# Role-Playing Game Modeling in VMTS Dylan Kiss

University of Antwerp dylan.kiss@student.uantwerpen.be

- About VMTS
- The RPG formalism
- Abstract syntax
- Operational semantics
- VMTS vs. AToMPM
- Conclusion

#### About VMTS

- The RPG formalism
- Abstract syntax
- Operational semantics
- VMTS vs. AToMPM
- Conclusion

# About VMTS

- Graph-based metamodeling and model processing framework
- N-layer
- Budapest University of Technology and Economics
- Allows you to define, customize and utilize
  - Languages
  - Transformations



T. Levendovszky, L. Lengyel, G. Mezei, H. Charaf, A Systematic Approach to Metamodeling Environments and Model Transformation Systems in VMTS, Electronic Notes in Theoretical Computer Science 127 (1) (2005) 65-75

#### About VMTS

#### The RPG formalism

- Abstract syntax
- Operational semantics
- VMTS vs. AToMPM
- Conclusion

# The RPG formalism

- One or more scenes
- Each scene consists of connected tiles
   EmptyTile, Obstacle, Trap or Door
- Two types of characters
  - Hero (exactly one) and Villain
- Three types of items
  - Goal (at least one), Weapon and Key
- Non-obstacle tile can contain one character and item

- About VMTS
- The RPG formalism
- Abstract syntax
- Operational semantics
- VMTS vs. AToMPM
- Conclusion



### Abstract syntax



### Abstract syntax: constraints

	View properties	Constraints	Matlab	VAF	Model Tran	sformation	VMP		
efi ne	ine ck all ck selected item(s)	context [N inv: true	IAME]			Evaluate	Define Generate va	lidation co	ode
		Insta	nce constraints				Define c	onstraints	
ſ	🛞 Define Constrants	;				$\Leftrightarrow$		×	
	Context		Method N	ame	Invariant				а
	TileMeta	Ŧ	rightNBConsist	ent	<b>V</b>	Edit code	Delet	te	H
	TileMeta	Ŧ	leftNBConsister	nt	<b>V</b>	Edit code	Delet	e	,
e	TileMeta	Ŧ	topNBConsister	nt	<b>√</b>	Edit code	Delet	e	F
2	TileMeta	•	bottomNBCons	istent	1	Edit code	Delet	e	ŀe
đ	ObstacleMeta	-	nothingOnObta	acle	1	Edit code	Delet	te	
e a	RPGMeta	-	onlyOneHero		1	Edit code	Delet	te	V
a	RPGMeta	•	goalExists		1	Edit code	Delet	te	Ľ
e		•			<b>√</b>	Edit code	Delet	te	
a									
t				Save					

### Abstract syntax: constraints

<b>V</b>	$\leftrightarrow$		Ж
OCL C#			
package Constraints context TileMeta inv rightNBConsistent:			*
<pre>1 (not self.RightTileEnd.oclIsUndefined()) implies (self.RightTileEnd endpackage</pre>	i.LeftTileEnd	i = self)	-
Use 💿 OCL 🔘 C#	ОК	Cance	



### Abstract syntax: constraints

package Constraints
 context ObstacleMeta
 inv nothingOnObstacle:
 self.Character.oclIsUndefined()
endpackage

package Constraints
 context RPGMeta
 inv onlyOneHero:
 self.HeroMeta->size() = 1
endpackage

```
package Constraints
  context RPGMeta
    inv goalExists:
        self.GoalMeta->size() > 0
endpackage
```

### Abstract syntax: example model



- About VMTS
- The RPG formalism
- Abstract syntax
- Operational semantics
- VMTS vs. AToMPM
- Conclusion

## **Operational semantics**

- Based on graph rewriting
- Two steps
  - Creating rewrite rules
  - Creating a transformation control flow

