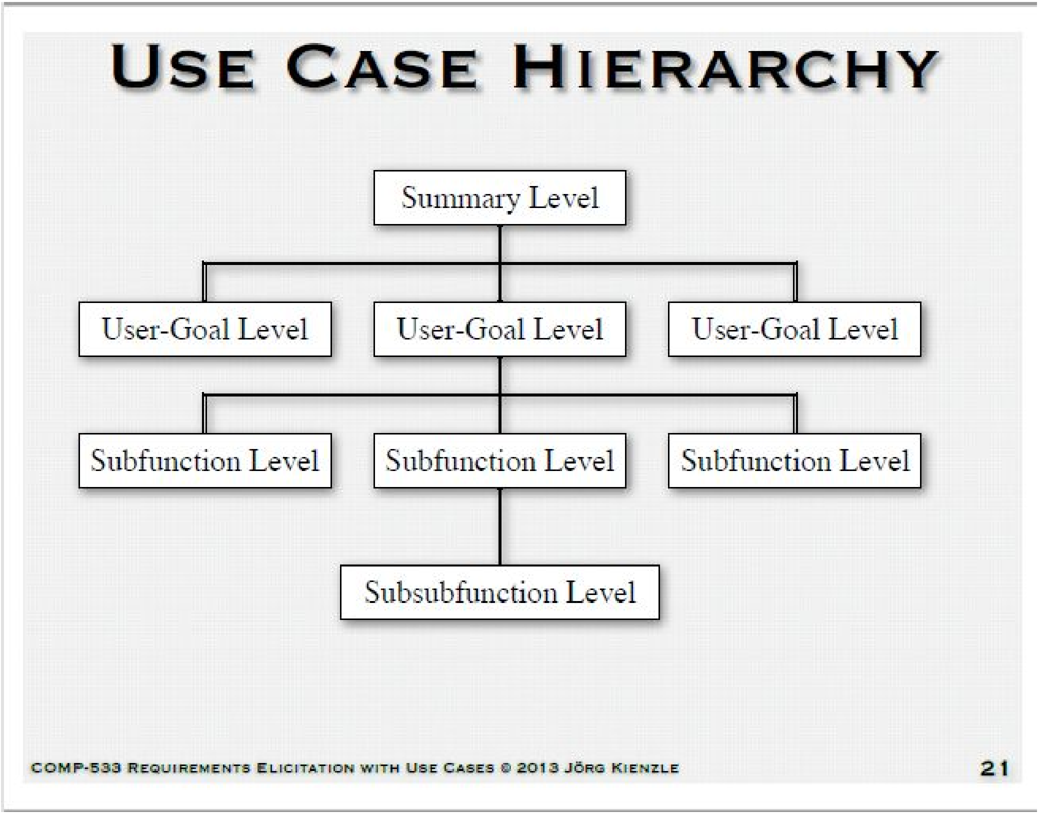
Workflow



Xtext
Generates
DSL Metamodel
(Ecore)
M2M
SD Metamodel
(Ecore)
Generates
instanceOf
instanceOf
DSL instance
model (Ecore)
M2M
SD instance
model (Ecore)
Parser,
Lexer,
...
Parse into
M2T
Use Case Text
WebSequenceDiagram.co
m Syntax

Use Case

- Use Case Name
- Scope
- Level
- Intention in Context
- Multiplicity
- Actors
- Main Success Scenario
- Extensions & Exceptions



Use Case Hierarchy

Summary Level

User-Goal Level

User-Goal Level

User-Goal Level

Subfunction Level

Subfunction Level

Subfunction Level

Subsubfunction Level

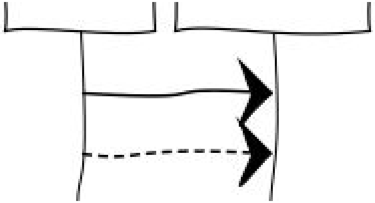
COMP-533 Requirements Elicitation with Use Cases © 2013 Jörg Kienzle

