A Visual Exploratory Notation for Object-based Multimedia

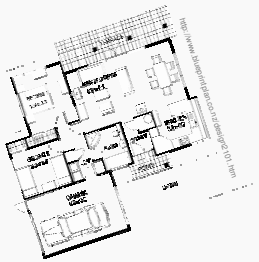
Mark Wilson – Unisys NZ & VUW
Mary Tate – Victoria University of Wellington (VUW)

Contents

Introduction

Imagine you are building a house…

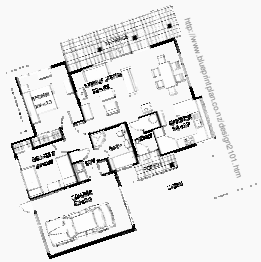
- You'll need some plans
- Your plans will need to represent the relevant information.
- This information needs to be represented in a way that is understandable to the relevant project stakeholders in a way that is relevant to what tasks they will be performing.



Introduction

Now imagine…

- You need to show a novice builder how to build your house in a certain way, without needing to demonstrate the whole thing.
- You will need represent series of spatio-temporal events that need to take place during the house construction.
- To avoid cognitive overload, your plans will need to represent only the relevant information – they will be an abstraction of the intended reality.
- The plans are an abstraction, but be careful – they still need to have a 'good fit' in the minds of your audience with that which you are trying to represent.
- The plans must also be quicker to and more effective than demonstrating step-by-step… otherwise what would be the point?



Introduction

Teaching reusable multimedia.

- Those teaching Multimedia face similar issues to what you would face in the house example.
- Effective Multimedia should be object based:
  - So that it is easy to update as requirements change
  - To reduce the development time by being able to reuse parts of a multimedia artefact
- Teaching this is more complicated than simply teaching students how to use multimedia development software (such as Macromedia Flash).

5

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui | VENOM | UNISYS Imagine it. Done.

Introduction

Teaching reusable multimedia.

- Teaching reusable multimedia can be complicated because:
  - Multimedia development is a creative process
  - There are potentially hundreds of user-interface steps involved in even a short animation.
  - Every project has slightly different requirements, for instance, different animations, etc…
  - Different situations require different object hierarchies, and therefore, case-by-case analysis is required.
  - Reusability is a moving target; depending on the situation some development approaches yield more reusable results than others.
- So a step-by-step approach is not the best.

6

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui | VENOM | UNISYS Imagine it. Done.

Motivation

An action research situation…

- This project was born out of the experiences gained by one researcher while employed part-time at Victoria University of Wellington to tutor Macromedia Flash to undergraduate students.
- These were students studying introductory level Electronic Commerce and Multimedia. Many of the students were having significant difficulty performing workshop tasks where animation was required within an object hierarchy (for example, an animated roll-over button).
- Students would often seem to lose their way within the object hierarchy of their Flash files, and end up editing the wrong symbols/objects by mistake (another instance of the 'lost in hyperspace' dilemma perhaps).
- Initial attempts to address these issues involved ad-hoc whiteboard diagrams combined with oral explanations.

7

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui | VENOM | UNISYS Imagine it. Done.

Motivation

A typical tutor-student discussion.

- Novice users getting lost in their own Flash animations may be due to their having an incomplete mental model of how to use Flash in an object-orientated way.
- Below is a typical tutor-student discussion:

S - "Why doesn't my button have those things?"
T - "Because that isn't a button?"
S - "I made a button"
T - "Yes, but now you are inside the movie clip that is inside the over-state of the button you made"
S - "How can you tell?
T - "You can tell by looking here. See? Scene1 > Button > Symbol 2. They are like breadcrumbs"
S - "What are breadcrumbs?"
T - "Those things on some web pages that... um… they tell you how you got to the page you are looking at."
S - "Ok. How did I get inside the movie clip?
T - "You probably double clicked on it"

8

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui | VENOM | UNISYS Imagine it. Done.

Motivation

Identify and improve successful aspects.

- It was in response to this type of student confusion that the tutor would attempt to explain using the whiteboard.
- This approach was helpful for many of the students. Students were observed looking back at the whiteboard while they worked to help them when they got stuck with a task.
- Clearly, some qualities of this approach were assisting students' learning. Since our aim was to scaffold student learning, we decided to try to identify and improve on the aspects of our teaching that seemed most effective.

Literature Review

Overview.

- The literature review can be divided into four main parts:
  - Defining the main issues in the situation.
  - Looking at how visual notations work to try and understand the reasons why the diagramming approach was working.
  - Looking at educational theory to try to understand the learning process that taking place.
  - Looking for a potential existing notation (or quick to customise) solution.
- Key aspects of the above are discussed over the following slides.

Literature Review

Defining the situation.

- There was no literature found specifically about difficulties involved in teaching object-based multimedia.
- But other people teaching object orientation had experienced similar issues, and attempted to solve them in similar ways.
- A series of articles published in 1999 in the Journal of Object Orientated Programming (JOOP), written by Michael Kolling was particularly useful.
- Kolling based his writing on his experience as a lecturer in Computer Science at Monash University, Melbourne.

Literature Review

Defining the situation.