# Operational semantics: rewrite rules

- New model of type MTRMETA
  - First: metamodel(s)



agram on Rule\_MoveHeroRight



# Operational semantics: rewrite

IUICS		Matched Relation Replaced Reference				
	andrainte Mallab VAE Madal Teachannation	➡ Matched Reference 📰 Rule Node				
Pron Mode	Model transformation VMP	E Replaced Relation				
DropOne Mode	ed Reference 📰 Rule Node	Elemente				
Autoplace Mode	ced Relation	Elements				
Project Explorer	xplorer X X Diagram on Kule MoveHeroRight					
	100% O	· 📄 🖬				
<ul> <li>▲ ■ RPGProject</li> <li>▶ ☎ RPGMeta</li> <li>▶ ☎ RPGModel</li> <li>▶ ☎ RPGModelSimple</li> <li>▶ ☎ Rule_GoalsNotCollected</li> <li>▶ ☎ Rule_MoveHeroBottom</li> </ul>	ContainsCharacter	haracterReplaced				
I I Rule_MoveHeroLeft     I II Rule_MoveHeroRight	igntivelgbor					
<ul> <li>▲ □ Diagrams</li> <li>▲ □ Diagram or u</li> <li>▶ □ DstTile</li> <li>▶ □ Hero</li> </ul>	Name: SrcTile	Matched from model 0 -				
Basic settings Name: SccTile		Matched from model				
Advanced settings						
Rule node meta types: <u>Empty lile</u> Constraints	add type remove all (CTRL +					
Advanced processing configurati	ion of rule node: <u>not modified</u>   <u>deleted</u>   <u>modified</u>   <u>c</u>	rreated 🗹 Clear obsolete attributes				
Match from selection Sel	lect Reference					
	Projects					
	🗏 HeroMeta					

# Operational semantics: rewrite rules

- Compiling rule constraints and actions failed
- Restrictions:
  - There are **no villains** in the game
  - There are only two types of tiles in the game: EmptyTile and Obstacle
  - There is only one type of item in the game: Goal
- 6 rewrite rules:
  - MoveHeroLeft/Right/Top/Bottom
  - PickUpGoal
  - GoalsNotCollected

# Operational semantics: rewrite rules

MoveHeroRight

SrcTile containsCharacter Hero containsCharacterReplaced DstTile EmptyTileMeta
Model 0
rightNeigbor

PickUpGoal



GoalsNotCollected



### Operational semantics: transformation control flow

- New model of type TCFMETA
  - First: metamodel(s)

Model Tran	sformation	VMP				
Refactorings	efactorings		C <sup>#</sup> Edit Imperative Code Match to multiple place options Edit Variab Edit Storag		Configuration	
		🗊 Select	Metamodels	Contro		X
10 RuleCo tainer: Rule_Mo oLeft	Rul tair Rule oal fE6	0	Metamodel RPGMeta	nish	Ac Rem Move Up Mod	ld ove Move Down dify d metas with

### Operational semantics: transformation control flow



### Operational semantics: transformation control flow

- No support for randomness
  - Deterministic order of rules
  - Unable to properly simulate an RPG



- About VMTS
- The RPG formalism
- Abstract syntax
- Operational semantics
- VMTS vs. AToMPM
- Conclusion

- Abstract syntax
  - Very similar in both tools
  - Simplified class diagram
  - Constraints
  - VMTS: n-layer

- Concrete visual syntax
  - Major difference
  - **AToMPM**: built-in visual syntax creation
  - VMTS: create your own plugin for visualization using WPF (Windows Presentation Foundation)

- Operational and denotational semantics
  - Both use graph rewriting
  - Visual representation differs
  - **ATOMPM**: clearly divided LHS, RHS, NAC
  - VMTS: one graph, no NAC

- Modeling environment
  - **AToMPM**: cross-platform, web-based
  - VMTS: written in C# using .NET framework, only Windows platform

- About VMTS
- The RPG formalism
- Abstract syntax
- Operational semantics
- VMTS vs. AToMPM
- Conclusion

# Conclusion

- Intentional plan
  - Abstract syntax
  - Concrete visual syntax
  - Operational semantics
  - Denotational semantics

# Conclusion

- Intentional plan
  - Abstract syntax
  - Concrete visual syntax
    - requires developing plugin with WPF
  - Operational semantics
    - partly possible: constraints/actions/NAC/randomness
  - Denotational semantics
    - Impossible: constraints/NAC

# Conclusion

#### VMTS

- + n-layer
- + default visual syntax
- AToMPM
  - + web-based tool
  - + built-in visual syntax editor
  - + randomness and NAC

My experience: AToMPM more user-friendly

# **Questions?**