Critic Authoring Templates for Specifying Domain-Specific Visual Language Tool Critics

Norhayati Mohd.Ali[1], John Hosking[1], Jun Huh[1]
and John Grundy[1,2]
[1]Department of Computer Science and
[2]Department of Electrical and Computer Engineering,
University of Auckland, New Zealand.

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Outline

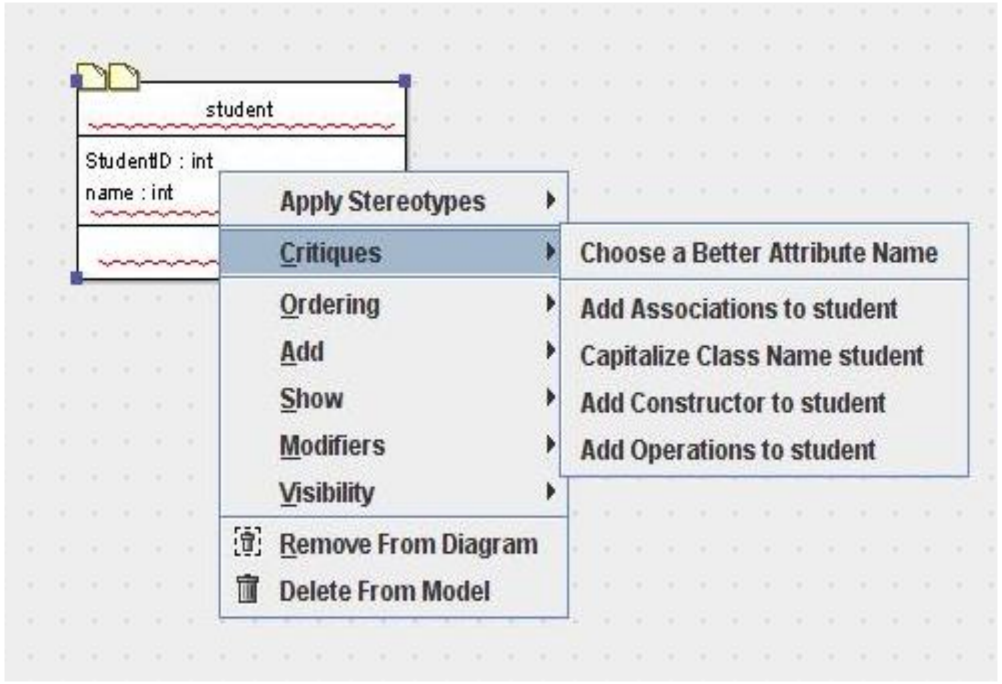
Introduction

- Miller described ‘*critic*’ as a software program that critiques human-generated solutions[13]
- Critic tools have demonstrated effectiveness in providing feedbacks.
- However, there has been little discussion of *critic authoring*.
- Our aim: is to describe a new approach using visual critic authoring templates to support tool & end user designers in specifying design critics

Background & Motivation

- Related work:
  - ArgoUML [3]-advise designer when a software architecture diagram violates the UML rules
  - IDEA [4]-specify critics that focus on design patterns to improve the UML model
  - JavaCritiquer [16]-detects statements in a student program code that can be improved
  - ABCDE-Critic [20]-specify critics that comment on UML class diagram-based designs

