

บทที่ 3

พัฒนาการของ Software Technology

บทนำ

เทคโนโลยีการพัฒนาองค์ประกอบของคอมพิวเตอร์นั้นมิได้มีความก้าวหน้าในการพัฒนาทางด้าน Hardware เท่านั้น แต่ในทาง Software ก็พัฒนามิได้ยิ่งหย่อนไปกว่ากัน กล่าวคือมีการพัฒนาเทคนิคของการโปรแกรมเชิงวัตถุ (Object Oriented Techniques) ผนวกเอาส่วนของ Integrated CASE tools , Code Generator , CASE repositories ,Class libraries , Formal methods เข้าด้วยกันในอันที่จะสนองตอบต่อการนำไปใช้ในการทำงานทางด้านการพัฒนาเทคโนโลยีของ Software (Software Development) โดยมีเจตนาที่จะให้ Software นั้นๆ สามารถนำกลับมาใช้ใหม่ได้อย่างคุ้มค่ายิ่งขึ้น

โดยตัวของ Object Oriented Programming เองส่วนเดิวนั้นจะไม่สามารถจะสนองตอบต่อเจตนารมย์ดังกล่าวได้ จึงมีความจำเป็นที่จะต้องผนวกเอาความรู้และพัฒนาการทางด้านอื่นเข้ามาาร่วมด้วย

เราต้องยอมรับประเด็นสำคัญอย่างหนึ่งที่เกิดขึ้นก็คือเรื่องของการพัฒนาระบบ (Redesign System) เป็นสิ่งที่หลีกเลี่ยงมิได้ และงานส่วนหนึ่งในการพัฒนาระบบที่เป็นส่วนที่สำคัญและใช้เวลาค่อนข้างมากในการจัดการก็คือการพัฒนาและสร้าง โปรแกรมขึ้นมาใช้งาน ดังนั้นการสร้าง Object Oriented Model ขึ้นมาใช้งาน ก็มีเจตนาที่จะแก้ปัญหาในส่วนนี้ให้ลุล่วงไปได้อย่างรวดเร็ว มีประสิทธิภาพและประหยัดค่าใช้จ่าย โดยที่เครื่องมือของ OOP จะทำให้เราสามารถถ่ายทอดกระบวนการดำเนินงานให้เป็นระบบเพื่อให้สามารถดำเนินงานได้อย่างเป็นอัตโนมัติ

พัฒนาการทางซอฟต์แวร์ (A Software Revolution)

ลักษณะสำคัญในการพัฒนาอุตสาหกรรมทางด้าน Software ก็คือต้องการเทคนิคและแนวทางในการที่จะทำให้ Software ที่สร้างขึ้นมานั้นสามารถทำงานได้เร็วขึ้นโดยมีต้นทุนในการผลิตที่ต่ำ และในการที่จะสามารถสนองตอบต่อความต้องการดังกล่าวได้นั้น Software ดังกล่าวก็จะต้องมีความซับซ้อนมากขึ้น มีความน่าเชื่อถือ มีหลักประกันของความถูกต้องในระดับสูง และดังที่กล่าวมาตั้งแต่แรกแล้วว่า OOP มิใช่แนวทางเดียวในการแก้ปัญหาในการพัฒนา Software ในลักษณะที่ต้องการ ดังนั้นจึงมีการนำเทคโนโลยีอื่นๆ ที่มีส่วนเกี่ยวข้องกันมาประสานเข้าด้วยกันกับ Software หลัก เพื่อดึงเอาความสามารถของแต่ละส่วนมาใช้เป็นประโยชน์

องค์ประกอบของคุณลักษณะที่ดี ของ Software ที่ต้องการจะรวมเรียกว่า Quantum Leap ซึ่งจะประกอบด้วยคุณสมบัติต่อไปนี้

- Reliability
- Design Capability
- Flexibility
- Speed of Development
- Ease of Change