- Over a series of four articles in 1999, Kolling develops on the following observations:
  - Teaching OO to people who are used to procedural methods of development is difficult.
  - There is a lack of suitable tools available to teach OO.
  - The development environment – available text-based interfaces do not accurately represent the structure of classes and objects. This lack of congruence can cause problems for novice users.
- These observations motivate Kolling's development of 'The Blue Environment'.
- The Blue Environment is an object-orientated programming environment that uses visualization to encourage students to think in terms of objects and classes while programming.

Literature Review

Defining the situation.

- Kolling's observations were relevant to our action research situation in the following ways:
  - Animation is procedural by nature.
  - Macromedia Flash is both OO and Procedural, as it has both a timeline and a reusable object hierarchy.
  - Visualisation, in the form of an interactive visual notation, was found to be helpful for to help Kolling's students understand OO.

13

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui | VENOM | UNISYS Imagine it. Done.

Literature Review

Understanding visual notations.

- Diagrams are a form of external graphical representation.
- They are used extensively in education, across a wide range of different subject areas.
- According to previous cognitive science research, there are a number of ways in which diagrams may be cognitively beneficial:
  - In general, diagrams support cognitive processes by serving as an "external aid to thought" (Addis, 1997 in Price, 2002).
  - More specifically, the actual form that the diagram takes (in order to represent processes and structures) can help people to conceptualize these things.
  - The capacity that diagrams have to represent information, even when very complex.
- Price (2002), drawing on previous research, identified the following advantages:
  - Simultaneous presentation of information (Larkin & Simon, 1987)
  - Perceptual availability of information (Larkin & Simon, 1987)
  - Reduction in memory load (Bauer & Johnson Laird, 1993)
  - Inference constraining (Zhang & Norman, 1994)

14

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui | VENOM | UNISYS Imagine it. Done.

Literature Review

Educational theory.

- It was important to look at the pedagogy behind what the notation was being used to achieve.
- Scaffolding is a key aspect of this.
- Venom works by scaffolding the development of a mental model that enables users to use Macromedia Flash in a certain way.
- The basic idea behind scaffolding is that:
  - People have their area of core knowledge and a surrounding zone of proximal development (Lewis, 2002).
  - When a person's core knowledge overlaps with another persons zone of proximal development, the first person can provide "scaffolding" for the second person; in effect, facilitating that person's learning (Lewis, 2002).
  - In a classroom situation, teachers are expected to perform the role of scaffolding student's understanding.
- When individuals use computer software it is helpful if the interface scaffolds the users understanding in a way that helps them to use the software to perform the required task/s.
- This is not always the case, as people can have difficulty fully understanding computer interfaces.
- The Venom methodology is intended to scaffold an understanding of using Macromedia Flash in an Object-based way.

15

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui | VENOM | UNISYS Imagine it. Done.

Literature Review

Assessing alternatives.

- Once a better understanding of how visual notations work was reached, it became apparent that a purpose-built notation would need to be developed.
- Even so, a number of pre-existing visual notations were considered and found unsuitable for this particular set of requirements, for example:
  - The Blue Environment (Kolling, 1999a), a tool used to teach OO programming to undergraduate programming students.
  - UML
  - STRPN, a Petri-net based notation.
- These notations (among others) were found unsuitable because:
  - an effective visual notation needs to have a good fit with its application domain, in this case: Teaching novices how to use Macromedia Flash in an object-based way.
  - this particular notation's purpose was to aid learning and analysis:
    - It needed to be faster to learn than Flash itself
    - It needed to constrain inferences in such a way as to assist OO analysis
    - It needed to facilitate creativity and allow the user to explore alternatives.

16

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui | VENOM | UNISYS Imagine it. Done.

Method

Overview.

- The notation needed to be developed and then tested.
- The development of the method for doing this required its own form of literature search;
  - in order to provide the theoretical foundation required to answer all of the 'how to' questions.
- The method was divided into the following main sections:
  - How to develop the visual notation to meet its requirements
  - How to teach the notation to participants prior to testing
  - How best for the participants to experience the notation
  - How best to test the notation

17

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

Method

Developing Venom.

- Venom was initially created with the following main inputs/influences:
  - Visual Language Theory
    - (Narayanan & Hubscher, 1997).
  - Unified Modeling Language (UML)
  - The existing symbolism within the Flash interface
  - Popular metaphors in HCI Theory
    - (Benyon & Imaz 1999).

18

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

Method

Developing Venom.

- The following two slides contain selected component examples of Venom.
- The third slide shows an example of a simple animation diagrammed using Venom.
- The full current version of Venom is outlined in the Appendix of this presentation.
- The version in the Appendix includes the changes made as a result of the evaluation session.

19

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

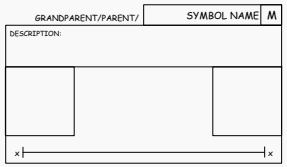
VENOM

UNISYS Imagine it. Done.

Method

Venom Components: The Basic Container.

- The notation is based around a container metaphor. Each type of symbol that you can create in Flash has its own container in the notation. You can think of the container as a filing card on which you store all relevant information about the symbol being represented. Displayed on the right is the movie container.
- The "grandparent/parent/" part is optional depending on how specific you want to be. It is for showing were in the hierarchy of symbols this symbol goes. Think of it like the path to a folder in Windows Explorer or DOS. You can use lines to show relationships instead of this path.



20

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

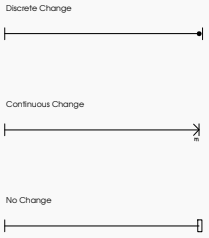
UNISYS Imagine it. Done.

