A Survey of Features in Visual IDEs for Non-Programmers from a Usability and Suitability Point of View

Jonathan Orbeck[1], Jean Michel Rouly[2], Dr. Eugene Syriani[3]

[1] University of Alabama, Tuscaloosa, Alabama, United States
[2] George Mason University, Fairfax, Virginia, United States
[3] Université de Montréal, Montréal, Quebec, Canada

THE UNIVERSITY OF ALABAMA

Université de Montréal

Research Question

What aspects of a visual programming IDE affect usability? Can these features be standardized?

Background

- Integrated Development Environments (IDEs)
- Visual languages
- Interface design

Background
IDEs

An ***I***ntegrated ***D***evelopment ***E***nvironment ...

- ➤ is generally domain specific
- ➤ supports development process
- ➤ integrates tools in uniform interface

A.N. Habermann and D. Notkin. Gandalf: Software development environments. Software Engineering, IEEE Transactions on, SE-12(12):1117{1127, Dec 1986. ISSN 0098-5589. doi: 10.1109/TSE.1986.6313007.

Background
Visual Languages

A visual language ...

- uses pictures to express computations
- consists of visual vocabulary, grammar, and semantics
- is more effective than text

D.L. Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. Software Engineering, IEEE Transactions on, 35(6):756–779, Nov 2009. ISSN 0098-5589. doi: 10.1109/TSE.2009.67.
Eric J Golin and Steven P Reiss. The specication of visual language syntax. Journal of Visual Languages & Computing, 1(2):141{157, 1990.

Background
Interface Design

Software Interfaces ...

- ➤ understand user desires and requirements
- ➤ plan for domain opportunities and constraints
- ➤ create useful, usable, and desirable products

Alan Cooper, Robert Reimann, and David Cronin. About face 3: the essentials of interaction design. John Wiley & Sons, 2007.

Research Proposal

1. **Select visual IDEs**
2. **Define features**
    a. select common IDE features
    b. formalize definitions
    c. establish value ranges
3. **Evaluate IDEs**
    a. measure IDEs for each feature
    b. conduct user study for qualitative features
4. **Prototype development framework**

Select Visual IDEs

IDEs by Domain

3D Modeling

Blender
Grasshopper 3D

Animation

Alice3

Modeling

AToMPM
MetaEdit+
UMLet
Violet
Visual Paradigm
Visual Use Case

Music

AudioMulch
Max

Prototyping

Cameleon

Simulation

MST
SimuLink
VisSim

Software

EMF
GNU Radio Companion
AppInventor
Piet Creator
Scratch
Stencyl
Tersus
TouchDevelop
WebRatio

Workflow

YAWL

Define Features

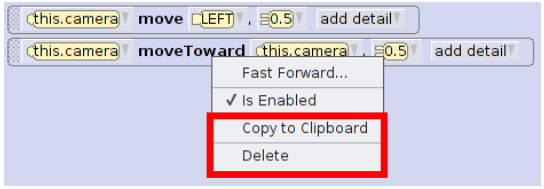
IDE Features

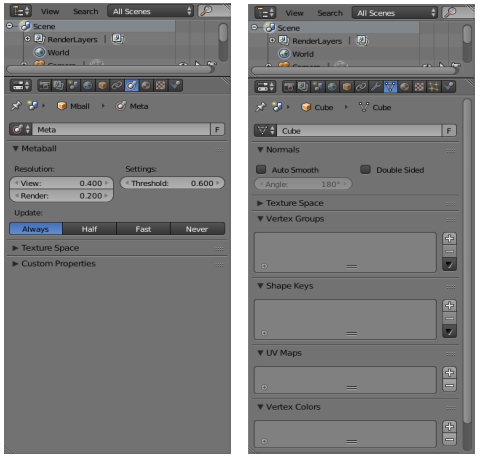
- **Define novel set of features**
- **Categories**
  - Audience
  - Chrome
  - Human Interface
  - Integration
  - Language Syntax