แนวความคิดของการโปรแกรมเชิงวัตถุ (Object Oriented Technique) ก็คือการนำเครื่องมือหลายๆอย่างมาผนวกเข้าด้วยกันในอันที่จะนำมาช่วยการออกแบบในส่วนต่อไปนี่คือ

- Object
- Behavior

การวิเคราะห์ในเชิงวัตถุจะเน้นไปที่การวิเคราะห์การสร้างวัตถุในสภาพแวดล้อมที่เกิดขึ้น รวมถึงการสร้างเหตุการณ์ (Events) ที่จะมีผลกระทบต่อวัตถุ การออกแบบนั้นจะดำเนินการโดยการใช้ Object Class ที่มีปรากฏอยู่แล้วให้สามารถนำมาใช้เป็นประโยชน์ หรือในบางครั้งก็อาจจะมีการสร้าง Class ขึ้นมาใช้ใหม่โดยใช้คุณสมบัติของการสืบทอดถึงกัน แนวความคิดในการสร้างระบบโดยวิธีการใหม่ก็คือพยายามแก้ปัญหาของระบบที่ซับซ้อนให้ง่ายขึ้น

Object-oriented technique simply the design of complex system

แนวคิดการจัดการกับระบบงานในเชิงวัตถุ เราสามารถมองระบบได้ว่าระบบก็คือ กลุ่มของวัตถุที่ประกอบกันเข้า โดยที่แต่ละวัตถุก็จะมีการกำหนดคุณสมบัติตามที่ระบบต้องการ และเราสามารถสั่งการให้วัตถุดำเนินการตามที่เรต้องการได้ก็โดยการกำหนดเหตุการณ์ให้กับวัตถุนั้นๆ โดยวิธีการดำเนินงานเช่นนี้ จึงทำให้เกิดการแนวทางปฏิบัติใหม่ในส่วนของ อุตสาหกรรม Software ซึ่งแตกต่างจากยุคของการพัฒนา Software ในรูปแบบเก่าที่เรียกว่า Monolithic Software

Killer Technology

ดังที่ได้กล่าวมาแล้วว่า การอาศัยแต่ Object-oriented technology แต่เพียงอย่างเดียวนั้นเราไม่อาจบรรลุวัตถุประสงค์ที่ต้องการได้ ดังนั้นจึงมีการนำเอา Software Technology อย่างอื่นเข้ามาประสานร่วมด้วย โดยที่บรรดา Software Technology ที่รวมเอามาเป็น Killer Technology for Software Development นั้นจะประกอบด้วย

- CASE and I-CASE
- Visual Programming
- Code Generator
- Repository and Repository Coordinator

- Repository-Based Methodologies
- Information Engineering
- Object-Oriented Databases
- Nonprocedural Languages
- Formal Mathematically-Based Methods
- Inference Engine
- Client-Server Technology
- Class Libraries which maximize reusability
- Object-Oriented Analysis and Design

ในบรรดา Killer Technologies ที่กล่าวมาแล้วนี้ได้ถูกผนวกเข้าไว้ด้วยกันให้เป็น Object-Oriented Framework

กระบวนการในการ สร้าง Software ก็เปรียบดังเช่นการสร้างเครื่องยนตขึ้นมาใช้งาน คือเราสามารถนำชิ้นส่วนบางชิ้นที่เคยใช้แล้วนำกลับมาใช้ใหม่ (Reuse) รวมถึงการซ่อนเร้นที่ทำให้ผู้ ออกแบบระบบอาจจะไม่รู้ด้วยซ้ำไปว่าภายในชิ้นส่วนนั้นดำเนินการทำงานได้อย่างไร

บรรดา Killer Software Technologies ทั้งหมดที่นำมารวมกัน เป็น Object-Oriented Framework จะประกอบด้วย Technology ต่อไปนี้