- List of critiques in ArgoUML tool

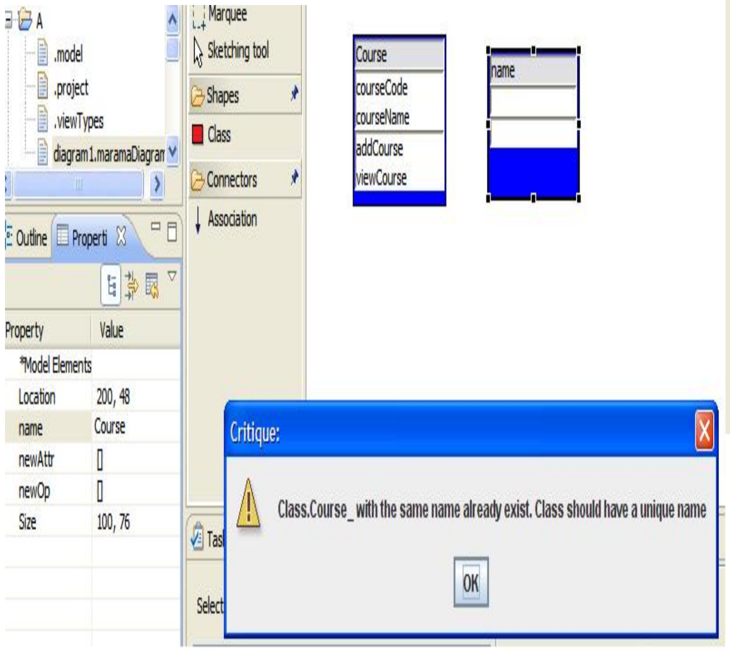

student
StudentID : int
name : int
Apply Stereotypes
Critiques
Ordering
Add
Show
Modifiers
Visibility
Remove From Diagram
Delete From Model
Choose a Better Attribute Name
Add Associations to student
Capitalize Class Name student
Add Constructor to student
Add Operations to student

Background & Motivation

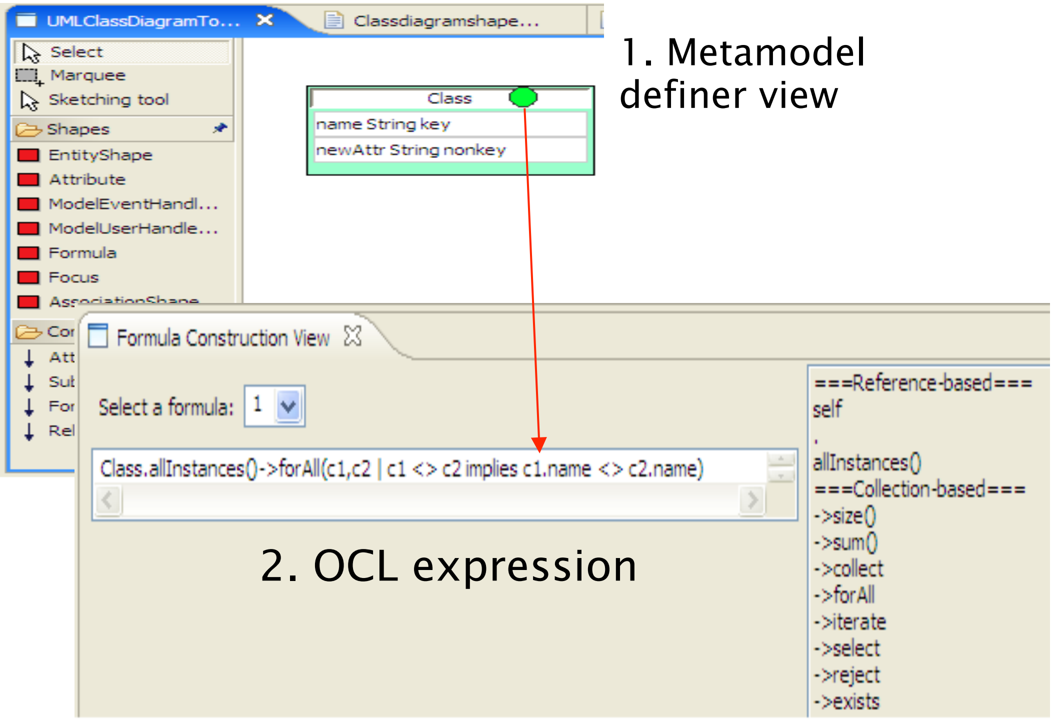
- Variety of approaches can be used in designing and realizing critics:
  - Rule-based
  - OCL expressions
  - Knowledge-based
  - Pattern-matching
  - Programming code
  - etc

Background & Motivation

- Motivating Example:



Class.Course_ with the same name already exist. Class should have a unique name



```
Class.allInstances()->forAll(c1,c2 | c1 <> c2 implies c1.name <> c2.name)
```

1. Metamodel definer view

2. OCL expression

3. Simple critic (same named class) violation

Background & Motivation

- Those approaches:
  - Require deep understanding of the tool platform
  - Customization of critics would not be easy

- Little attention has been given to provide an authoring facility for user to add/modify critics
  - ArgoUML [3]
  - IDEA [4]
  - Java Critiquer [16]
  - ABCDE-Critic [20]