IDE Features
Audience

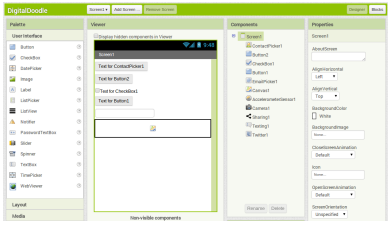
- **Domain**
  - field of knowledge
  - eg. 3D modeling, animation, music, software, etc.
- **Skill Level**
  - requisite entry-level skill
  - eg. novice, intermediate, expert, general

IDE Features
Chrome

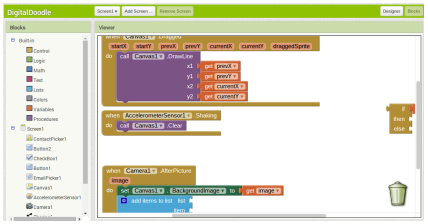




- **General Operations**
  - most frequently used IDE features
  - includes delete, save, paste, content assist, etc.



- **Context Sensitive Tools**
  - tools that change given context

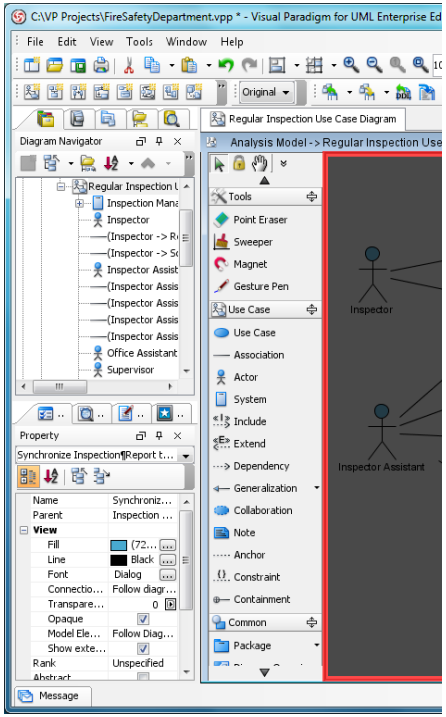


- **Multiplicity of Perspectives**
  - number of available predefined tool configurations

G.C. Murphy, M. Kersten, and L. Findlater. How are java software developers using the Eclipse IDE? Software, IEEE, 23(4):76–83, July 2006. ISSN 0740-7459. doi: 10.1109/MS.2006.105.c

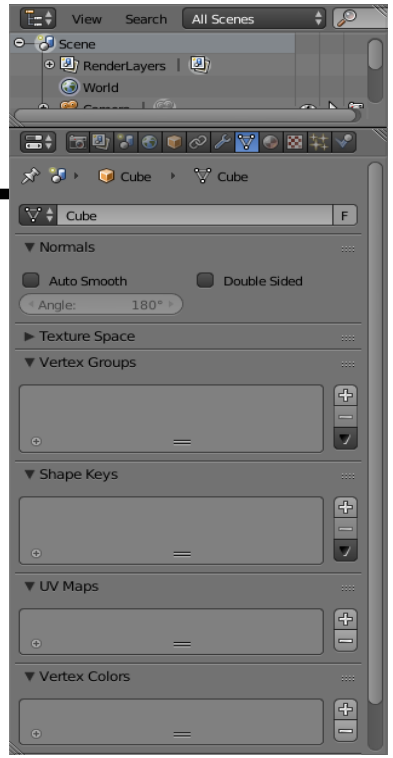
IDE Features
Chrome

- **Degree of Interface Visual Richness**
  - increase visual discriminability between tools
  - eg. icons, shape, size, color, etc.
- **Visual Clutter**
  - the number and organization of tools on the screen
  - qualitative metric

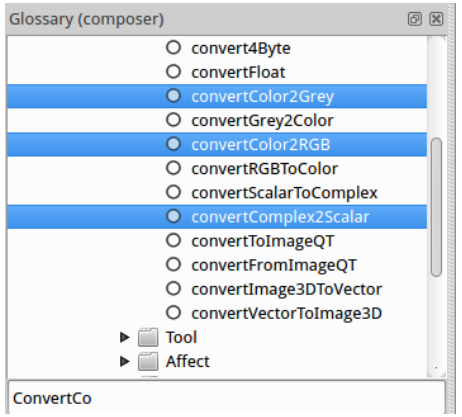


D.L. Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. Software Engineering, IEEE Transactions on, 35(6):756–779, Nov 2009. ISSN 0098-5589. doi: 10.1109/TSE.2009.67.