- **CASE** (Computer-Aided Software Engineering) นับเป็น Technology ที่สำคัญมากอัน ก่อให้เกิด พัฒนาการทางด้านอุตสาหกรรม Software โดยที่ เราจะนำ CASE Tool มา ใช้ช่วยการออกแบบในเชิง Graphic ซึ่งจะปรากฏออกมาหลายลักษณะงาน เช่น การวางแผนการดำเนินงานขององค์กร การออกแบบวงจรไฟฟ้า หรือแม้กระทั่งการออกแบบ Chip ของคอมพิวเตอร์ ภายหลังจากการออกแบบก็สามารถที่จะ Generate ออกมาเป็น Software ได้ด้วย
- **I-CASE** (Integrated CASE) เป็นคุณลักษณะพิเศษของ CASE Tools ที่จะใช้ใน เรื่องของการทำ Code Generator คุณสมบัติของ Code Generator นับเป็นสิ่งที่อำนวยความสะดวกอย่างมากอย่างยิ่งของการสร้างรหัสขึ้นมาใช้งาน
- **Visual Programming** จัดว่าเป็น CASE ประเภทหนึ่ง แต่จะเน้นในการนำไปใช้ในการออกแบบในเชิง Graphic การให้สีภาพ สามารถสร้าง Metafile ได้ด้วย โดยที่ Object ที่ปรากฏนั้นสามารถสื่อสารเป็น Visual Form ได้
- **Code Generator** เป็นเครื่องมือที่เกิดขึ้นจากความพยายามของมนุษย์ที่จะสร้างให้มี Software Tool ที่สามารถสร้างโปรแกรมในระดับของ High Level Language โดยดำเนินการสร้างโปรแกรมจาก ตัวแบบ หรือคุณลักษณะที่มนุษย์เรากำหนดขึ้นมาได้นั้น

นับเป็นความใฝ่ฝันที่ผู้เขียนโปรแกรมต้องการมาก โดยที่ Software ดังกล่าวสามารถนำองค์ประกอบเหล่านี้คือ Decision Tables , Rule , Action Diagram , Diagram of Events ; State Transition Diagrams ซึ่งเป็น Object หรือคุณสมบัติของ Object มาสร้างเป็นให้เป็นโปรแกรมที่ปลอดจาก Syntax error ได้ ซึ่งนับเป็นเครื่องมือที่สุดยอดสำหรับผู้เขียนโปรแกรม

- **Repository** นับเป็นลักษณะพิเศษอย่างหนึ่งที่ซึ่งช่วยจัดการในส่วนของ การเก็บคลังแห่งความรู้ ที่จะช่วยนักวิเคราะห์สามารถนำใช้ประโยชน์ต่อไปในการทำงาน โดยที่แผนการดำเนินการจะถูกส่งไปจัดเก็บในส่วนของ Repository ภายหลังจากมีการนำมาใช้งานในสภาพของตัวแบบ หรืออาจจะเป็นการใช้ เครื่องมือวิเคราะห์ (Analysis Tool) ก็ได้ การนำสารสนเทศที่จะเป็นพื้นฐานของการสร้างโปรแกรมต่อไปนั้น อาจจะถูกเก็บในส่วนของ Repository โดยการดำเนินการผ่าน Design Tool ก็ได้ นอกเหนือจากนี้ Repository ยังช่วยในการสร้างส่วนประสาน (Interface) ระหว่าง เครื่องมือต่างๆ ใน I-CASE นอกจากนั้น Repository Coordinators Software ยังใช้ช่วยจัดการในเรื่องของการสร้างหลักประกันกับคุณลักษณะที่เป็น Integrity ของสารสนเทศใน Repository ด้วย

The repository is a mechanism for defining , storing and managing information about an enterprise , its data and systems.

The repository coordinator applies methods to the data in the repository to ensure that the data and their CASE representation have consistency and integrity .