Method

Venom Components: Timelines.

- In Venom, timelines go from left to right, the same as they do in Flash.
- You can put markers with frame numbers on your timeline, or use seconds, as preferred. Movie and Graphics symbols can both have timelines.
- With Venom, timelines can be used to represent different types of change:

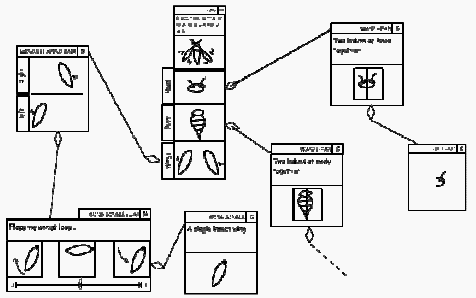
1. Discrete change
2. Continuous change (tweening)
3. No change (or no change at a particular level of the symbol hierarchy)



Method

Venom Example: A Wasp.

- To the right, is one way of structuring an animation of a wasp flapping its wings.
- A movie like this one could be set to follow a motion guide around a circuit.
- It is displayed and diagramed here in one spot, so it doesn't fly off the screen.
- This is a simple version; an example of a more complex version would be if the wings were only set to flap when the insect was flying.



Method

Refining Venom – CDs Questionnaire.

- This study uses the Cognitive Dimensions (CDs) of Notations framework to help evaluate Venom for the purpose of refining it further.
  - According to Green and Blackwell, "Cognitive dimensions are tools to aid non-HCI specialists in evaluating the usability of information-based artefacts" (Green & Blackwell, 1998)
  - Version 5.1 of *A Cognitive Dimensions of Notations Questionnaire* (A. Blackwell & T. Green, 2000) was used during an evaluation session.
- The study also uses expert evaluation in a focus group session.

Method

Refining Venom – Focus group.

- According to Nielson there are a range of benefits associated with expert and heuristic evaluation:
  - Quick, cheap feedback is provided to designers (Nielson & Mack, 1994)
  - Works with as little as two or three evaluators (Nielson, c.1996)
  - The results generate good ideas for how to improve the user interface (Nielson & Mack, 1994)
  - Useful for testing early prototypes before actual users are brought in. Such prototypes do not need to be an interactive system, for instance, they could be pencil paper and string prototypes. (Nielson, c.1995)
  - The researcher can communicate with the evaluator to answer questions about usage. This is especially useful when the system is domain specific and the evaluator lacks and in-depth knowledge of the domain. (Nielson, c.1995)
- The CDs questionnaire was given individually, before the focus group first, and the questions used as the heuristics for the expert evaluation focus group.

Method

The evaluation session.

- Both the CDs questionnaire and the focus group were conducted at during a four hour long (excluding breaks) evaluation session:
- An the outline of the evaluation session was as follows:
  - The purpose of the visual notation explained in detail
  - A short revision lesson was held. This explained how and why to use symbols in Flash, and included some activities
  - A lesson was held explaining how to use the visual notation method
  - Participants were asked to plan a selection of typical Flash animations using the notation.
  - Participants were asked to create one or more of the animations based on their plans.
  - Participants completed a slightly condensed version of version 5.1 of *A Cognitive Dimensions of Notations questionnaire* evaluating the effectiveness of the notation.
  - Participants discussed the notation in a focus group setting
- It was intended that the Flash revision lesson would take about 30 minutes. On the day it took about an hour an a half to get started and complete the revision lesson.

Method

The evaluation session.

- The evaluation session was 4 hours long, excluding breaks.
- The group of six participants included experts in:
  - Teaching Multimedia
  - Multimedia development methodologies
  - Multimedia development using Macromedia Flash
  - Modeling languages
  - Visual design
- All participants had an understanding of UML, some had an expert understanding
- At the beginning of the evaluation session participants were informed of the session's purpose; to obtain a heuristic evaluation of the notation by a group of experts. Their role as subject matter experts was explained.
- Participants were assured that as the purpose of the session was evaluate the effectiveness of the notation, not technical ability in Flash, and help on how to use Macromedia Flash would be provided as required.

Method

Teaching Venom.

- Upon further consideration, it became apparent that we would have to test prerequisite knowledge prior to teaching Venom at the evaluation session.
- A Flash basics lesson was developed in order to check that evaluation session participants would all have basic level of understanding of symbols in Flash (symbols are Flash's reusable objects).
- With this material in place the participants could proceed to a lesson designed to teach Venom itself.

Method

More about the CDs questionnaire.

- The questionnaire contains questions that relate to various cognitive dimensions:
  - Diffuseness & Viscosity (very similar to each other in a non-interactive, i.e. paper based, notation)
  - Hard mental operations
  - Error proneness
  - Closeness of mapping
  - Role expressiveness
  - Provisionality
  - Consistency
  - Novelty
  - Improvements
- From the answers to these questions the notation can be evaluated along the different dimensions.
- This proves a type of cognitive profile of the notation.
- Different types of activities are best supported by different profiles – the dimension themselves are not intrinsically good or bad (Green & Blackwell, 1998).
- These profiles are as follows:
  - Incrimination
  - Transcription
  - Modification
  - Exploration
  - Search

Method

More about the CDs questionnaire.