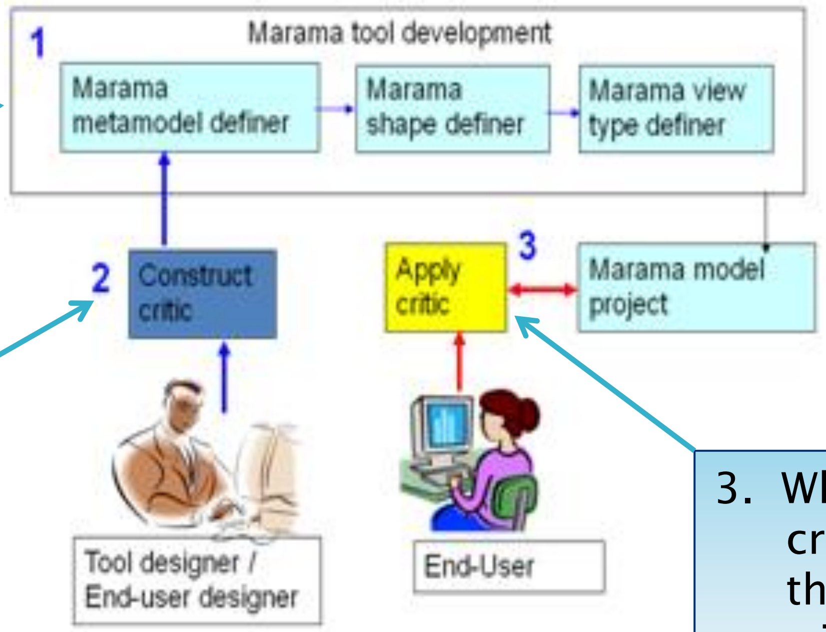
Background & Motivation

- The extension :
  - To use a visual design notation to represent critics
  - To specify and design critics in a simple way by using an easy-to-use, high-level language
  - To allow critic authoring based on business rule templates
  - To provide a new meta-tool facility for our Marama-based tools

Our Approach

- Marama visual critic development approach

1. Designer uses the Marama meta-tools to develop Marama-based tools



2. Critics are specified via Marama metamodel definer view

3. When a diagram is created, critics for that particular tool will be applied.

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Critic Authoring Template

- Inspired by Business Rules-driven Object Oriented Design (BROOD) approach [11,23]
- Proposed a Business Rule (BR) template that contains three main types [11,23]:
  - Constraint (attribute constraint & relationship constraint)
  - Action assertion
  - Derivation
- The rule templates are formal sentence patterns that allow the expression of business rules [11,23]

Critic Authoring Template

- Why we use the templates in software tool domain (visual critic authoring tool):
  - The use of language definition based on the context-free grammar EBNF that defines sentence patterns for rule statements
  - The use of natural language that is easily understood to represent the rules
  - The templates are more general in nature and are easily adapted for use in the critic domain

Critic Authoring Template

- Initially we only covers the attribute constraint templates and relationship constraint templates
- The critic rules templates that correspond to the attribute and relationship constraints are as follow:

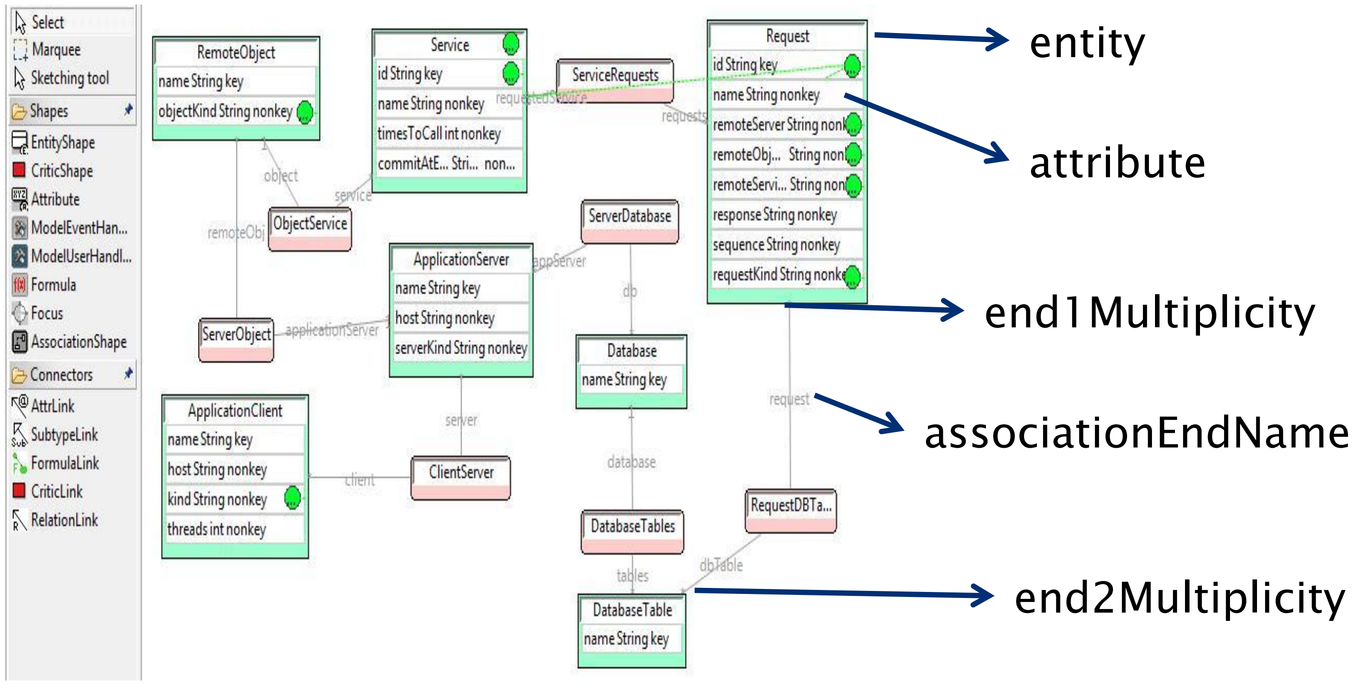
Critic Authoring Template

- Attribute and relationship constraint templates [23]

Attribute Constraint	<entity> must have | may have a [unique] <attributeTerm> <attributeTerm1>must be |may be <relationalOperator><value>| <attributeTerm2>
Relationship Constraint	[<cardinality>]<entity1> is a/an <role> of [<cardinality>]<entity2> [<cardinality>]<entity1> is associated with [<cardinality>] <entity2> <entity1> must have |may have [<cardinality>]<entity2> <entity1> is a/an <entity2>

Example Usage

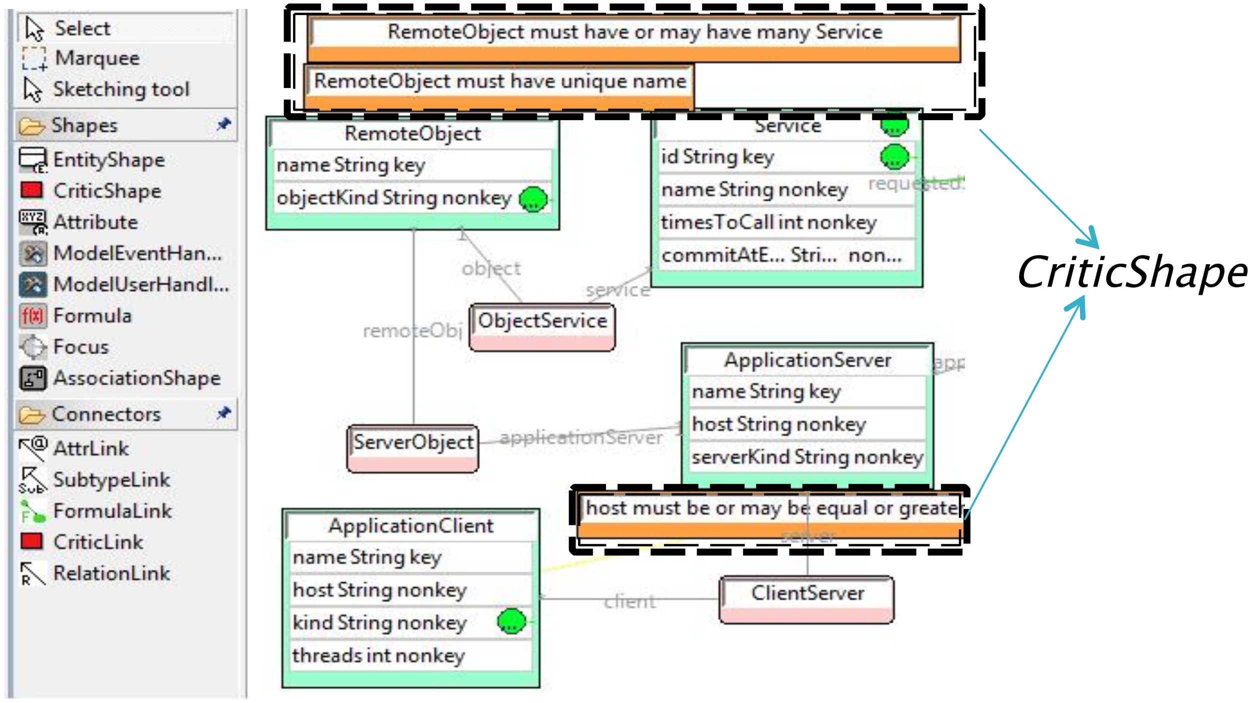
- We illustrate the use of critic authoring facilities via MaramaMTE software architecture design tool [8]



