A Taxonomy of Computer-supported Critics

Norhayati Mohd.Ali,
John Hosking
University of Auckland,
New Zealand.

John Grundy
Swinburne University of Technology,
Australia

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

Outline

- Introduction
- Critic Taxonomy Development
- Critic Taxonomy–groups & elements
- Conclusion
- Q&A

Introduction

- Critics have emerged over the last several years as a specific tool feature to support users in computer-mediated tasks.
- The term ‘critic’ was initially introduced by Miller (1986)-> a software program that critiques human-generated solutions.
- Aim of this paper:
  - Present an initial critic taxonomy with the necessary groups and elements.
- A taxonomy is “a system for naming and organizing things…into groups which share similar qualities” (Cambridge dictionary).

Critic Taxonomy Development

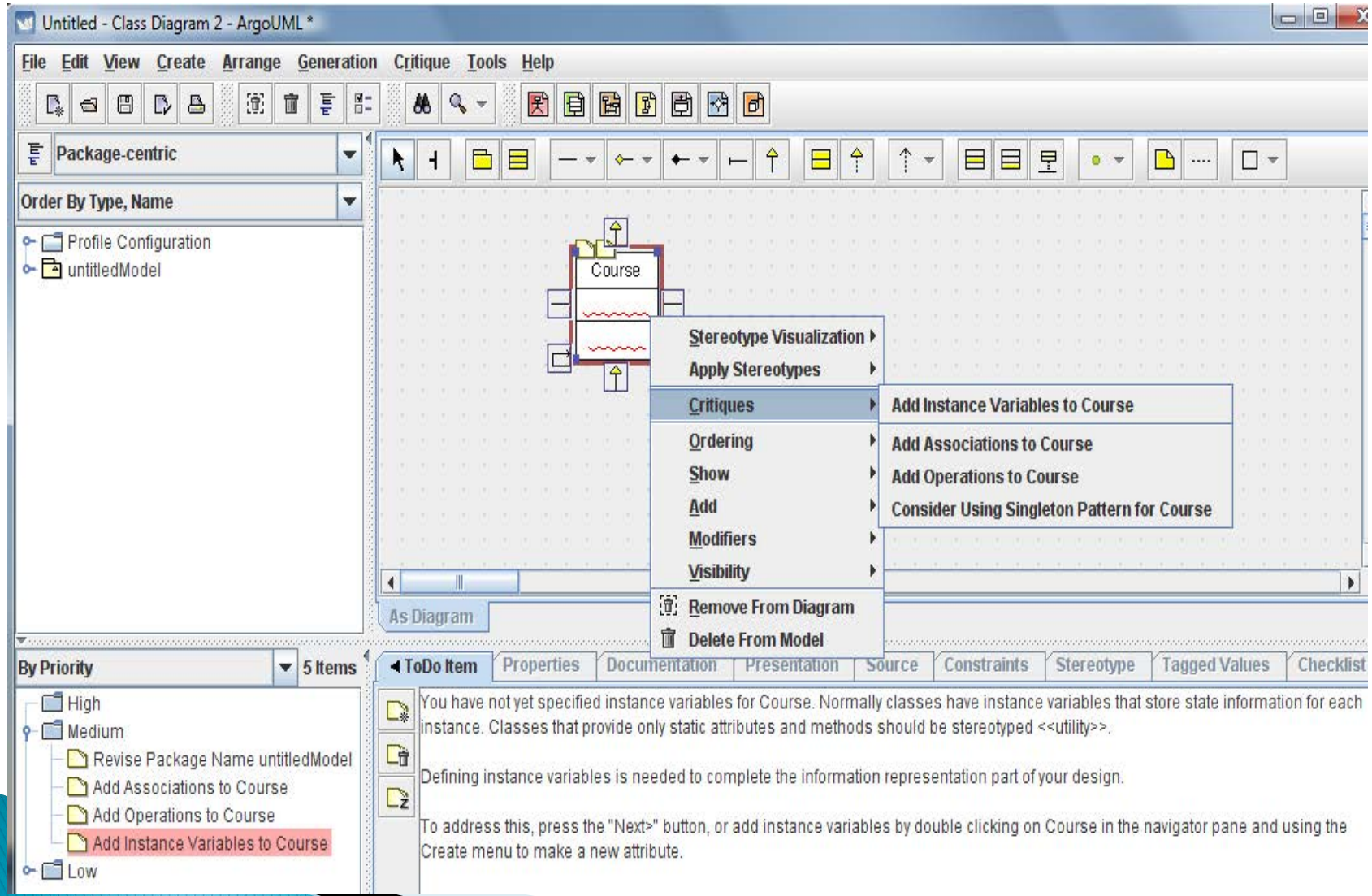
- Review related research concerning critics
- Discover different kinds of elements involve in critics
- Classify critic information into groups:
  - Critic domain
  - Critiquing approach
  - Critic dimension
  - Critic type
  - Modes of critic feedback
  - Types of critic feedback
  - Critic implementation approach
  - Critic rules authoring

