

# CSEN1082: Visual Programming

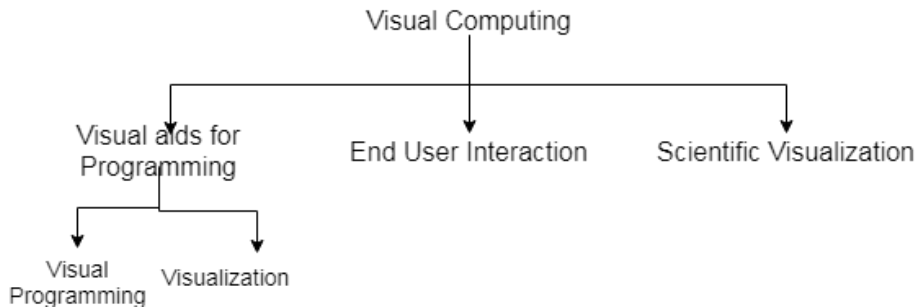
## Week 7: Evaluation of a VPL

Nada Sharaf

Spring Semester 2019

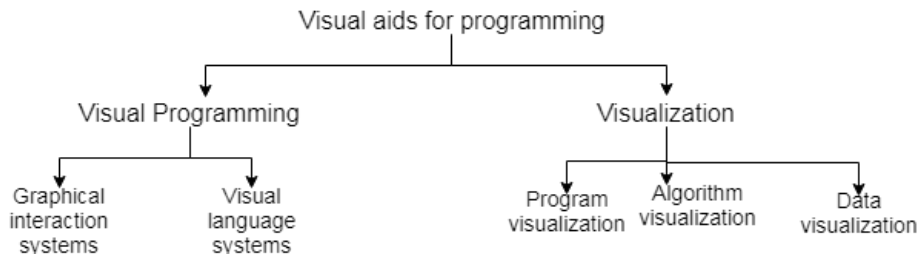
- Visual approaches are appealing.
- But are they good or better than corresponding textual approaches
- What is a *Complete* visual language.
- Is there a turing test for VPLs?

# Visual Computing



- Visual Programming allows the user to specify the program in two or more dimensions
- A textual program has a one-dimensional stream

# Visual Aids for Programming



# Graphical Interaction vs. Visual Language Systems

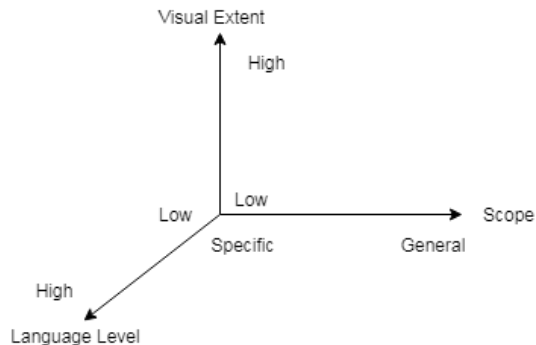
In a graphical interaction system

- The system learns from the sequence of actions of the user
- Programming By Example
- *Do What I Did*
  - ▶ The user specifies everything about the program
  - ▶ The system remembers the examples to use later on
- *Do What I Mean*
  - ▶ The system infers the program structure from the examples provided by the user
  - ▶ Automatic Programming

## Visual Language Systems

- use icons, symbols, ... etc to specify a program
- It is all about the **spatial arrangement**.
- Visual Sentences with spatial interpretations are formed

# Classification



- Visual Extent
  - ▶ The use of graphical objects as its programming constructs
- Scope
  - ▶ Indicates whether the language could be applied in a specific/limited area or to a wider range
- Language level
  - ▶ High vs low level language



# Classification Contd.

- There are different other methods to classify.
- Paradigms, features, ... etc

# Evaluation of a Visual Programming Language (VPL)

- Turing test for VPLs
- A programmer creates a program in the candidate VPL
- Another programmer creates equivalent program in a textual language.
- An examiner asks both questions to see who has better understanding in a shorter time.
- *When does a VPL pass the test?* If the examiner cannot distinguish both or if they judge the VPL programmer to have a better understanding

# More Pragmatic Methods

- visual nature
- functionality
- ease of comprehension
- paradigm support
- scalability.