- **Repository - Based Methodologies** Software Development ที่มีประสิทธิภาพเพิ่มขึ้นนั้นมักจะมีพื้นฐานมาจากการสนับสนุน ของส่วนที่เป็น Repository Based ทั่วโลกในการพัฒนาระบบ Software นั้นจะมีความสัมพันธ์กับกรรมวิธีและการดำเนินการในการจัดเก็บและจัดการกับองค์แห่งความรู้ในส่วนของ Repository Based เพื่อที่ส่วนของ Repository จะได้นำไปใช้ทำ Code Generator ต่อไป

A repository – based methodology is designed to take full advantage of an I-CASE toolset and to maximize reusability.

- **Information Engineering** จะเป็นการประยุกต์ Repository Based เพื่อนำไปดำเนินการในส่วนของการสร้างระบบในส่วนของการบูรณาการ งานวางแผนงานออกแบบ และงานสร้างระบบ โดยจะดำเนินการสร้างตัวแบบและพัฒนาปรับตัวแบบให้มีประสิทธิภาพมากที่สุดเท่าที่จะกระทำได้

Information Engineering applies modeling and design techniques to the enterprise as a whole (or to a large section of the enterprise) rather than to merely one project.

- **Object-Oriented Databases** . นับว่าเป็นฐานข้อมูลที่มีรูปแบบที่ดี ทั้งในแง่ของการจัดเก็บ และการดำเนินงานกับข้อมูลอย่างชาญฉลาด แทนที่จะเป็นที่จัดเก็บข้อมูลแต่เพียงอย่างเดียว Object-Oriented Databases ได้รับการพัฒนาและออกแบบมาให้มีรูปแบบที่สามารถจัดการกับข้อมูลเชิงกายภาพในหลายรูปลักษณะและในเชิงซับซ้อนได้อย่างมีประสิทธิภาพ ตลอดจนถึงระบบความปลอดภัยที่ป้องกันการที่จะดึงข้อมูลไปใช้โดยตรง โดยการเรียกใช้ข้อมูลนั้นจะต้องกระทำผ่านโดยการ Operation เท่านั้น

An Object-Oriented Databases is designed to store object data and methods with techniques that are efficient for object-oriented processing

CASE repository จัดว่าเป็น Object-Oriented Databases (ถึงแม้ว่า CASE tools นั้นจะไม่สนองตอบต่อแนวการดำเนินงานแบบ Object-Oriented ในเรื่อง Analysis , Design และ การเขียนโปรแกรมก็ตาม

- **Nonprocedural language** การเขียนโปรแกรมในแนวทางของ Procedural Language เช่น ภาษา COBOL , C นั้น เราจะต้องดำเนินการเขียนเรียงตามขั้นตอนที่จะต้องดำเนินการ แต่ในกรณีของการเขียนโปรแกรมในรูปแบบของ Nonprocedural language เราเพียงแต่กำหนดผลลัพธ์ที่เราต้องการเท่านั้น ส่วนหน้าที่ของการสร้างโปรแกรมนั้นจะเป็นหน้าที่ของตัว Code Generator

Nonprocedural languages defined what is wanted, rather than how it is programmed.

ในปัจจุบันจะมี ภาษา Fourth GL หลายภาษาเช่น Focus ,FoxPro ที่สามารถดำเนินการได้ทั้งในสองส่วนคือ Nonprocedural languages และ Procedural languages

- **Formal Mathematically-based Methods** . ได้มีการนำเทคนิคเชิงคณิตศาสตร์มาช่วยในการสร้าง Specification ของโปรแกรมตลอดจนการทำหน้าที่ในการตรวจสอบความผิดพลาดที่เกิดขึ้นในโปรแกรม ตัวอย่างของ Mathematically-Based Methods . ซึ่งมีองค์ประกอบของ Z Specification language และการใช้ OCCAM Programming Language

Formal methods are used to guarantee that programs perform as intended with zero error