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

Critic Taxonomy

A. Critic Domain						
B. Critiquing approach	**C. Modes of critic feedback**	**D. Critic rule authoring**	**E. Critic realization approach**	**F. Critic dimension**	**G. Types of critic feedback**	**H. Critic Types**
comparative	textual	Insert critic rule	Rule-based	Active	Explanation	Correctness
analytical	Graphical (visual)	Modify critic rule	Knowledge-based	Passive	Argumentation	Completeness
	3D visualization	Delete critic rule	Predicates	Reactive	Suggestion	Consistency
		Enable/ disable critic	Pattern-matching	Proactive	Examples	Optimization
		Critic rule authoring	OCL	Local	Simulation	Alternative
			Programming code	Global	Demonstration	Evolvability
					Interpretation	Presentation
					Positive feedback	tool
					Negative feedback	Experiential
					Constructive feedback	Organization
						Design pattern

Critic Example: ArgoUML tool



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

Critic Domain

- What domain (s) of discourse is the critic used in?
- Examples of domains- medical, education, software engineering, ect.
- Understanding the domain knowledge ->able to define and specify meaningful critics

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

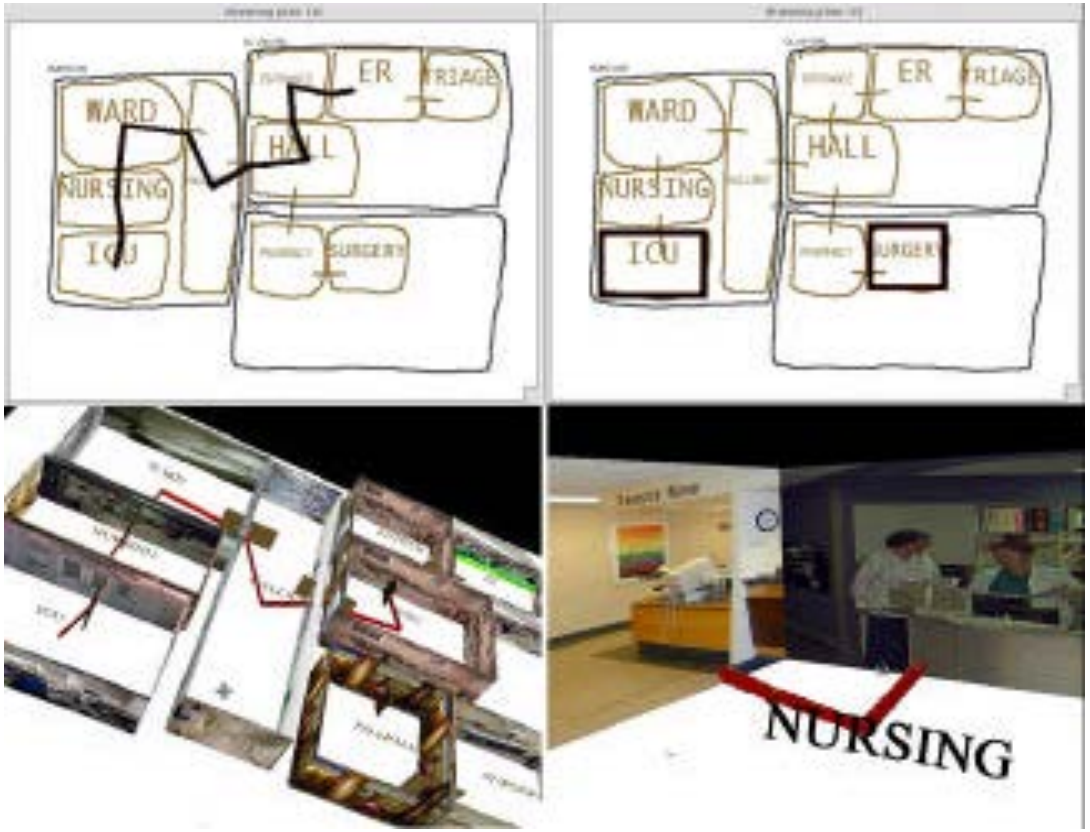
Critiquing Approach

- Does it compare or analyze target domain elements?
- A way to generate valid reasoning
- Comparative critiquing:
  - Complete and extensive domain knowledge
  - Comparison between user's and system's solution
  - Guides user to a known solution
- Analytical critiquing:
  - Incomplete domain knowledge
  - Analyses user-proposed solutions via set of rules
  - Guides user away from recognized problems

Modes of Critic feedback

- How does it provide end users with feedback?
- Presenting critic feedbacks (or feedbacks or critiques):
  - Textual messages
  - Graphic annotation
  - 3D visualizations

Feedback Example: Design Evaluator tool



Critique visualizations in architectural floor plan. Sketch annotation (top) and 3D model (bottom)

[source from Y. Oh, E.Y.-L. Do, and M.D. Gross, “Intelligent critiquing of design sketches”]

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

Critic Rule Authoring

- How are the rules embodied by the critic encoded?
- Critic rules are one of the building blocks in critics
- Critiquing capacity and issues may need to be adjusted in various situations
- Essential to allow users understand critic rules and able to modify or expand the rules
- Ability to enable/disable critic
- Facility to author own critics

Critic Realization Approach

- How is the critic built or realized in the target tool(s)?
- Various approaches can be used to implement critics:
  - Rule-based
  - Knowledge-based
  - Pattern-matching
