A Systematic Comparison of *i** Modelling Tools Based on Syntactic and Well-formedness Rules

Catarina Almeida, *Miguel Goulão, João Araújo*

CITI, FCT, Universidade Nova de Lisboa, Portugal

acg.almeida@campus.fct.unl.pt

{mgoul, joao.araujo}@fct.unl.pt

Roadmap

- Introduction
- Objectives of the Research
- Analysed Tools
- *i** Syntax Converage
- Well-formedness Rules
- Conclusions

Introduction (I)

- There are several *i** variations: Yu'95, TROPOS, Secure Tropos, Iterative Tropos, GRL
- There are several tools available to create *i** models
- Different tools provide different kinds of support for the specification of an *i** model

Introduction (II)

The wiki page includes a comparison of the *i** tools, which covers:

- the purpose of the tool
- the *i** framework it supports
- details on availability, base platform, maturity
- details on the tool modelling suitability, usability, extensability and interoperability

We present: **a comparison of syntactic and semanctic features supported by the different *i** tool**

Objectives of the Research

Answer two research questions:

RQ1: Which of the syntactic constructs described in the *i** wiki are supported by each *i** tool?

RQ2: To what extent does each *i** tool support semantic checking of the *i** models built using it?

Analysed Tools (I)

Inclusion criteria:

- Presence in the *i** wiki page
- Availability of a functional URL

*i** Tool	Institution	*i** Variant	Platform	Technology
OpenOME	Univ. Toronto	Yu'95	All	Java (JRE)
TAOM4E	Univ. Trento	Tropos	All	Eclipse plug-in
GR-Tool	Univ. Trento	Tropos	All	Java (JRE)
STS-Tool	Univ. Trento	Trops	All	Java (JRE)
jUCMNav	Univ. Ottawa	GRL	All	Eclipse plug-in
DesCARTES	U. C. Louvain	Yu'95 / Tropos	All	Eclipse plug-in

Analysed Tools (II)

OpenOME

Eclipse-based tool designed to support goal-oriented, agent-oriented and aspects-oriented modelling and analysis

TAOM4E

Eclipse plug-in that supports a model-driven, agent-oriented software development

Analysed Tools (III)

GR-Tool

Graphical tool for forward and backward goal reasoning in Tropos

STS-Tool

Socio-technical security modelling tool to draw Tropos and Secure Tropos models and to perform the effective formal analysis of functional and security requirements

Analysed Tools (IV)

jUCMNav

Eclipse plug-in for modelling, analysis and transformation in both GRL and UCM (Use Case Map)

DesCARTES

Eclipse plug-in that allows the development of the methodology analysis and design models as well as forward engineering capabilities and an integrated software project management module

*i** Syntax Coverage (I)

Aims to check if the tool has:

a) the basic *i** syntax, and

b) the graphical notation of the *i**

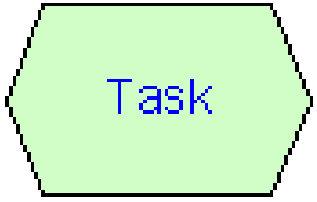
(according to the *i** wiki page)

*i** Syntax Coverage (II)