MaramaMTE metamodel definer view

Example Usage

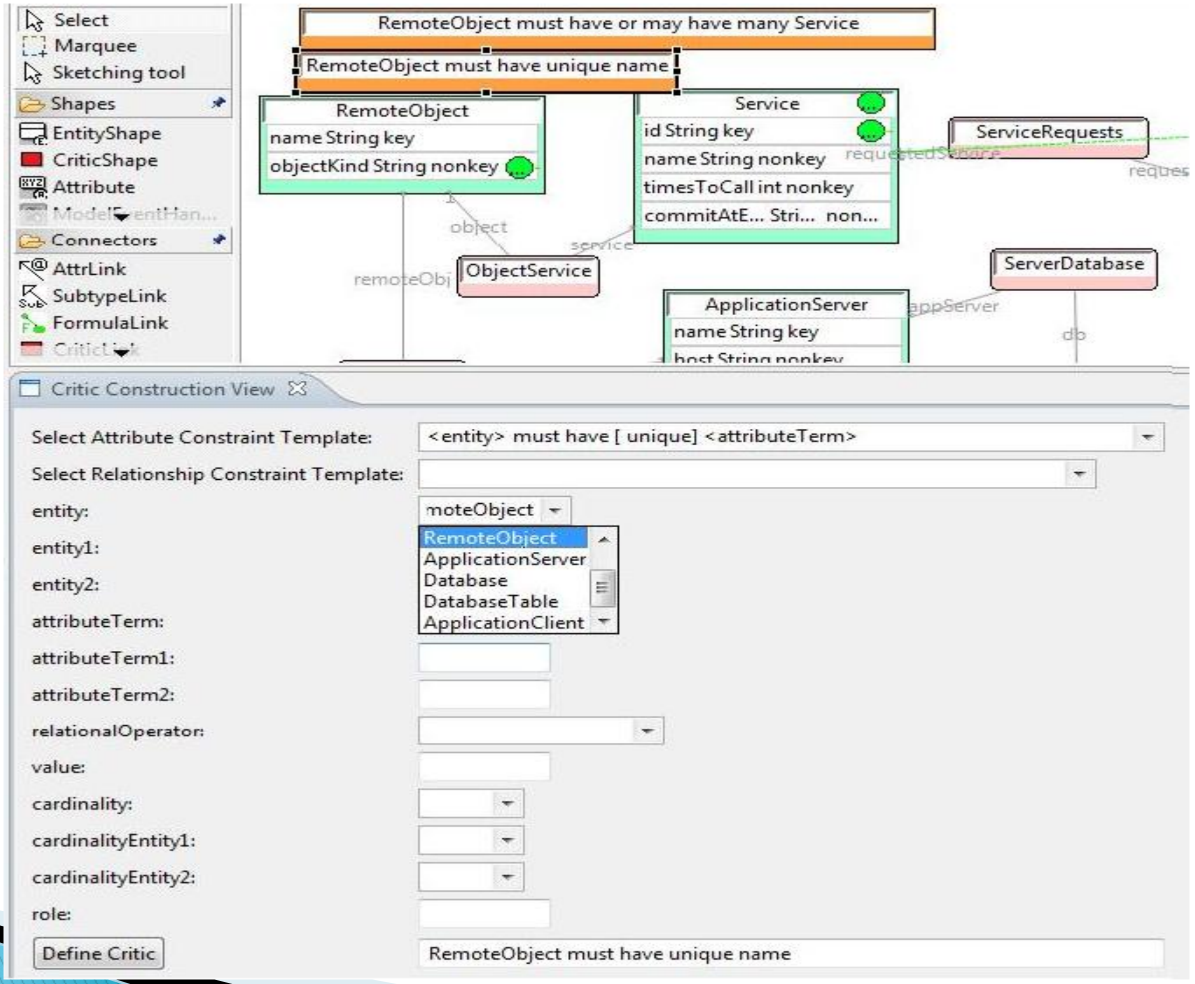
- MaramaMTE with critic function



CriticShape functions added to the MaramaMTE metamodel definer view

Example Usage

- Critic construction view:



Select Attribute Constraint Template: <entity> must have [unique] <attributeTerm>

Select Relationship Constraint Template:

entity: moteObject

RemoteObject
ApplicationServer
Database
DatabaseTable
ApplicationClient

entity1:

entity2:

attributeTerm:

attributeTerm1:

attributeTerm2:

relationalOperator:

value:

cardinality:

cardinalityEntity1:

cardinalityEntity2:

role:

Define Critic

RemoteObject must have unique name

Example Usage

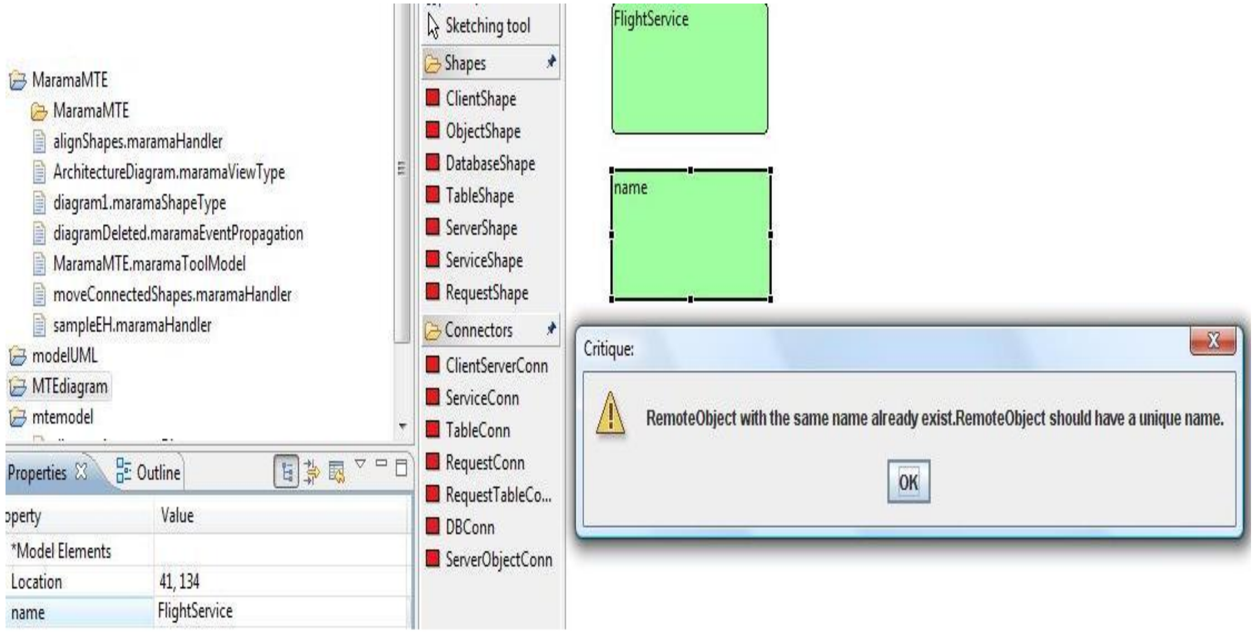
- Critics for MaramaMTE tool:

Entity	Critic statement	Critic rule template	Type
RemoteObject	RemoteObject must have a unique name	<entity> must a [unique] attributeTerm>	Attribute constraint
ApplicationServer	ServerKind must be equal one	<attributeTerm1>must be <relationalOperator> <value>	Attribute constraint
Request	Request must have many Services	<entity1> must have [<cardinality>]<entity2 >	Relationship constraint
ApplicationServer	One ApplicationServer is associated with many RemoteObject	[<cardinality>]<entity1 > is associated with [<cardinality>] <entity2>	Relationship constraint

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Example Usage

- Example of attribute constraint:



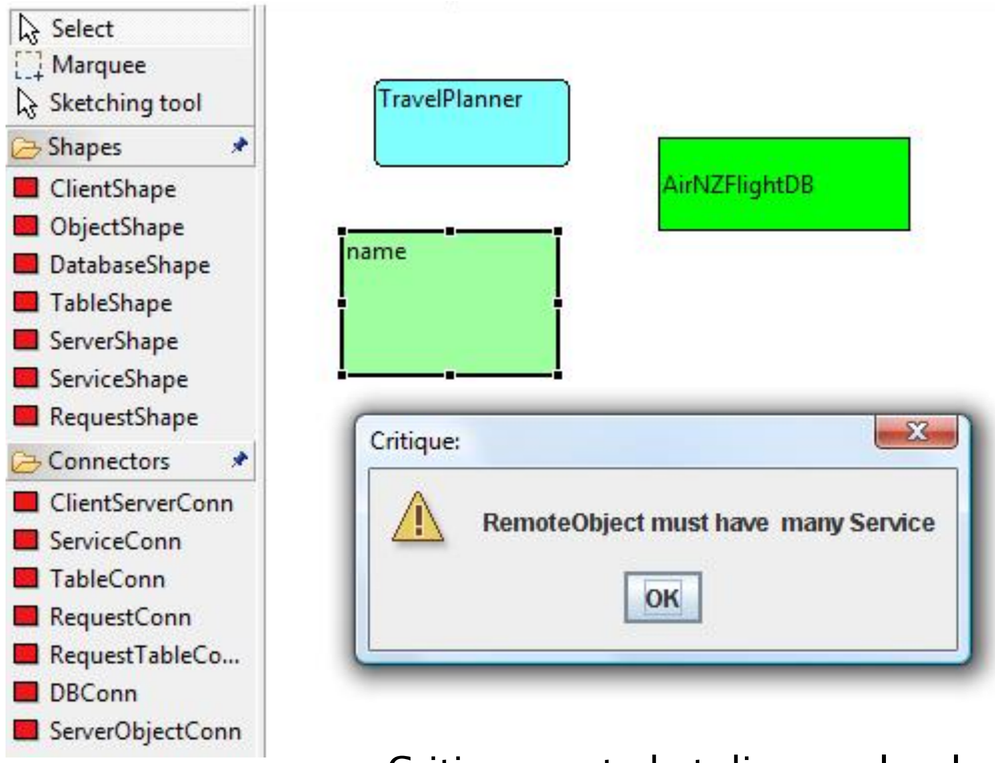
operty	Value
*Model Elements	
Location	41, 134
name	FlightService

RemoteObject with the same name already exist.RemoteObject should have a unique name.

Critic executed at diagram level

Example Usage

- Example of relationship constraint:



RemoteObject must have many Service

Critic executed at diagram level

Design & Implementation

- Create a new function (*CriticShape*) at the Marama metamodel editor
- Critic authoring template interface
- Critics repository
- Critic engine, event listener & critic processor
- Each critic as a concrete class

Discussions

- The critic authoring templates made it far easier and quicker
- Key benefits of the approach:
  - Provides a simple way to express critics;
  - Novice designer may easily construct the critics;
  - Offers a structured form in expressing the critic phrase
  - Marama instantiates critic rule processors when opening a tool and uses Marama's built-in event handler mechanism to proactively check changing designs

Discussions

- Main limitations:
  - Currently supports fairly simple critics construction
  - Critics can be defined only based upon the available templates
  - Very complex critics are not able to be specified via attribute and relationship constraint templates
  - Only limited actions are supported
  - The critic engine implemented in Marama uses a simple approach to determine interested critics

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Conclusions & Future Work

- Describes an approach for specifying and authoring critics
- Develops critic authoring templates (attribute and relationship constraint)
- Develops a prototype of the visual critic authoring template approach
- Illustrates the use of visual critic authoring tool

Conclusions & Future Work

- Provides a proof of concept that critic authoring templates support the construction of critics in a simple way for Marama-based tools
- Plans for future work include:
  - Construction of complex critics via action assertion and derivation templates
  - Creating critic feedback facilities
  - Expanding the critic authoring templates
  - Evaluation of the prototype by target users

Thank You

- Comp. Sc. Dept., UoA & Postgraduate Research Student Support account – funding support
- Prof. John Grundy
- Prof. John Hosking
- Jun Huh