- **Inference Engine** มีหน้าที่ดำเนินการในส่วนที่เกี่ยวข้องกับกฎเกณฑ์ต่างที่ปรากฏใน Knowledge based โดยการคัดเลือก เชื่อมโยงประสานองค์แห่งความรู้ตามหลักเหตุผล จากกฎที่กำหนด การดำเนินการดังกล่าวอาจจะดำเนินการได้ทั้งรูปแบบของ Forward Chaining (Input Direct Reasoning) หรือ Goal Direct Reasoning (Backward Chaining) หรือดำเนินการทั้งสองแบบก็ได้ ด้วยกระบวนการดังกล่าวจึงทำให้คอมพิวเตอร์สามารถดำเนินการในส่วนของ Complex Deduction ได้โดยไม่ต้องพึ่งพิงการจัดการด้วยโปรแกรม แนวทางดังกล่าวจัดเป็นพื้นฐานความรู้ขั้นปฐมภูมิอันนำไปสู่การสร้าง Artificial Intelligence Software

An inference engine is software that makes deductions form facts and rules by using techniques of logical inference

- **Client Server Technology**

วิวัฒนาการทางด้าน Software ทำให้เราสามารถใช้ Software ดำเนินการคอมพิวเตอร์บนระบบเครือข่ายได้ ดังเช่นการปฏิบัติงานโดยระบบ LAN Server System , Coopreative System , Distributed Computing และระบบ Parallel Computer Client Server Computing จัดว่าเป็นเทคโนโลยีในการผสมผสานของ Software โดยแนวทางต่อไปนี้

A Client is a software module that requests an operation

A server is a software module that responds to that request

- **Class Libraries** จัดเป็นแหล่งที่อยู่ของ Object Type ต่างๆ ที่ซึ่ง Programmer จะนำทรัพยากรเหล่านี้ไปใช้ในการเขียนโปรแกรม อันทำให้ลดเวลาในการพัฒนาโปรแกรมไปได้ ในลักษณะดังกล่าวจึงทำให้เราสามารถนำของที่ใช้แล้วกลับมาใช้ใหม่ได้ จึงทำให้เกิดการเพิ่มขีดพัฒนาการทางด้าน Software ให้สูงขึ้น

A class is a software implementation of an object type A class can have many subclass

- **Object Analysis and Design** การวิเคราะห์ระบบในปัจจุบันนี้ ได้เปลี่ยนจากการวิเคราะห์ระบบในรูปแบบสมัยดั้งเดิม มาเป็นการออกแบบในเชิงวัตถุแทน ทั้งในส่วนของ การจัดโครงสร้างข้อมูล แนวทางในการดำเนินงาน ตลอดจนถึงการกำหนดเหตุการณ์ (Event) ที่จะนำมาสั่งในการดำเนินงาน โดยที่ปฏิบัติการที่เกิดบน Object หนึ่งๆนั้นจะมีผลมาจาก การสั่งการของ Object อื่น นักวิเคราะห์ระบบจะดำเนินการสร้างตัวแบบของ Object Structure และ Event ซึ่งจะเปลี่ยนแปลงให้ Object ดำเนินการตามต้องการ ในขณะที่พวกที่ทำหน้าที่ออกแบบโมเดลนั้นก็จะมีงานคล้ายๆกับ นักวิเคราะห์ระบบเพียงแต่จะมีการเข้าถึงในรายละเอียดในขั้นของการสร้างรหัสเพิ่ม ขึ้นมาด้วย

Object-oriented analysis and design model the world in terms of objects that have properties and behavior, as well as events that trigger operations that change the state of the objects . Objects interest formally with other objects .

สรุปผลที่เกิดจากการใช้ OO Techniques ว่าจะก่อให้เกิดการปฏิวัติอุตสาหกรรม Software
ดังนี้

- The entire software industry
- The way application packages are sold
- The way we use computers
- The way we use networks
- The way we analyze systems
- The way we design system
- The way we use CASE tool
- The way we re-engineer corporations
- The job of all I.S. professionals .