Elements

DesCARTES

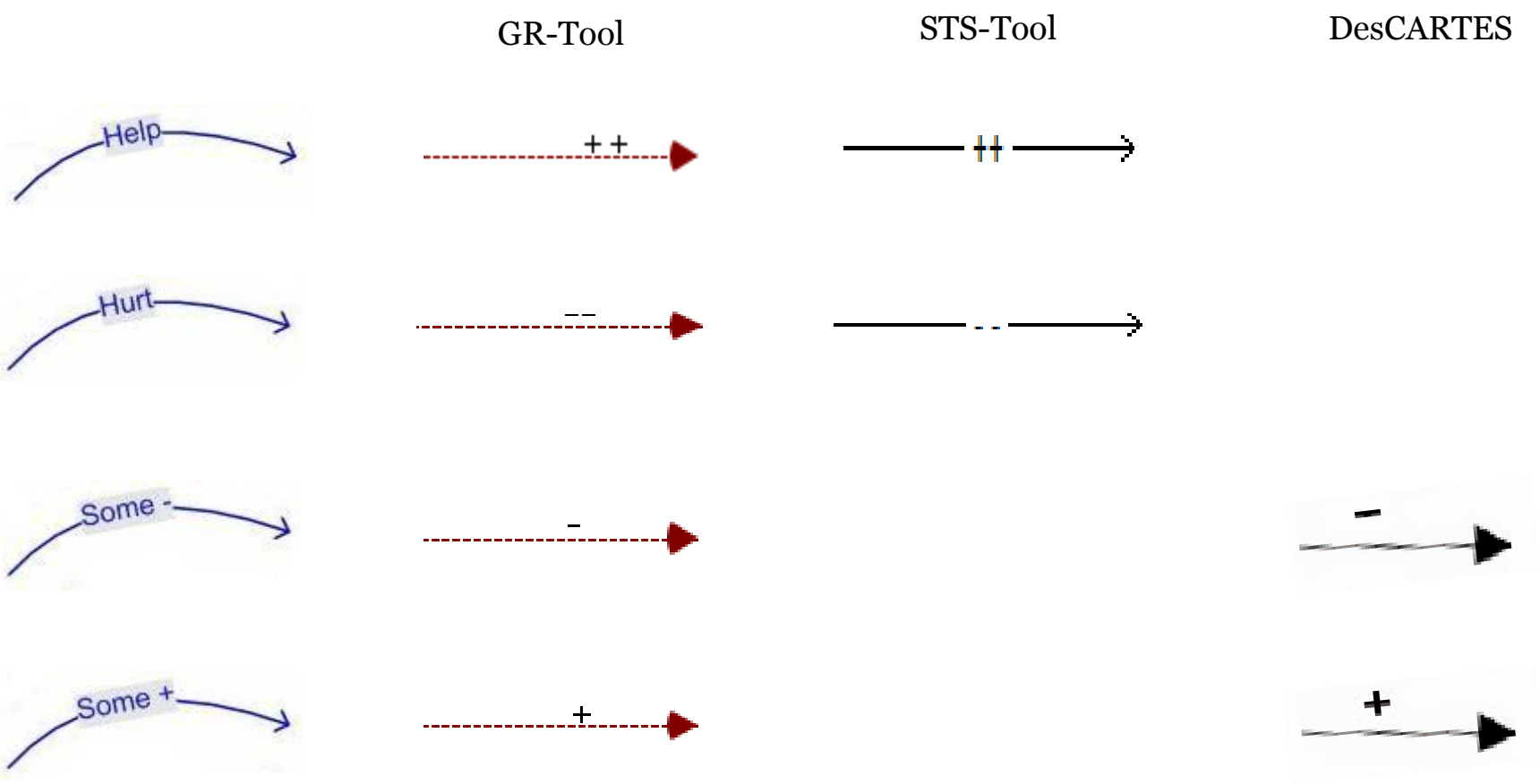
Alternative notation



TAOM4E

*i** Syntax Coverage (III)

Links and Contribution Links



*i** Syntax Coverage (IV)

Discussion

- All the tools support goals, the "and" link and have at least two types of contribution links
- It is in the contribution links that the variation of the graphical notation is higher
- OpenOME is the tool with the widest syntax coverage according to the two criteria

Well-formedness Rules (I)

- Determine the level of correctness checking of the created models (using the descriptions and guidelines available in the *i** wiki page)

- Analyse if the tool checks when a modelling error is made

Well-formedness Rules (II)

Actors and Dependencies

	OpenOME	TAOM4E	GR-Tool	STS-Tool	jUCMNav	DesCARTES
Actors and relations						
Actors without links	Yes*	No	N/A	No	No	No
Actor inside another actor boundary	Yes	Yes	N/A	Yes	No	Yes
Dependencies						
Dependency link without a dependum	Yes*	Yes	N/A	N/A	Yes	Yes
Dependency links with different directions	No	Yes	N/A	N/A	No	Yes
Dependency link inside an actor boundary	Yes*	Yes	N/A	N/A	Yes	N/A
Other link rather than dependency link between an element and an actor	Yes	Yes	N/A	Yes	No	Yes

Well-formedness Rules (III)

Associations

	OpenOME	TAOM4E	GR-Tool	STS-Tool	jUCMNav	DesCARTES
Associations						
ISA between actors of different types	No	N/A	N/A	N/A	Yes	N/A
Is-part-of between actors of different types	No	N/A	N/A	N/A	Yes	N/A
Other association rather than Plays between Agent and Role	No	N/A	N/A	Yes	Yes	N/A
Other association rather than Covers between Position and Role	No	N/A	N/A	N/A	Yes	N/A
Other association rather than Occupies between Agent and Position	No	N/A	N/A	N/A	Yes	N/A
INS between others than agents	No	N/A	N/A	N/A	Yes	N/A
Associations between elements that are not actors	Yes	N/A	N/A	Yes	Yes	N/A

Well-formedness Rules (IV)

Internal Elements

	OpenOME	TAOM4E	GR-Tool	STS-Tool	jUCMNav	DesCARTES
Internal Elements						
SR elements outside actor boundary	No	Yes	N/A	Yes	No	N/A
Softgoal decomposition in sub-softgoals or sub-tasks	No	No	N/A	N/A	No	Yes
Goal decomposition in sub-goals or sub-taks	Yes*	No	N/A	N/A	No	Yes
Goal decomposition without means-end	No	No	N/A	N/A	No	Yes
Means-end where a goal is the mean	No	No	N/A	N/A	Yes	No
Means-end different from "task–>goal"	No	No	N/A	N/A	No	No
Decomposition beyond the actor boundary	No	Yes	N/A	N/A	No	N/A
Means-end beyond the actor boundary	No	Yes	N/A	N/A	No	N/A
Means-end decomposition to refine a softgoal	No	No	N/A	N/A	No	Yes
Softgoal decomposition without contribution links	No	No	N/A	N/A	No	No
Any kind of direct relation between goals	Yes*	No	N/A	N/A	No	Yes
Link between an element inside the actor boundary and that actor	Yes	Yes	N/A	Yes	Yes	N/A

Well-formedness Rules (V)

Contribution Links

	OpenOME	TAOM4E	GR-Tool	STS-Tool	jUCMNav	DesCARTES
Contribution Links						
Contribution links between any element to any element rather than softgoal	No	No	N/A	N/A	Yes	No
Contribution link between actors	Yes	Yes	N/A	Yes	Yes	Yes
Contribution link between goals and sub-goals or sub-tasks	No	No	No	Yes	Yes	No

Well-formedness Rules (VI)

Discussion

- On average, about 39% of the considered modelling erros are not applicable

- jUCMNav has the highest number of verified errors, with a verification percentage of 50%, followed by OpenOME and TAOM4E

Conclusions

- The tools present a great variation of the *i** syntax, usually alligned with one of the *i** frameworks

- Error detection is not a common practice, since that less than 50% of the errors are verified

Questions