21

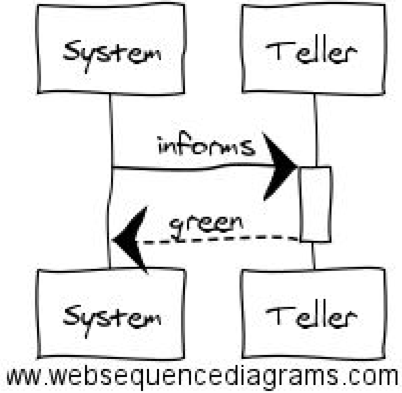
Sequence Diagram

The available constructs from WebSequenceDiagrams.com are

- Only Synchronous Messages and Replies

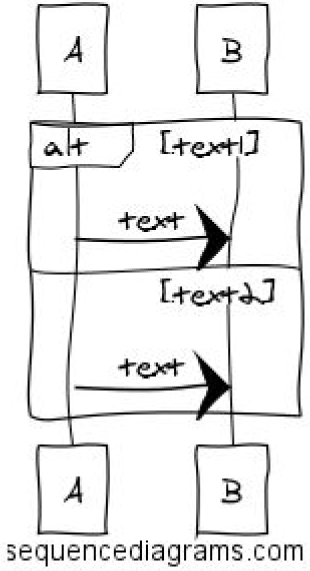


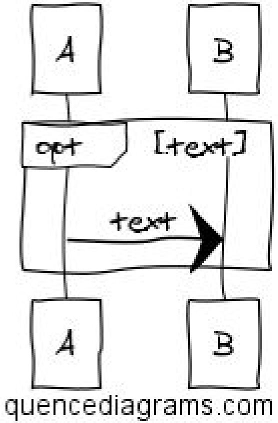
- Blocking Messages

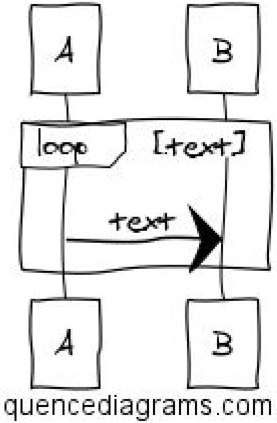


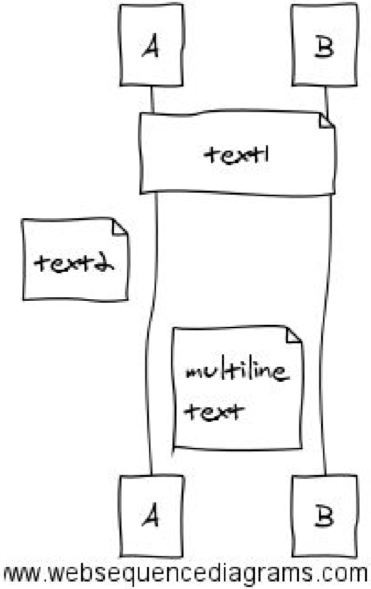
