

Introduction

Field of “Comparative Literature”

Working on Artificial Intelligence

Computer Language as Human Language

Constructing an Analytic Position

Determinacy, Indeterminacy, Narrativity

Conceptual Relations and Abstract Machines

Primary Concerns

Formulating a Point of Inquiry

Conceptualizing the Structure of Inquiry

Presenting the Subject of Inquiry

A Calculus of Aspect Relations

Intended Audience

Programmers

Interface Designers

Anyone Concerned with Indeterminacy

Anyone Concerned with Formulating
Structures of Intelligence

Section 1

Objects

Programmatic Manipulation and Concept Design

Object-Orientation

Concept-Based Design

Properties and Messages

Properties

Units of Memory

Organizations in Hardware

Objects as the Computer Sees Them

Messages

Black Box

Units of Conceptual Duration

Objects as One Interacts With Them

Machine Expectations

Terms of Contingency

Demanded by Substrate

Hard Limits Determining Options

Program Expectations

Terms of Contingency

Demanded by Anticipated Utilization

Flexible Terms of General Expectation

Managed Runtime Environments

Virtual Environments

Starting at Zero

Redefining Terms of Contingency

Operative Terms of Interaction/Intervention

Modularity and Inheritance

Environmental Orientation

Inversion of Control

Join Points and Point Cuts

Monkey Patching

Closures

Section 2

Architectural Circuits

Indeterminate Concepts and
Domain Knowledge

Modeling an Environment

Where do We Start?

Systems and Sub-Systems

Organizations of Program Flow

Model-View-Controller

Architecture for Modeling Program Flow

Aspect-Based Design

All Concepts Theoretically Follow

Domain Knowledge

Triangulation of Flow Structures

[Diagram]

Turing's Imitation Game

“The Turing Test”

A System of Inquiry

Reflexive Identification with Difference

Viewer Modulated by View

Control Structure for Inquiry

Reflexivity

The Sending of Automation

The Return of Sending

Circuits of Anticipation

A-B-C - M-V-C

[Diagram of Imitation Game]

The Object is an Aspect

C is the double-framing of A

A is B's Verification

A is the Indeterminacy of Inquiry

C is the Control Structure

B is the View

Section 3

Aspects and Controllers

Reflexive Circuit Design

Turing's Imitation Game as M-V-C

Organizing Domain Knowledge

Constructing Inquiry as a Subject

Constructing a Machine of Inquiry

Inquiry Returns Knowledge to its Origin

Connecting Concepts

Black Box Principles

One-One Relations

Program Freedoms

Interface Freedoms

Programming Aspects

Abstracting Interface Freedoms

Determining Internal Constraints

Measuring External Contingencies

The Pre-Determined Comes to Life

Aspect Tensions

The Object is Determined
Beforehand, by the Concept

The Concept is an Interface

The Interface is Programmatic

The Program is a Second Interface

Section 4

Degrees of Freedom

Abstracting Relations and
Environmental Mechanics

Developing Intelligent Control Structures

Modular Encapsulation

Reflexive Interface Dynamics

Coding for Interfaces

Replacing Machine Expectations
with Concept Expectations

Designing Possibilities of Orientation

How Things Are

How We Interact with Things

The Difference

Code Organized by Expression

Interface is What an Object Does

Interface is the “Technology”

Technology is the Capacity to Interface

Forward-Driven Program Activity

Programming Domain Knowledge

[M-V-C Push/Pull Diagram]

Relating Domains of Knowledge

[MVC and the Imitation Game Diagram]

[MV-VM One-One Relations Diagram]

[Cascading Model-Control Diagram]

[Converging Control-Control Diagram]

Controllers as Unifiers of Scope

Controllers Consolidate Multiple Scopes

Form an Interstice between Models

Delimit Domains of Activity

Conclusion

Tools to Organize Architecture

Isolate Control Aspects

Build from Zero

Construct Freedoms, not Constraints

Design for Two Articulations

Language is the Barrier, Objects are the
Conceptual Solution