- After reflecting on the profiles, it was decided that Venom was an exploratory notation.
- The table below outlines the desirable profile for this type of user activity:

CD	Effects on Exploratory Notation
viscosity	harmful
hidden dependencies	acceptable for small tasks
premature commitment	harmful
abstraction hunger	harmful
secondary notation	very harmful
visibility/juxtaposability	important

Results

Overview.

- Overall, the results of the first evaluation session provided:
  - Information about the CDs and effectiveness of Venom
  - A list of improvements that can be made to Venom
  - Information about how better to teach Venom
- These findings are summarised over the following slides.

Results

The CDs questionnaire.

- The table below summarizes the responses to the cognitive dimensions of notations questionnaire filled out by the participants about their experiences with Venom:

CD	Evaluation
Visibility	medium/high – good visibility; it is all on one page
Diffuseness	medium/low – not too long-winded
Hard mental operations	medium – some comments to do with Venom but about half were a reflection of hard mental operations in the application domain
Error proneness	medium – some people found that they made errors
Closeness of mapping	medium/high – quite closely mapped to the application domain
Role expressiveness	medium/high – quite expressive
Provisionality	high – enables exploration
Consistency	high – very constant
Novelty	low – not many novel uses described
Improvements	medium – some improvements suggested

Results

The CDs questionnaire – Discussion.

- Venom performed well in the Cognitive Dimensions of: visibility, diffuseness, closeness of mapping, role-expressiveness, provisionality and consistency.
- These results are very encouraging.
- Of particular importance is the good result for provisionality. This indicates that Venom would likely be an effective exploratory notation.
- The closeness of mapping and the role-expressiveness indicate that Venom would potentially make a good tool for scaffolding novice developer understanding of object-based Flash, and quite likely the Flash application itself.
- Poorer results for error-proneness and hard mental operations reinforce some of the comments made by participants (during the focus group) about how to improve Venom.

Results

The focus group.

- This findings of the expert evaluation focus group can be divided into three main areas:
  - The learnability of the notation
  - The efficiency of the notation
  - Fit of the notation to the application domain
- The most emphasized issue was the learnability.

33

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

Results

The focus group.

- Participants of the focus group made many suggestions on how to improve the methods and materials used to teach Venom:
  - The use of examples to scaffold novice developers understanding of object-based Flash and how to use Venom. This would include examples of Flash source files accompanied by Venom diagrams, and exercises like constructing a Venom diagram from a completed Flash source file.
  - Step-by-step instructions on how to use Venom
  - A process for analysing a potential animation in order to determine what symbols can be made reusable and diagrammed as such in Venom. It is likely that scaffolded examples would be useful for teaching this process.
  - Visual aides and an instructor demonstration to compliment the existing self-paced Venom learning resource.

34

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

Results

The focus group.

- It is likely that the incorporation of the suggestions on the previous slide would significantly improve the ability of Venom to be tested in an experimental design.
- In addition to these improvements, potential participants in an experiential design would need to be pre-screened for an understanding of Macromedia Flash basics.
- It would be asking too much of participants in an experiment to learn Flash basics as well as Venom within the duration of a single experimental session.

35

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

Results

Evaluation session overall.

- Based on the results of the evaluation session overall, the following main improvements to Venom are required:
  - A better method for representing layers. This would need to include some way of making layers more distinguishable from other areas inside the main container object for each symbol is required.
  - The inclusion of an 'A' container to enable Venom to be used to help plan the reuse of code as well as animations and graphics.
  - Improvements to the description area of the main symbol containers, including separate areas for object instance properties, methods, textual descriptions and visual descriptions.
  - Better guidelines for visual descriptions (the thumbnail sketches), such as an explanation of how to represent animations such as rotation.

36

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

Future Research

Overview.

- A second focus group has been conducted, this time with Flash development experts from a successful Wellington web-design and multimedia company.
- The Venom methodology will be revised based on this feedback
- At this stage, the next phase looks to be an experimental design with before and after control group confirmation.

37

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

References (slide 1 of 3).

- Benyon, D. and Imaz, M. (1999). Metaphors and Models: Conceptual Foundations of Representations for Interactive Systems Design. *Human-Computer Interaction*, Vol 14, pp 159-189.
- Biddle, R., Noble, J., & Tempero, E. (2001). *Use Case Cards and Roleplay for Object Oriented Development* (Technical Report). Wellington: School of Mathematical and Computing Sciences, Victoria University of Wellington.
- Blackwell, A., & Green, T. (2000). *A Cognitive Dimensions Questionnaire*, 2003, from www.cl.cam.ac.uk/~afb21/CognitiveDimensions/CDquestionnaire.pdf
- Blackwell, A. F., Britton, C., Cox, A., Green, T. R. G., Gurr, C., Kadoda, G., et al. (2001). Cognitive Dimensions of Notations: Design Tools for Cognitive Technology. In M. Beynon, C. L. Nehaniv & K. Dautenhahn (Eds.), *Cognitive Technology 2001* (pp. 325-342): Springer-Verlag.
- Blackwell, A. F., & Green, T. R. G. (2000). *A Cognitive Dimensions Questionnaire Optimised for Users*. Paper presented at the 12th Workshop of the Psychology of Programming Interest Group, Cozenza, Italy.
- Costello, W. (2001). *Computer-Based Simulations as Learning Tools: Changing Student Mental Models of Real-World Dynamical Systems*, 2003, from www.clexchange.org/ftp/documents/system-ed/ SE2001-04ChangingStuMentMod.pdf
- Creswell, J. (1994). *Research Design: Qualitative and quantitative approaches*. Thousand Oaks, CA: Sage.
- Goguen, J. A. (1996, 2002). *Semiotic Morphisms*. Retrieved October 11, 2003, from http://www.cs.ucsd.edu/users/goguen/papers/sm/smm.html
- Green, T., & Blackwell, A. (1998). *Cognitive Dimensions of Information Artefacts: a tutorial*, 2003, from www.cl.cam.ac.uk/~afb21/CognitiveDimensions/ CDtutorial.pdf

38

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

References (slide 2 of 3).