- A visual language should depend on graphical/visual components
- e.g. Vopr uses concentric circles
- This criteria is not related to how easy the language is to use but to how visual it is

# Visual Nature Metrics

Quality assessed	Highest rating				Lowest rating
Use of graphics	5. Entirely graphic, e.g. iconic	4. Primarily graphic	3. Limited graphic with text annotation	2. Text with graphic decorations	1. Entirely textual
Type of graphic used	5. Meaningful icons and diagrams	4. Fairly meaningful icons and diagrams	3. Less meaningful icons and diagrams	2. Forms or simple graphics	1. No graphics
Thoroughness of graphic use	5. Applicable throughout language	4. Applicable to most semantics	3. Applicable to about half of semantics	4. Applicable to a few semantics	1. No graphics
Effective use of spatial arrangement	5. Effective use throughout	4. Effective use in some aspects	3. Average effectiveness	2. Limited spatial effectiveness	1. Poor use of spatial arrangement
Effective use of color	5. Effective use throughout	4. Effective use of color in some aspects	3. Color used for discrimination	2. Sparse use of color	1. No use or inappropriate use of color

# Visual Nature Metrics

Quality assessed	Highest rating				Lowest rating
Use of graphics	5. Entirely graphic, e.g. iconic	4. Primarily graphic	3. Limited graphic with text annotation	2. Text with graphic decorations	1. Entirely textual
Type of graphic used	5. Meaningful icons and diagrams	4. Fairly meaningful icons and diagrams	3. Less meaningful icons and diagrams	2. Forms or simple graphics	1. No graphics
Thoroughness of graphic use	5. Applicable throughout language	4. Applicable to most semantics	3. Applicable to about half of semantics	4. Applicable to a few semantics	1. No graphics
Effective use of spatial arrangement	5. Effective use throughout	4. Effective use in some aspects	3. Average effectiveness	2. Limited spatial effectiveness	1. Poor use of spatial arrangement
Effective use of color	5. Effective use throughout	4. Effective use of color in some aspects	3. Color used for discrimination	2. Sparse use of color	1. No use or inappropriate use of color

- Restricting the domain can help make the language more visual
- However a *complete* VPL is a general purpose one
- Vipr meet the criteria of a general purpose language
- An operational proof would be to build a translator from the VPL to a general purpose language

# Functional Completeness Metrics

Quality assessed	Highest rating				Lowest rating	
Functional completeness	5. General purpose functionality	4. Missing a few capabilities	3. Applicable to many, but not all areas	2. Applicable to several areas	1. Special purpose	
Naturalness of application	5. Natural to all applications	4. Natural to most	3. Natural to many	2. Natural to several	1. Natural to a few	



# Ease of Comprehension

- Depends on who the system is tested with
- textual Languages do have advantages

# Ease of Comprehension Metrics

Quality to be assessed	Highest rating				Lowest rating
Ease of comprehension	5. Much easier than comparison language	4. Moderately easier than comparison language	3. About the same as comparison language	2. Somewhat less than comparison language	1. Much less than comparison language
Ease for programmers	5. Much easier	4. Moderately easier	3. About the same	2. Somewhat less	1. Much less
Ease for technical non-programmers	5. Much easier	4. Moderately easier	3. About the same	2. Somewhat less	1. Much less
Ease for managers	5. Much easier	4. Moderately easier	3. About the same	2. Somewhat less	1. Much less
Ease for programming novices	5. Much easier	4. Moderately easier	3. About the same	2. Somewhat less	1. Much less

# Paradigm Support

- For effectivity, a VPL should support a paradigm
- For better usability, it should be customizable to multiple paradigms
- e.g. Vista

# Paradigm Support Metrics

- For effectivity, a VPL should support a paradigm
- For better usability, it should be customizable to multiple paradigms
- e.g. Vista

- Can it express a big program
- Can it organize portions of the program for better viewing
- *Zooming*
- Proof by Example

# Scalability Metrics

---

Quality to be assessed	Highest rating				Lowest rating
Support for modularity	5. Strong	4. Moderate	3. Some support	2. Weak	1. None
Support for abstraction	5. Strong	4. Moderate	3. Some support	2. Weak	1. None
Support for information hiding	5. Strong	4. Moderate	3. Some support	2. Weak	1. None
Support for data encapsulation	5. Strong	4. Moderate	3. Some support	2. Weak	1. None

---

# Thank You