Sequence Diagram: Combined Fragment

- Alt
- Opt
- Loop
- Notes



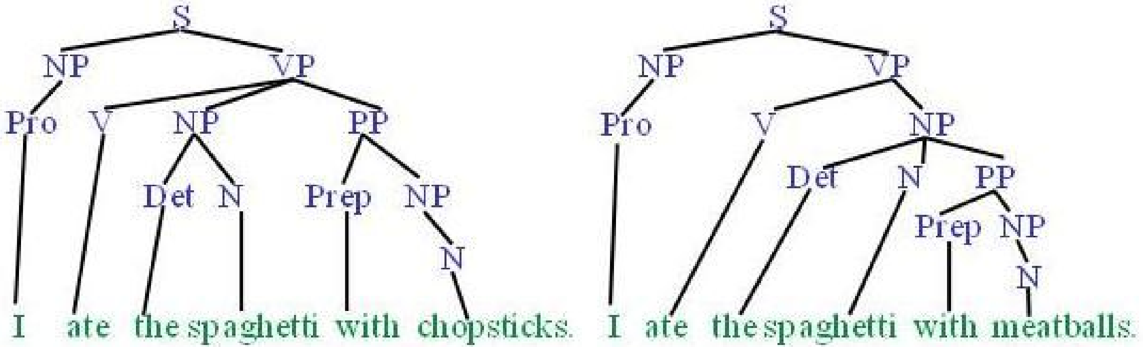






Natural Languages

- Ambiguous
- Complex
- Hard to process

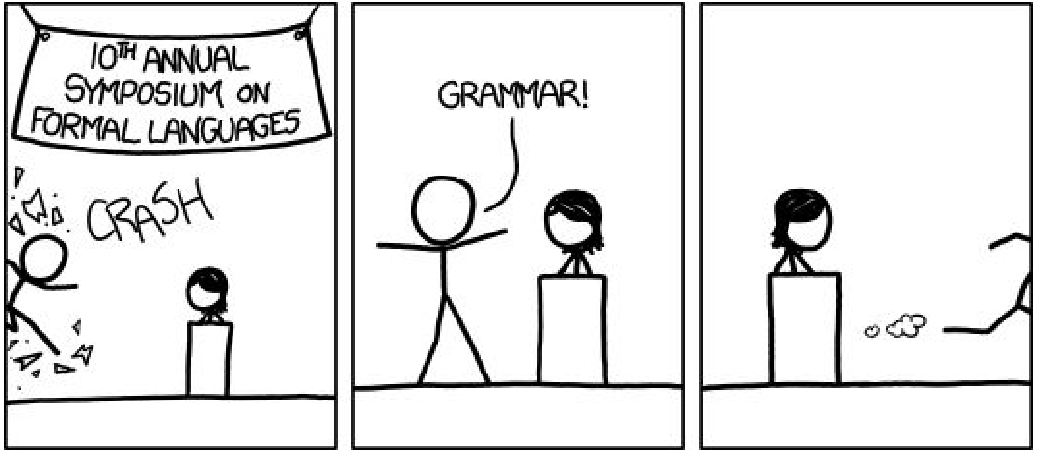

S
NP
VP
Pro
V
NP
PP
Det
N
Prep
NP
N
I ate the spaghetti with chopsticks.
S
NP
VP
Pro
V
NP
Det
N
PP
Prep
NP
N
I ate the spaghetti with meatballs.