IDE Features
Chrome



- **Object Properties Window**
  - display mode of object properties dialog or window
- **Searchable Toolspace**
  - available tools can be reached through searching
- **Toolbar Styles**
  - interface component idioms
  - eg. sliders, toolbars, trees, icons, etc.



IDE Features
Human Interface

- **Essential Efficiency**
  - amount of mental load to complete a standardized task

$$1 - \frac{\text{Number of steps in concrete use case}}{\text{Number of steps in essential use case}}$$

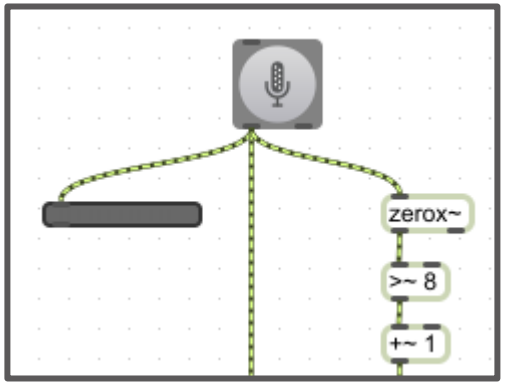
- **Interface Efficiency**
  - amount of physical action to complete a standardized task

$$1 - \frac{\text{Number of physical actions to complete use case}}{\text{Number of steps in essential use case}}$$

L.L. Constantine. “Usage-centered software engineering: new models, methods, and metrics ”. In Software Engineering: Education and Practice, 1996. Proceedings. International Conference, pages 2–9, Jan 1996. doi: 10.1109/SEEP.1996.533974.

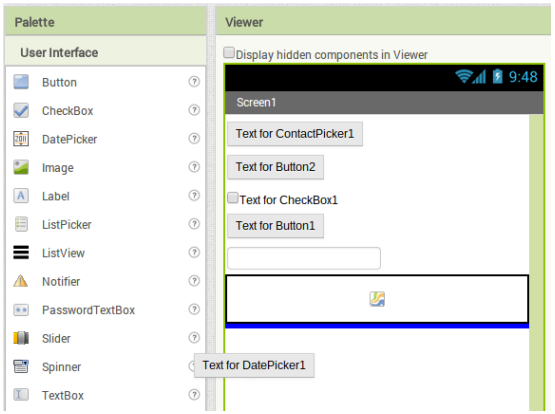
IDE Features
Human Interface

- **Keyboard Use**
  - level of interface support for keyboards



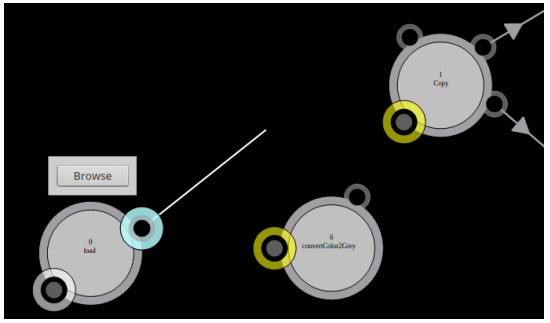
- **Tertiary Interface Devices**
  - level of interface support for third-party devices

- **Mode of Element Creation**
  - process to create elements in workspace
  - eg. drag n drop, point n click



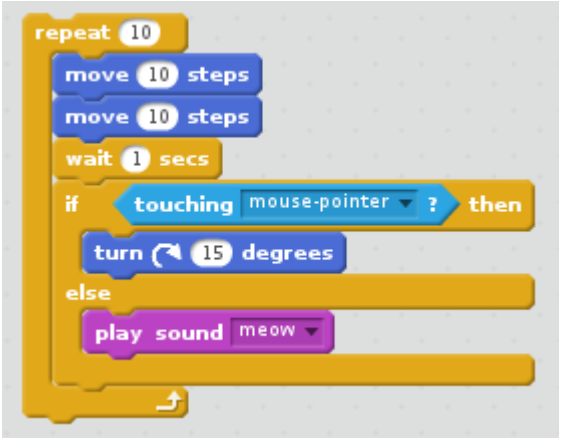
IDE Features
Integration

- **Allowed Relations Indicated**
  - syntactically correct connections highlighted