References

- [1] Ali, N.M. A Generic Visual Critic Authoring Tool, Proceeding VLHCC'07, IEEE CS Press, 2007, pp.260-261.
- [2] Ali, N.M. Specifying Visual Design Critic Framework, Proceeding NZCSRSC'08, April 2008, Christchurch, New Zealand, pp.184-187.
- [3] ArgoUML, http://argouml.tigris.org/
- [4] Bergenti, F. and Poggi. A. Improving UML Designs Using Automatic Design Pattern Detection, In Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering (SEKE), 2000, pp. 336-343.
- [5] Eclipse, http://www.eclipse.org/
- [6] ExtendedBNF, http://www.cl.cam.ac.uk/~mgk25/iso-14977.pdf
- [7] Grundy, J.C., Hosking, J.G., Huh, J. and Li, N. Marama: an Eclipse meta-toolset for generating multi-view environments, Formal demonstration paper, 2008 IEEE/ACM International Conference on Software Engineering, Liepzig, Germany, May 2008, ACM Press. See also: *https://wiki.auckland.ac.nz/display/csidst*
- [8] Grundy, J.C., Hosking, J.G., Li, L. and Liu, N. Performance engineering of service compositions, ICSE 2006 Workshop on Service-oriented Software Engineering, Shanghai, May 2006.
- [9] Irandoust, H. 2006. Critiquing systems for decision support. DRDC Valcartier TR 2003-321. http://pubs.drdc.gc.ca/PDFS/unc44/p524782.pdf.
- [10] Liu, N., Hosking, J.G. and Grundy, J.C. MaramaTatau: Extending a domain specific visual language meta-tool with a declarative constraint mechanism, Proceeding VLHCC'07, IEEE CS Press, 2007, pp. 95-103.
- [11] Loucopoulus, P., and Wan Kadir, W.M.N. "BROOD:Business Rules-driven Object Oriented Design", *Journal of Database Management*, Vol.19, Issue 1, 2008, pp. 41-73.
- [12] Markowitz, V. Extended Entity Relationship Diagram, http://sdm.lbl.gov/OPM/DM_TOOLS/OPM/ER/ER.html
- [13] Miller, P. *Expert Critiquing Systems: Practice-based Medical Consultation by Computer*. Springer Verlag, New York, 1986.

References

- [14] Oh,Y., Do, E.Y.-L, and Gross, M.D., "Intellligent Critiquing of Design Sketches", in JL Randall Davis, T Stahovich, R Miller and E Saund (eds), *Making Pen-based Interaction Intelligent and Natural,* The AAAI Press, Arlington, Virginia, 2004, pp 127-133.
- [15] Oh,Y., Gross, M.D and Do, E.Y.-L, Computer-Aided Critiquing Systems, Lessons Learned and New Research Directions. http://code.arc.cmu.edu/lab/upload/caadria-oh.0.pdf
- [16] Qiu, L., and Riesbeck, C.K., "An Incremental Model for Developing Educational Critiquing Systems: Experiences with the Java Critiquer", *Journal of Interactive Learning Research*, 2008(19), pp.119-145.
- [17] Robbins, J.E. 1998. Design Critiquing Systems, Technical Report UCI-98-41. http://www.ics.uci.edu/~jrobbins/papers/CritiquingSurvey.pdf.
- [18] Robbins, J.E., Hilbert, D.M. Redmiles, D.F. Software Architecture Critics in Argo. Intelligent User Interfaces 1998, pp. 141-144
- [19] Robbins, J.E., Redmiles, D.F. Software architecture critics in the Argo design environment. Knowledge-Based Systems 11(1), 1998, pp. 47-60.
- [20] Souza, C.R.B., et al. A Group Critic System for Object-Oriented Analysis and Design, In Proceedings of the 15th IEEE Conference on Automated Software Engineering, IEEE Press, 2000, pp. 313-316.
- [21] Sourrouille, J.L. and Caplat, G. Constraint Checking in UML Modeling, In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02), 2002, pp. 217-224.
- [22] Sprinkle, J., and Karsai, G." A Domain-Specific Visual Language for Domain Model Evolution", *Journal of Visual Languages and Computing*, Vol.15, Issues 3-4, June-August 2004, pp 291-307.
- [23] Wan Kadir, W.M.N., and Loucopoulus, P. "Relating evolving business rules to software design", *Journal of Systems Architecture*, 50(7), Elsevier, 2004, pp.367-382.