https://www.cs.utexas.edu/~mooney/cs388/

Controlled Natural Languages

- Control them by using templates

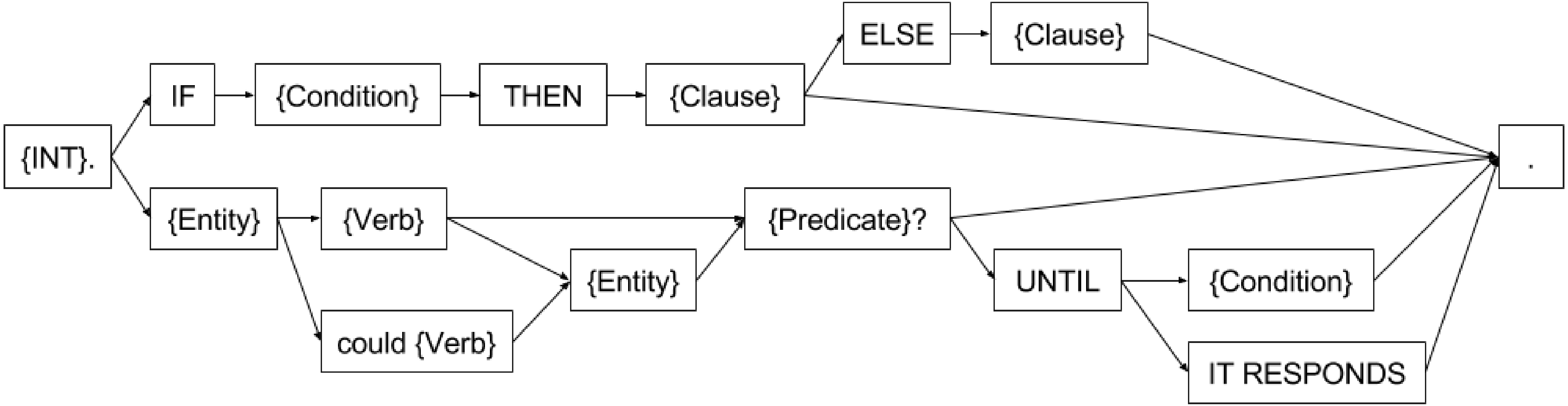
- Normalize them



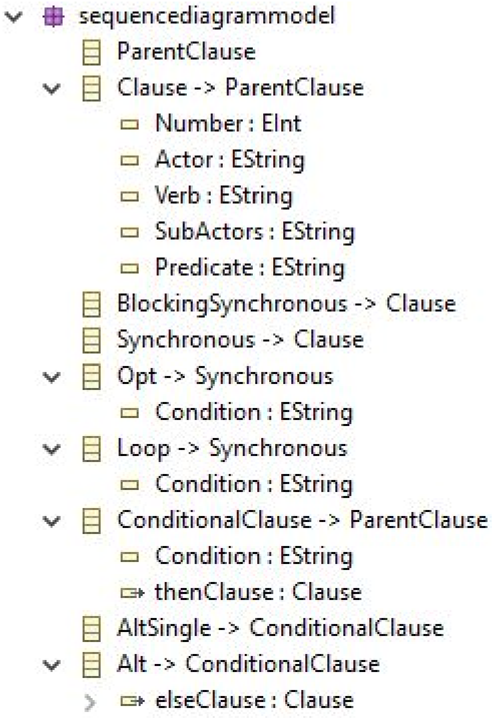
https://xkcd.com/1090/

First Step: Specify the Grammar

Interesting bit: Main Success Scenario Steps



Second Step: M2M Transformation

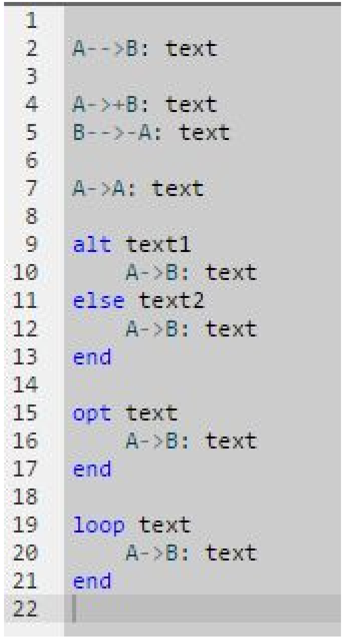


```
sequencediagrammodel
    ParentClause
    Clause -> ParentClause
        Number : EInt
        Actor : EString
        Verb : EString
        SubActors : EString
        Predicate : EString
    BlockingSynchronous -> Clause
    Synchronous -> Clause
    Opt -> Synchronous
        Condition : EString
    Loop -> Synchronous
        Condition : EString
    ConditionalClause -> ParentClause
        Condition : EString
        thenClause : Clause
    AltSingle -> ConditionalClause
    Alt -> ConditionalClause
        elseClause : Clause
```

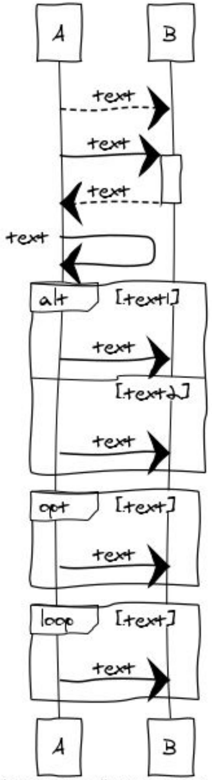
- Specify Target Ecore Model
- UseCaseM2M.java traverses the parsed use case
- UseCaseToSequenceDiagram.xtend helps to instantiate elements of the target Ecode Model

Third Step: M2T Code Generation

Ecore Sequence
Diagram
Instance Model



```

A-->B: text

A->+B: text
B-->-A: text

A->A: text

alt text1
    A->B: text
else text2
    A->B: text
end

opt text
    A->B: text
end

loop text
    A->B: text
end

```



ebsequencediagrams.com

Example

actor: Teller

secondary: System

scenario: {

1. Teller "requests" the System "to deposit money on an account, providing sum of money".

2. the System "validates" itself "the deposit, credit account with the requested amount, records details of the transaction".