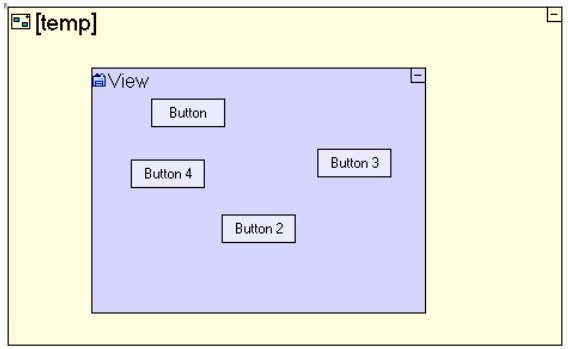


- **Output Generation Style**
  - relationship between user-created model and final output

- **Syntax Enforcement**
  - how the IDE enforces language syntax
  - explicit vs implicit enforcement

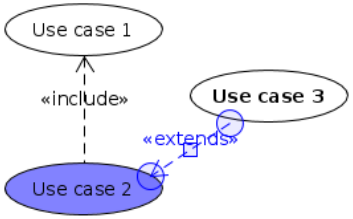


IDE Features
Language Syntax



- **Complexity Management**
  - characteristics to reduce language complexity

- **Connection Style**
  - mode by which connections are created and displayed



- **Degree of Language Visual Richness**
  - used to increase visual discriminability between elements
  - eg. icons, shape, size, color, etc.

Evaluate IDEs

Data Collection

- **Measure IDEs**
  - for each IDE, measure values of each variable
  - some variables required in-depth analysis
    - essential & interface efficiency
    - visual clutter

Data Collection
Efficiency

- **Create essential use cases**
  - 3 for each IDE
  - increasing amount of complexity
  - highest tier determined to be most representative

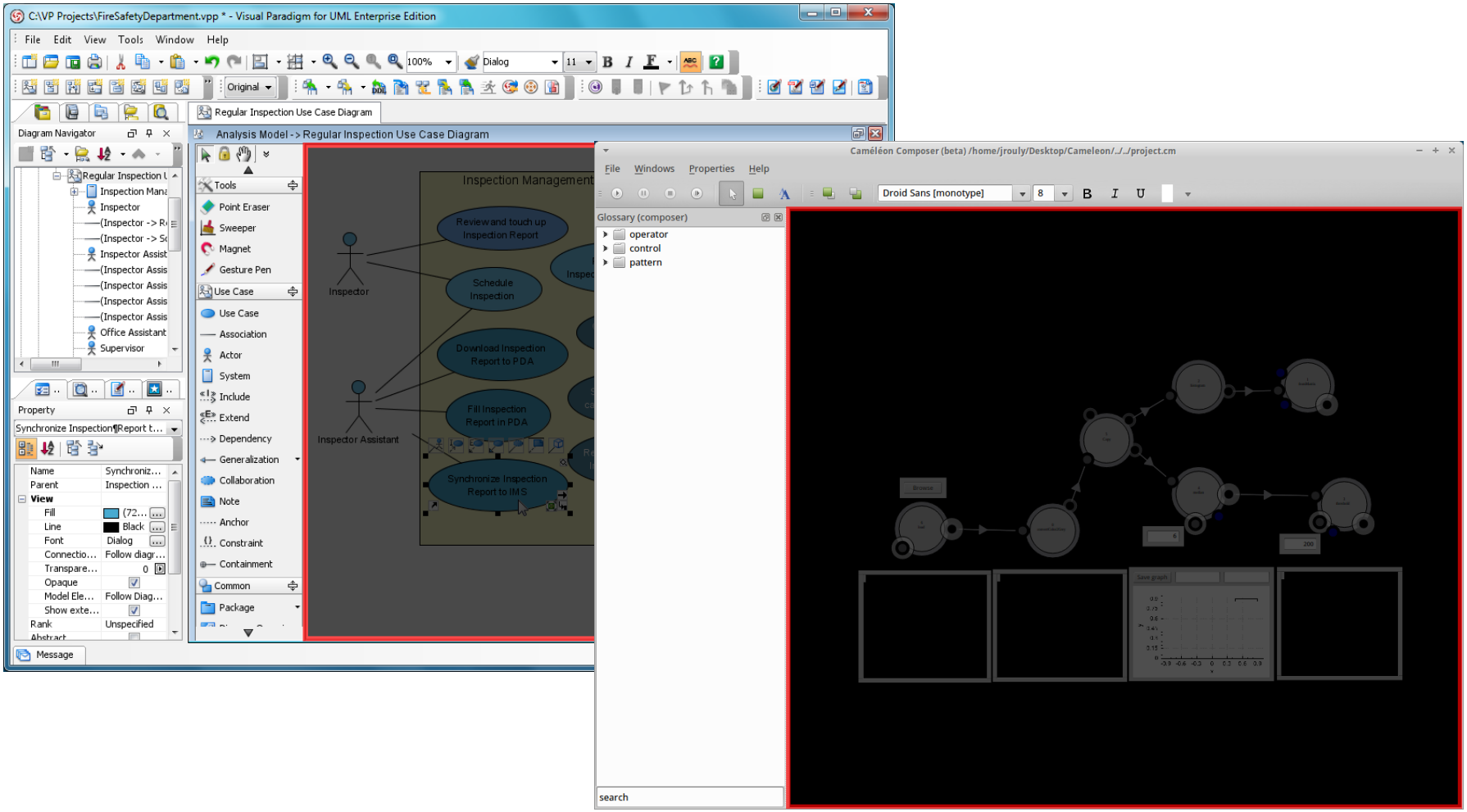
- **Assess concrete use cases**
  - concretely execute each use case
  - record steps & physical actions