The Integrating Paradigm เทคนิคของ OOP ทำให้อุตสาหกรรมทาง Software ได้รับประโยชน์เพิ่มขึ้น ไม่ว่าจะเพิ่มพูนทรัพยากรในแหล่งของ Object Libraries ซึ่งสามารถนำมาเชื่อมต่อกับ CASE Repositories เพื่อจะได้สร้าง Class หรือพัฒนา Class ขึ้นมาใช้ใหม่ตามความต้องการอย่างสะดวกและง่ายดาย และในขณะที่ เทคโนโลยีทางด้าน Software ได้พัฒนารุดหน้าไป เทคโนโลยีทางด้าน Hardware ก็ได้พัฒนาไปอย่างรวดเร็วเช่นกัน ไม่ว่าจะพัฒนาทางด้านความเร็วของการทำงานของไมโครโปรเซสเซอร์ หรือการพัฒนาที่จะให้มีหลายๆ โปรเซสเซอร์ ในคอมพิวเตอร์หนึ่งระบบ หรือการพัฒนา ระบบ Parallel Processing หรือการพัฒนา เทคโนโลยีทางด้าน Network เป็นต้น ดังนั้น เทคนิคของ OOP จึงจำต้องถูกนำมาพัฒนาไปพร้อมกับการพัฒนาเทคโนโลยีทางด้าน Hardware ด้วย นั่นหมายความว่า CASE Tool ที่สร้างขึ้นมานั้นจะต้องถูกออกแบบมาให้สามารถสนองตอบต่อสภาพแวดล้อมต่างๆของ Hardware ซึ่ง OO Technique นั้นจะต้องกำหนดมาตรฐานขึ้นมาสำหรับ Software ต่างๆด้วย

มาถึงในทศวรรษนี้แนวคิดในการสร้างและโปรแกรมได้เปลี่ยนโฉมจากภาพอดีต
ในแนวคิดของ Data Driven Program มาเป็น Event Driven Program หรือที่เรียกว่าการออกแบบเชิง
วัตถุ บทสรุปจากความคิดเห็นของคนในแวดวงคอมพิวเตอร์ที่ให้ออกคิดเกี่ยวกับ OOP ปากกุดังนี้
คือ

- *All computer analysts and designers should be trained in OO techniques.*
- *All universities, technical colleges , and business schools should teach OO thinking.*
- *All organizations involved with software should fully support the evolving standard from the Object Management Group.*
- *AN CASE tool vendors should upgrade their tools to support OO modeling , design , and code generation .*
- *Repository standards should evolve to support OO modeling ,design , and code generator .*
- *There should be industry wide acceptance **of** the OO diagramming standards.*
- *AN enterprises with an I.S. organization should create OO models of their operations with a view toward through redesign of business processes and better communication between I.S. and business people.*
- *Software vendors should evolve rapidly toward distributed object architectures.*
- *Software vendors should cooperate to create reusable class libraries and multi vendor software (as opposed to monolithic software).*

คำถามท้ายบท

1. จงอธิบายถึงความหมายของคำว่า Integrated Software
2. ผลประโยชน์ที่เกิดขึ้นในการพัฒนาอุตสาหกรรมทางด้าน Software มีอะไรบ้าง
3. คุณสมบัติของ Code Generator คืออะไร มีส่วนช่วย Programmer อย่างไร
4. ท่านเข้าใจความหมายของคำว่า Killer Technology อย่างไร
5. ประโยชน์ที่ได้จาก Visual Programming คืออะไร และเกี่ยวข้องกับอย่างไรกับงานที่เป็น User Interface
6. เราใช้ประโยชน์อะไรได้บ้างจากส่วนของ Repository
7. จงอธิบายถึงความแตกต่างระหว่าง Procedural Language กับ Non-procedural Language
8. ท่านคิดว่าในอนาคตวิธีดำเนินการของการสร้างโปรแกรมจะมีรูปแบบอย่างไร
9. ทำไมจึงจะต้องมีการกำหนดมาตรฐานของ CASE Tool