3. the System "informs" Teller "that deposit was successful and waits" until it responds.

}

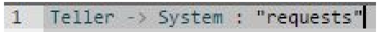
1. Teller "requests" the System "to deposit money on an account, providing sum of money".

Entity: Teller

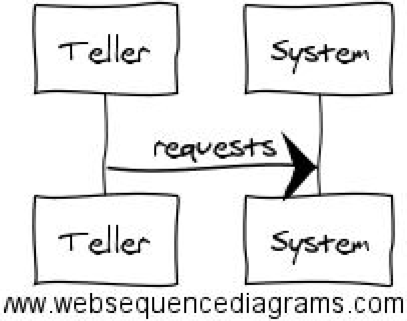
Verb: “requests”

Entity: the System

Predicate: "to deposit money on an account, providing sum of money"



```
Teller -> System : "requests"
```



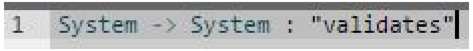
2. the System "validates" itself "the deposit, credit account with the requested amount, records details of the transaction".

Entity: the System

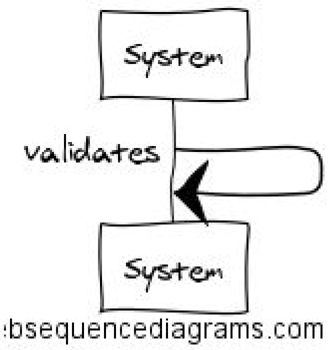
Verb: “validates”

Entity: itself

Predicate: "the deposit, credit account with the requested amount, records details of the transaction"



```
System -> System : "validates"
```



3. the System "informs" Teller "that deposit was successful and waits" until it responds.

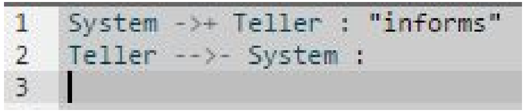
Entity: the System

Verb: “informs”

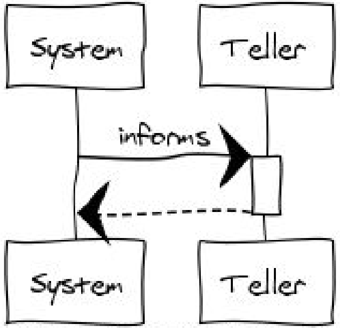
Entity: Teller

Predicate: "that deposit was successful and waits"

Time: until it responds



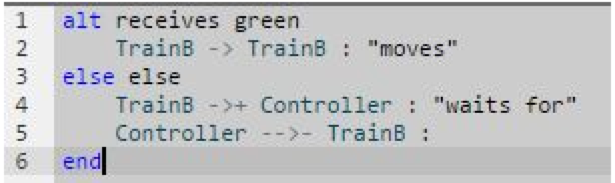
```
System ->+ Teller : "informs"
Teller -->- System :
```



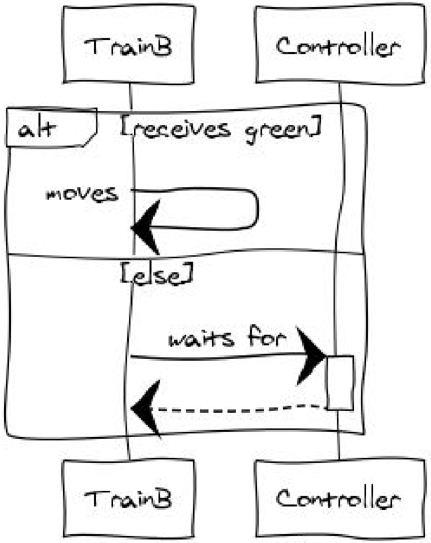
www.websequencediagrams.com

Combined Fragments Example: alt

7. if "receives green" then the TrainB "moves" itself "to the exit," else the TrainB "waits for" the controller until it responds.