Data Collection
Visual Clutter

- **User study performed on Amazon.com Mechanical Turk**
  - workers rated screenshots for clutter
  - 3 screenshots per IDE, varying complexity
  - 5 unique workers per screenshot
  - calculated averages for final values
  - inter-rater reliability good, ICC=0.648

amazon mechanical turk™
Artificial Artificial Intelligence

Data Collection
Visual Clutter



Prototype Development Framework

AToMPM

AToMPM is “a research framework from which you can generate domain-specific modeling web-based tools that run on the cloud”

E. Syriani, H. Vangheluwe, R. Mannadiar, C. Hansen, S. Van Mierlo and H. Ergin. AToMPM: A Web-based Modeling Environment. MODELS'13: Invited Talks, Demos, Posters, and ACM SRC. CEUR-WS.org: 1115, Miami, U.S.A., oct (2013).

AToMPM

- **Contributions**
  - guided by results of IDE analysis
  - intended to increase ease of use
  - developed API Plugin
    - extract common interface actions
    - decrease required system familiarity for end user


```
API.openModelViewer   / Open model selection dialog
API.fireStatechart    / Broadcast a known event to Statecharts
API.drawElement       / Draw a custom canvas element
API.deleteElement     / Delete a canvas element by ID
API.drawEdge          / Construct an edge between two elements
API.dotConvert        / Convert a filepath to ArkM3 notation
```

Outcomes and Future Work

Outcomes

- Set of formal interface feature definitions
- Evaluation technique for new IDEs
- Paper detailing results
- Foundation for AToMPM API

Future Work

- **AToMPM**
  - generalize functions to AToMPM API
  - incorporate visual variables
  - complete user study of platform usability

- **IDE Survey**
  - complete user studies of more variables
  - perform statistical analysis & validation of results
  - add more IDEs, develop more features of analysis

Questions


AtomPM
Alice3
WebRatio
VioletUML
MST
Grasshopper3D
SimuLink
MITAppInventor2
VisSim
Tersus
PietCreator
GNURadioCompanion
YAWL
Stencyl
Max
UMLet
VisualUseCase
Scratch
Blender
Cameleon
TouchDevelop
MetaEdit+
EMF
VisualParadigm
AudioMulch

Appendices

IDEs Explored