- Green, T. R. G. (1996, November). *An Introduction to the Cognitive Dimensions Framework*. Paper presented at the MIRA Workshop, Monselice, Italy.
- Hsu, P.-Y., Chang, Y.-B., & Chen, Y.-L. (2003). STRPN: A Petri-Net Approach for Modelling Spatial-Temporal Relations between Moving Multimedia Objects. *IEEE Transactions on Software Engineering, 29*(1), 64-76.
- Johnson-Laird, P. N. (1980). Mental Models in Cognitive Science. *Cognitive Science, 1*(4), 71-115.
- Kolling, M. (1999a). The Problem of Teaching Object-Oriented Programming, Part 1: Languages. *Journal of Object Orientated Programming, January*, 8-16.
- Kolling, M. (1999b). The Problem of Teaching Object-Oriented Programming, Part 2: Environments. *Journal of Object Orientated Programming, February*, 6-13.
- Kolling, M. (1999c). Teaching Object Orientation with the Blue Environment. *Journal of Object Orientated Programming, May*, 14-23.
- Lewis, R. (2002). Learning Communities - old and new. International Conference on Computers in Education, Auckland, New Zealand.
- Liu, Y., & Ginther, D. (2003). *Cognitive Styles and Distance Education*, 2003, from www.westga.edu/~distance/liu23.html
- Najjar, L. J. (1995). *Does Multimedia Information Help People Learn?* (No. GIT-GVU-95-28). Atlanta: Georgia Institute of Technology.
- Narayanan, N. H., & Hubscher, R. (1997). Visual Language Theory: Towards a Human-Computer Interaction Perspective. In B. Meyer & K. Marriott (Eds.), *Visual Language Theory* (Vol. 2002).

39

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

References (slide 3 of 3).

- Nielson, J. (c.1995). *Heuristic evaluation*. Retrieved October 11, 2003, from http://www.usabilitynet.org/tools/expertheuristic.htm
- Nielson, J. (c.1996). *How to Conduct a Heuristic Evaluation*. Retrieved October 11, 2003, from http://www.useit.com/papers/heuristic/heuristic_evaluation.html
- Nielson, J., & Mack, R. L. (1994). *Usability Inspection Methods*: John Wiley & Sons, Inc.
- Phillips, R. (1997). *The Developer's Handbook to Interactive Multimedia: A Practical Guide for Educational Applications*: Kogan Page Limited.
- Post, E. (2000). *Jade for Developers* (3rd ed.). New Zealand: Publishing Press.
- Price, S. J. (2002, 2002). *Diagram Representation: The Cognitive Basis for Understanding Animation in Education*. Retrieved May 5, 2003, from http://www.educationau.edu.au/archives/cp/REFS/reeves_paradigms.htm
- Reeves, T. (1996). A Hopefully Humble Paradigm Review. In itforum@uga.cc.uga.edu (Ed.): Instructional Technology Forum.
- Rieman, J. (1999, 15-20 May). *Testing and Revising JSketch: A Drawing Tool for Informal Graphics*. Paper presented at the ACM Conference on Human Factors in Computing Systems, Pittsburgh USA.
- Rosson, M. B., & Carroll, J. M. (1996). Scaffolded examples for learning object-orientated design. *Association for Computing Machinery. Communications of the ACM, 39*(4), 46-49.
- Thomas, J. C., Lee, A., & Danis, C. (2002). Enhancing Creative Design via Software Tools. *Communications of the ACM, 45*(10), 112-120.
- Tidwell, J. (1999). *Common Ground: A Pattern Language for Human-Computer Interface Design*. Retrieved May 5, 2003, from www.mit.edu/~jtidwell/interaction_patterns.html
- Weiten, W. (1995). *Psychology: Themes and Variations* (3rd ed.). California: Cole Publishing Company.

40

Victoria UNIVERSITY OF WELLINGTON Te Whare Wānanga o te Ūpoko o te Ika a Māui

VENOM

UNISYS Imagine it. Done.

Appendix:
Using VENOM 2.0

Learning Objectives

- By the end of this lesson, participants should be able to:
  - Demonstrate an ability to use Venom, by diagramming a simple animation.
  - Demonstrate an understanding of how Venom translates into Macromedia Flash, by creating the previously diagramed animation in Flash.
- This learning object is designed with two purposes in mind:
  1. As a lesson to help people to become familiar with using Venom.
  2. As a memory jogger for those using the notation.