  - OCL expressions
  - Predicate logic
  - Programming code

Critic Dimension

- Is the critic active, passive, reactive, proactive etc?

Critic Dimension	Brief Description
Active critics	Continuously critique a user's design or task.
Passive critics	Wait until a user asks for a critique
Reactive critics	Critique on the design or task that the user has done.
Proactive critics	Guide the user by presenting guidelines before the user make a decision.
Local critics	Critics that evaluate individual design elements.
Global critics	Critics that consider interactions between most or all of the elements in a design.

Types of Feedback

- Ways to present critic feedback
- Combination of styles in presenting critic feedbacks certainly facilitates users to clarify their understandings and improve their knowledge
- Common techniques:
  - Explanation
  - Suggestion
  - Argumentation

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

Types of Critic

- Does the critic check for completeness, correctness, consistency, alternatives, a mixture?

Critic Types	Brief Description
Correctness critics	identify syntactic and semantics flaws
Completeness critics	remind the designer to finalize design works
Consistency critics	show contradictions within the design
Optimization critics	advice better values for design parameters
Alternative critics	prompt the architect to consider options to a specified design decision
Evolvability critics	deal with issues such as modularization, that affect the effort needed to modify the design over time
Presentation critics	search for awkward use of notation that reduces readability
Tool critics	notify the designer of other accessible design tools at the times when those tools are useful
Experiential critics	offer reminders of previous experiences with similar designs or design elements
Organization critics	express the importance of other stakeholders in the development organization

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

Conclusions & Future Work

- We proposed and illustrated a critic taxonomy
- Utilities of our critic taxonomy:
  - Provide an overview of critic research
  - Identify & distinguish critic elements
  - Recognize techniques & methods applied in critics
- The taxonomy provide meaningful way to describe critics
- The critic taxonomy has guide our development on visual critic authoring tool
- Future work: improve our current critic tool and plan a larger evaluation with end users

Thank You…

- Ministry of Malaysia Higher Education & Universiti Putra Malaysia – scholarship support
- Postgraduate Research Student Support account, University of Auckland – financial support
- Software Process and Product Improvement (SPPI), University of Auckland– financial support
- ITSim 2010 for this opportunity to present