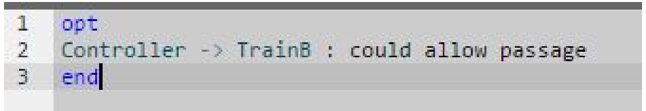
```
alt receives green
    TrainB -> TrainB : "moves"
else else
    TrainB ->+ Controller : "waits for"
    Controller -->- TrainB :
end
```



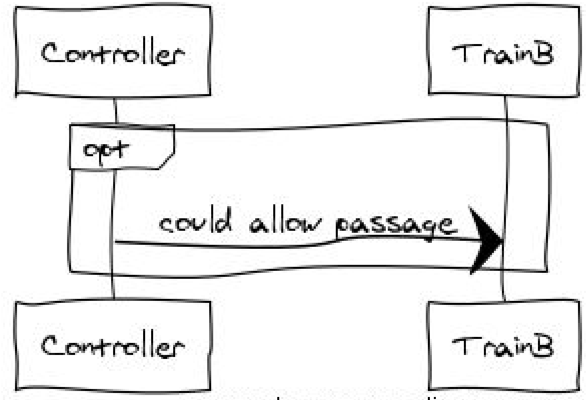
www.websequencediagrams.com

Combined Fragments Example: opt

6. the Controller could "allow passage" to TrainB "if it wishes".



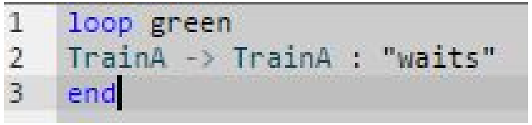
```
opt
Controller -> TrainB : could allow passage
end
```



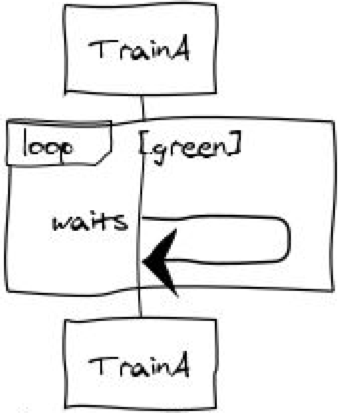
www.websequencediagrams.com

Combined Fragment Example: loop

3. TrainA "waits" itself until "green".

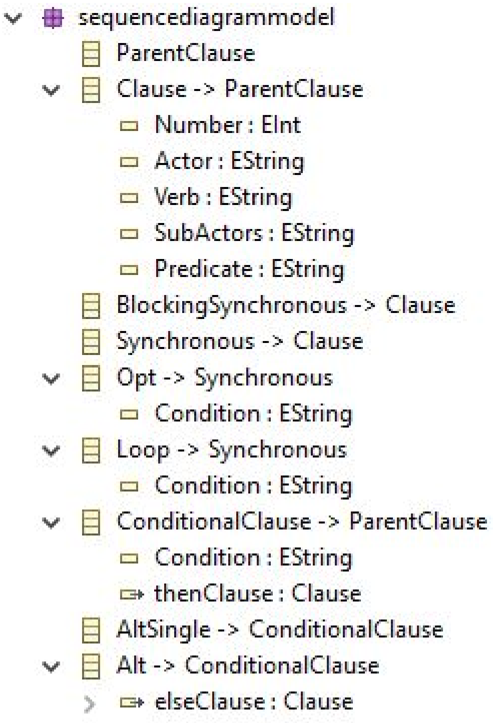


```
loop green
TrainA -> TrainA : "waits"
end
```