Assumed Prerequisites

This lesson assumes that you have experience with the following:

- The Flash User Interface
  - The basic metaphor
  - The main panels
- Using Flash
  - Creating lines, shapes and fills
  - Editing lines, shapes and fills
  - Creating and editing symbols

Concept Check

Discuss Your Symbol Usage

1. How do you use Symbols in Flash?
2. Do you use any planning techniques?
3. Ever used similar diagramming notations before, such as UML?

Lesson Overview

1. If viewed in order, these lesson slides begin by explain a bit more about Venom
2. Then each part of the Venom diagramming system is detailed in turn.
3. This is followed by an example of an animation diagrammed using the notation.
4. The lesson slides conclude with some suggested exercises for learning the notation.

About VENOM

Introduction

- VENOM is short for Visual Exploratory Notation for Object-based Multimedia.
- Venom is a non-interactive exploratory visual notation for planning object-based multimedia prior to commencing development.
- It is designed with two audience-purposes in mind:
  - To help those new to object-based multimedia applications, such as Macromedia Flash.
  - To help more experienced multimedia developers by servings as a planning and communication tool.
- The idea for Venom arose out of tutoring 200-level eCommerce and Multimedia students on how to use Flash
- This version of the notation, Venom 2.0 is designed for use with Macromedia Flash.

About VENOM

The Purpose

- Applications like Flash can be hard for people learn, as they require people to think in both a procedural and object-oriented ways at the same time.
  - For instance, the Flash user-interface has a timeline on which symbols (objects) can be positioned (the timeline is procedural). These symbols are stored in a library, and can be re-used (this is object-orientated).
- As you know, updating a symbol in the Flash library will update all instances of that symbol, regardless of when and where it appears in the Flash movie. This can be a very powerful tool. Without it, the user would have to laboriously update their movie by copying and pasting every time a change is required. Each of the symbols in the Flash library has its own timeline, which in turn can hold more symbols, and so on. The structure of these symbols can quickly become complicated.
- The purpose of Venom is to help with this situation in three ways:
  - First, Venom aims to help people understand the hierarchical structure of symbols.
  - Second, Venom aims to help people to understand how this hierarchy of objects relates to the timeline/s.
  - Third, Venom aims to increase the awareness of the need for an object-based approach to multimedia, by helping people to explore possibilities that they may not have been aware of before.

About VENOM

What's in a name?

As mentioned in the introduction, Venom is a non-interactive exploratory visual notation for planning object-based multimedia prior to commencing development. This slide seeks to clarify some of these terms:

- **Non-Interactive:**
  - Some notations are interactive, Venom is not. It is designed to be drawn on paper in pencil. Examples of interactive notations are Rational Rose, used to create UML, or other any form of graphical user interface.
- **Exploratory:**
  - Venom is intended to help novice developers explore how they will construct a piece of interactive multimedia (or an animation or graphic). It is not intended as a tool for detailed system specifications. It is tool for helping people think about where, when, and how they will position re-useable symbols in a timeline. In this way, it is designed to help the cognitive system (the human). It is not a tool for communicating with a computational system; the Flash user interface already does this.
- **Object-based:**
  - Flash is not completely object-orientated. You can create reusable symbols from one of Flash's three pre-defined classes (movie, button, and graphic) but you can not readily create your own classes. Each symbol has its own properties, and can have actions associated with it (like a method). Within a Flash movie, timelines can be thought of as a collection (or aggregation) of objects that are played through in a default sequence. This default sequence can be controlled, to a certain degree, by the use of actions.
- **Visual Notation:**
  - A well known example of a visual notion is sheet music. This uses different graphical symbols to represent different sounds. A visual notation is distinct from a textual notation, which would use alpha-numeric characters. Venom uses graphical symbols to comprise a diagram, but also uses some alpha-numeric characters.

VENOM Components

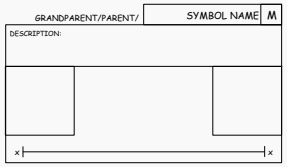
Introduction

- The following section outlines the different components of Venom.
- It may also help you to think of it as a type of toolkit to use when creating a diagram.

VENOM Components

The Basic Container

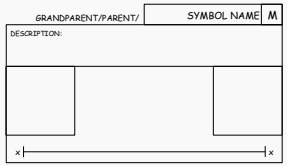
- The notation is based around a container metaphor. Each type of symbol that you can create in Flash has its own container in the notation. You can think of the container as a filing card on which you store all relevant information about the symbol being represented. Displayed on the right is the movie container.
- The "grandparent/parent/" part is optional depending on how specific you want to be. It is for showing were in the hierarchy of symbols this symbol goes. Think of it like the path to a folder in Windows Explorer or DOS. You can use lines to show relationships instead of this path.

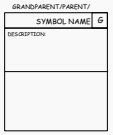


VENOM Components

Container Types

- Movie symbols are distinguished by an "M" in a box at the top right of the container.
- Graphic symbols use a "G".
- Button symbols use a "B".
- Movie symbols containing important reusable ActionScript can be represented using an "A"
- Movies and graphics can have timelines (shown top right) or be single frames (shown bottom right).

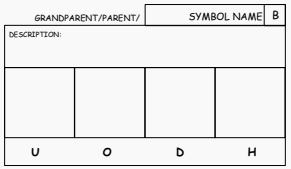




VENOM Components

The Button Container

- Button symbols have four boxes for the four difference button states in Flash: Up. Over, Down, and Hit (active area).
- These states are represented by the four boxes within the button container illustrated here.
- Each button state can have other symbols aggregated into it (see the relationships slide, later in these slides, for more info on this).