References

- A. Kleppe and J. Warmer, "The Semantics of the OCL Action Clause", in C. Tony & J.Warmer (Eds.), Object Modeling with the OCL, LNCS, vol.2263, Springer-Verlag Berlin Heidelberg, 2002, pp. 213-227.
- A. S. Gertner and B. L. Webber, TraumaTIQ:online decision support for trauma management, IEEE Intelligent Systems, pp. 32-39, 1998.
- B. G. Silverman, Survey of expert critiquing systems:practical and theoretical frontiers, Communications of the ACM, vol.35(4), pp. 106-127, April 1992.
- B. G. Silverman and T.M. Mehzer, Expert critics in engineering design: lessons learned and research needs, AI Magazine, vol.13(1), (1992) (AAAI).
- Cambridge Dictionary, http://dictionary.cambridge.org/
- C. R. B. Souza, J.S. Ferreira Jr, K.M. Goncalves, and J. Wainer, 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.
- E. Knauss, D. Lubke, and S. Meyer, Feedback-driven requirements engineering: the heuristic requirements assistant, In Proceedings of IEEE 31st International Conference on Software Engineering, 16-24 May 2009, pp. 587-590.
- E. Tyugu, Algorithms and architectures of artificial intelligence, Frontiers in AI and Applications, vol. 159, IOS Press, 2007.
- F. Bergenti and A. Poggi, 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.
- G. A. Bolcer, User interface design assistance for alrge-scale software development, Automated Software Engineering, vol.2(3), pp. 203-218, September 1995.
- G. Fischer, Human-computer interaction software: lessons learned, challenges ahead, IEEE Software, vol.6, pp.44 - 52, January 1989.

- G. Fischer, A. C. Lemke, and T. Mastaglio, Critics: an emerging apporach to knowledge-based human computer interaction, International Journal o Man-Machine Studies, 35, pp. 695-721, 1991.
- H. Irandoust, 2006. Critiquing systems for decision support. DRDC Valcartier TR 2003-321. http://pubs.drdc.gc.ca/PDFS/unc44/p524782.pdf.
- J. Bézivin and F. Jouault, Using ATL for checking models. Electronic Notes in Theoretical, Computer Science, 152, Elsevier, 2006, pp. 69-81.
- J. E. Robbins, 1998. Design critiquing systems, Technical Report UCI-98-41. http://www.ics.uci.edu/~jrobbins/papers/CritiquingSurvey.pdf
- J. E Robbins, and D. F. Redmiles, Software architecture critics in the Argo design environment. Knowledge-Based Systems 11(1), 1998, pp. 47-60.
- J. E Robbins, and D. F. Redmiles, Cognitive support, UML adherence, and XMI interchange in Argo/UML, Information and Software Technology, vol.42(2), pp. 79-89, January 2000.
- L. Qiu, and C. K. Riesbeck, "An incremental model for developing educational critiquing systems: experiences with the Java Critiquer", Journal of Interactive Learning Research, 2008(19), pp.119-145.
- M. C. Fu, C. C. Hayes, and E. W. East, SEDAR: Expert critiquing system for flat and low-slope roof design and review, Journal of Computing in Civil Engineering, vol.11(1), pp. 60 - 68, January 1997.
- N. M. Ali, J. Hosking, J. Huh, and J. Grundy, Template-based critic authoring for domain-specific visual language tools, In Proceedings of the 2009 IEEE Symposium on Visual Languages and Human-Centric Computing, Corvallis, Oregon, USA, pp. 111-118.
- P. L. Miller, Expert critiquing systems: practice-based medical consultation by computer. Springer Verlag, New York, 1986.
- W. M. K. Trochim, Outcome Pattern Matching and Program Theory, Journal of Evaluation and Program Planning, vol.12(4), pp. 355-366, January 1989.
- Y. Oh, E.Y.-L. Do, and M.D. Gross, "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.
- Y. Oh, M.D. Gross, and E.Y.-L. Do, Computer-aided critiquing systems, lessons learned and new research directions, In Proceedings of the 13th International Conference on Computer Aided Architectural Design Research in Asia, Chiang Mai (Thailand) 9-12 April 2008, pp.161-167.