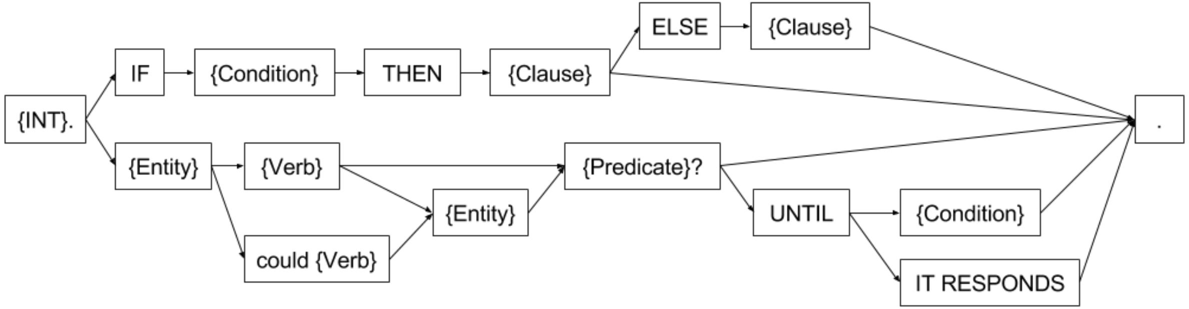


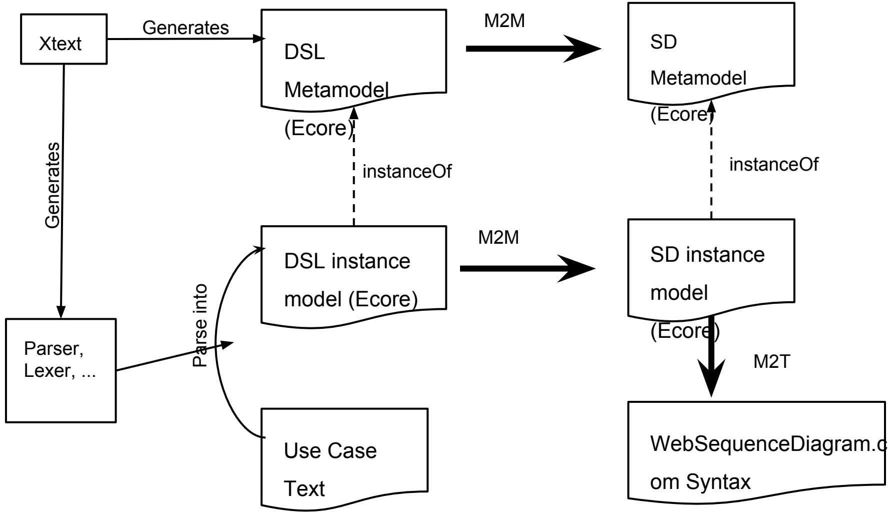
websequencediagrams.com

Textual Use Case DSL with Sequence Diagram Transformation



- sequencediagrammodel
  - ParentClause
  - Clause -> ParentClause
    - Number : EInt
    - Actor : EString
    - Verb : EString
    - SubActors : EString
    - Predicate : EString
  - BlockingSynchronous -> Clause
  - Synchronous -> Clause
  - Opt -> Synchronous
    - Condition : EString
  - Loop -> Synchronous
    - Condition : EString
  - ConditionalClause -> ParentClause
    - Condition : EString
    - thenClause : Clause
  - AltSingle -> ConditionalClause
  - Alt -> ConditionalClause
    - elseClause : Clause


{INT}.
IF
{Condition}
THEN
{Clause}
ELSE
{Clause}
.
{Entity}
{Verb}
{Predicate}?
{Entity}
could {Verb}
UNTIL
{Condition}
IT RESPONDS


Xtext
Generates
DSL
Metamodel
(Ecore)
M2M
SD
Metamodel
(Ecore)
Generates
instanceOf
instanceOf
DSL instance
model (Ecore)
M2M
SD instance
model
(Ecore)
Parser,
Lexer, ...
Parse into
M2T
Use Case
Text
WebSequenceDiagram.com Syntax