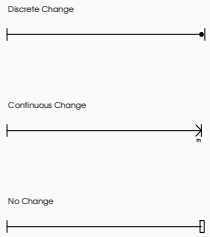


VENOM Components

Timelines

- In Venom, timelines go from left to right, the same as they do in Flash. You can put markers with frame numbers on your timeline, or use seconds, as preferred. Movie and Graphics symbols can both have timelines. With Venom, timelines can be used to represent different types of change:

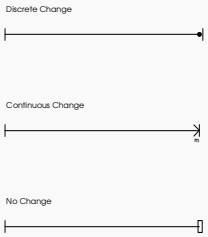
  1. Discrete change
  2. Continuous change (tweening)
  3. No change (or no change at a particular level of the symbol hierarchy)



VENOM Components

More on Timelines and Changes

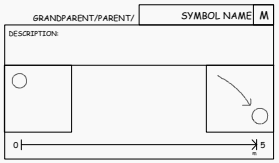
- Discrete change, is when the image on screen is swapped, but with no transitional effect. Continuous change (called tweening in flash) is when a change occurs gradually over time. If you are representing a tween with Venom, it is a good idea to write 's' or 'm' above the tween arrow to indicate whether it is a shape or a motion tween. For instance, the little 'm' underneath the continuous change arrow (middle on the left) head indicates that the change is a motion tween.
- No change, could mean that there is no change, or that there is no change at a particular level of the symbol hierarchy. This is because often symbols are put on a timeline but no change is actually taking place in that timeline, just inside the symbols on the timeline. For instance, a button may just sit there doing nothing until the user rolls over it - the rollover effect would be inside a move symbol, which in turn would be inside the Over frame of the button.



VENOM Components

Thumbnail Sketches

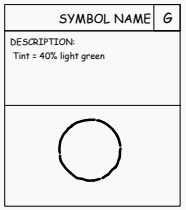
- The boxes within the main container are where you can make a thumbnail sketch of what goes in a key frame. You can use as many of these as you require (but you may need to draw your container bigger if you do).
- These thumbnails should be as quick and simple as possible; try and focus on the most relevant details. You would only need to make these more detailed if you were using Venom in a collaborative environment.



VENOM Components

Description Area Types

- All of the symbol types can have one or many description areas. This is where you can write notes to yourself. You would use a description area to indicate which properties are changing or if an object has actions.
- Only include relevant properties in the description area. A relevant property is one that is either changing over time, or in a particular instance. For example, if you were changing width and height properties over time, you might write: W=100% H=20% above the thumbnail box on the left and W=100% H=100% above the thumbnail box on the right.
- If the color of an instance was going to be different from the library in a significant way, you could write: Tint = 40% Green above the thumbnail box. If you were tweening the tine effect, you would include start and finish values for the tween.
- In some cases may be helpful to include extra graphical notations in the description area, or in/around the thumbnail box. Please see the extra notations slide for more information on this.
- Use a different description area for symbol instance properties, actionscript methods, textual description of the symbol, etc.



VENOM Components

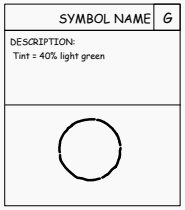
Relationships

- In Flash the main relationships you will need to represent are Parent/Child relationships. For this you use a line with a diamond on the end closest to the parent. The parent is the symbol with the timeline that contains the other symbol.
- The main Flash timeline is called the root level or root timeline (like the root of a tree). You can represent this with a symbol called "_root" if you wish. However, you do not need to start all of your Venom diagrams at the root timeline, or have relationships pointing back to the root. It is often helpful to just sketch a fragment that you are having trouble with.
- If you have multiple instances of one symbol you can write the number of instances on the line. Write this number at the end of the relationship line that is closest to the child symbol.
- Tip: If you want to keep your diagram clear, it is better to used curved lines than overlapping ones.

VENOM Components

Description Area Types

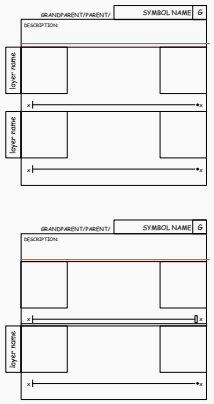
- All of the symbol types can have one or many description areas. This is where you can write notes to yourself. You would use a description area to indicate which properties are changing or if an object has actions.
- Only include relevant properties in the description area. A relevant property is one that is either changing over time, or in a particular instance. For example, if you were changing width and height properties over time, you might write: W=100% H=20% above the thumbnail box on the left and W=100% H=100% above the thumbnail box on the right.
- If the color of an instance was going to be different from the library in a significant way, you could write: Tint = 40% Green above the thumbnail box. If you were tweening the tine effect, you would include start and finish values for the tween.
- In some cases may be helpful to include extra graphical notations in the description area, or in/around the thumbnail box. Please see the extra notations slide for more information on this.
- Use a different description area for symbol instance properties, actionscript methods, textual description of the symbol, etc.



VENOM Components

Layers

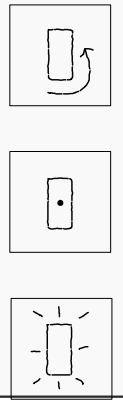
- In Venom it is a good idea to represent each symbol on a different layer, even if you don't intend to put them on different layers. You can divide layers from one another by using a horizontal line.
- If you need to name your layers, a box running up either side of the layer can be used
- It is sometimes useful to have an overall set of thumbnails for the parent as well as some having layers for the individual symbols as well. This is like a visual description of the symbol. When you do this, you should separate the parent (or aggregate) thumbnail with a double horizontal line.
- You can put thumbnails on the individual layers if you need to, or rely on the thumbnails in the child symbols instead, or both. In cases where there is a change in the actual parent symbol (not just a change carried through from a child symbol) you should probably use both.
- Layers should be separated from the description areas with a double ruled line.



VENOM Components

Additional Notations

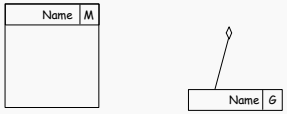
- In some cases it may help to include extra notations in a description area, or in and around a thumbnail box. Some useful examples of extra notations are:
  - Lightly sketched boxes around a part of a thumbnail to help indicate an orientation change (like a rotation).
  - Short arrows indicating the direction of a rotation.
  - A small dot for the reference point of a symbol, or for the centre point of a rotation.
  - Short radiating lines around a shape in a thumbnail, might show that instance of the symbol is lit up.
- When in doubt, use something from the Flash interface.



VENOM Components

Symbol Shortcuts

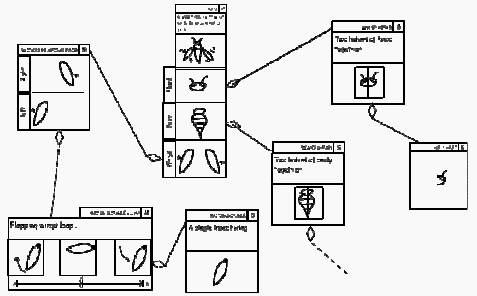
- Symbol Shortcuts
- Shortcuts can be used in cases where it does not suit you to draw or redraw the full container symbol.
- You can use short cuts to avoid having too many intersecting lines in a diagram.
- An example of a useful place to use a shortcut is a rollover button that has the same basic symbol in each of its states (U, O, D, H). Drawing the relationship line in from a short cut would be helpful in this situation.



VENOM Example

A Wasp

- To the right, is one way of structuring an animation of a wasp flapping its wings.
- A movie like this one could be set to follow a motion guide around a circuit.
- It is displayed and diagramed here in one spot, so it doesn't fly off the screen.
- This is a simple version; an example of a more complex version would be if the wings were only set to flap when the insect was flying.



Tip

The Venom Process

- Diagramming using Venom can be an iterative process, for instance:
  1. You may do an initial Venom diagram as a type of brainstorming activity.
  2. You could then do a prototype from this diagram in Flash
  3. More changes, or better ways of doing things, could occur to you and so you could go back to your diagram to work them out.
- Discuss top down vs bottom up

Activities

Creating a VENOM Diagram

1. Working as part of a team with the researcher, create a Venom diagram of roll-over button such as 'example_rollover_button_v2.0"
2. In Flash, open the file 'flower_example_v02.fla' and have a go at doing a Venom diagram of it.
3. Open the file 'wasp_animation_v02.fla' and see if you can spot the differences between the what is in the file, and the diagram used in a previous slide. Discuss.
4. Choose a simple animation or interactivity task that you might typically use Flash for and do a Venom diagram of it. Then have a go at creating it in Flash.
5. Repeat (4) with another typical task if time is available.

Regroup

- How did it go?
- What questions do you have?
- Conclude with the following:
  1. Completing the questionnaire
  2. Discussing your experiences Using Venom, and any ideas that occurred to you while completing the questionnaire.

Revisions Since v1.1

1. Due to a change in the target audience the focus group session now no longer includes the lesson, 'Using Symbols in Flash'. Instead, just a concept check is included.
2. The Venom lesson is now instructor-led as opposed to self-paced.
3. Example fla files are included and used in an activity to help scaffold an understanding of Venom.
4. Use the fla file in combination with a completed diagram.
5. Include an explanation of Venom as an iterative process.
6. Discuss top down vs bottom up.
7. Include ActionScript symbol; 'A'
8. Clarify different options for description area
9. Amend wasp diagram to include single wing.
10. Clarify transition types by relating to Flash tweening, etc.
11. Clearly explain that individual layers of a symbol can be represented but must be separated from main symbol components by a double [ruled] line and named as layers.
12. More techniques for representing animation are required. Suggest using Flash interface icons and notation for these.
13. Explain usage as a potential planning and commutation tool for experts.
14. Explain that timeline can use seconds or frames as preferred.
15. Explain that different parts of the timeline (such as frames for up, over, down, and hit in the button type symbol) can have symbols aggregated into them.
16. Provide a handout of the lesson material.

Author Information.

- Mark Wilson
  - Mark currently works at Unisys in a technical sale support and business development role. Based in New Zealand, Mark's job is to support Unisys Data Centre Services in the Asia-Pacific region. Mark also studies part-time at VUW.
  - Email: mark.wilson@nz.unisys.com
- Mary Tate
  - Mary is a lecturer in the School of Information Management at Victoria University of Wellington (VUW). Mary has been Mark's supervisor for over three years.
  - Email: mary.tate@vuw.ac.nz