บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ประวัติจาวา

เริ่มต้นมาจากทางบริษัทซันไมโครซิสเท็มได้มีความคิดที่จะเปิดตลาดอุตสาหกรรมอุปกรณ์ควบคุมสัญญาณภาพทีวี (Set top box) เหมือนกล่องที่ใช้เปลี่ยนช่องสัญญาณทีวี ในระบบเคเบิล โดยทางทีมงานจำนวน 13 คนภายใต้ชื่อโครงการว่า “The Green Project” ได้สร้างอุปกรณ์ดังกล่าว โดยมีภาษาที่ใช้ควบคุมการทำงานที่ออกแบบมาโดยเฉพาะ และได้ตั้งชื่อภาษาว่าโอ๊ค (Oak) ตามชื่อต้นไม้ที่ขึ้นอยู่บริเวณสถานที่ทำงาน แต่มีปัญหาเรื่องการจดสิทธิบัตร จึงเปลี่ยนชื่อเป็นจาวา (Java) ตามชื่อเมล็ดกาแฟที่ใช้ดื่มในทีมงาน ซึ่งจะสังเกตเห็นสัญลักษณ์ภาษาจาวาเป็นแก้วกาแฟ และมีตัวนำโชค (Mascot) เป็นเมล็ดกาแฟชื่อ ดุ๊ก

ภายหลังอุปกรณ์ควบคุมสัญญาณภาพทีวีไม่ประสบผลสำเร็จ จึงได้นำเอาภาษาจาวาไปเผยแพร่ทางเว็บบนอินเทอร์เน็ต เมื่อวันที่ 23 พฤษภาคม 2538 ซึ่งถือเป็นวันกำเนิดของภาษาจาวา พร้อมออกชุดพัฒนาให้โปรแกรมเมอร์ได้ดาวน์โหลดเพื่อนำไปใช้งานได้ฟรีร่วมกับบราวเซอร์ที่ชื่อ Hot Java

ในยุคนั้นข้อมูลอินเทอร์เน็ตมีลักษณะเป็นเพียงตัวอักษรปกติ (plain text) แต่หลังจากที่โปรแกรมภาษาจาวาซึ่งทำงานบน Hot Java เรียกโปรแกรมว่า Java Applet ซึ่งทำให้ข้อมูลมีการแสดงเป็นทั้งภาพ ภาพเคลื่อนไหว และเสียงทำให้นักพัฒนาเริ่มให้ความสนใจตัวภาษาเป็นอย่างมาก ดังจะเห็นได้จากจำนวนครั้งในการดาวน์โหลดชุดพัฒนา จำนวนกลุ่มสนทนา (forum) ที่เพิ่มขึ้นจากเดิมเป็นอย่างมากในปัจจุบัน

เขียนครั้งเดียวใช้ได้ทุกที่ (Write Once, Run Anywhere)

เขียนขึ้นครั้งเดียวทำงานได้ทุกที่ เป็นนิยามของโปรแกรมที่สร้างจากภาษาจาวา โดยทางบริษัทซันไมโครซิสเท็มได้วางลักษณะของตัวภาษาจาวาเอาไว้ กล่าวคือเมื่อต้องการสร้างโปรแกรมขึ้นมาใช้งาน หากโปรแกรมเมอร์เลือกใช้ภาษาจาวาแล้วจะใช้เวลาเขียนไวยากรณ์ภาษาจาวาสำหรับโปรแกรมนั้นเพียงครั้งเดียว เมื่อได้โปรแกรมแล้วก็ส่งไปให้ผู้ใช้ เพื่อทำการใช้งานไม่ว่าผู้ใช้จะใช้โปรแกรมดังกล่าวบนระบบปฏิบัติการใด ๆ เช่น วินโดวส์ ยูนิกส์ ลีนุ๊กซ์ เป็นต้น โดยไม่ต้องมีการปรับเปลี่ยนไวยากรณ์ภาษาจาวาของโปรแกรมต่างกันหากโปรแกรมเมอร์เลือกใช้ภาษาอื่น ๆ แล้วโปรแกรมเมอร์จะต้องนำภาษาดังกล่าวไปสร้างเป็นโปรแกรมสำหรั บให้ผู้ใช้ สามารถใช้งานในระบบปฏิบัติการหนึ่ง เช่น วินโดวส์ หากแต่ผู้ใช้ ไม่สามารถนำโปรแกรมนั้นไปใช้ งานได้ ในระบบปฏิบัติการอื่นนอกเหนือจากวินโดวส์โดยปราศจากการแก้ไขและการปรับปรุงจากโปรแกรมเมอร์

โปรแกรมที่สร้างจากภาษาจาวา ทางบริษัทซัน ไมโครซิสเท็มเรียกทั้งหมดไม่ว่าจะเป็นสิ่งใด ๆ ว่า จาวาแพลตฟอร์ม (Java Platform) ปัจจุบันจาวาแพลตฟอร์มไม่ใช้เป็นแค่เครื่องคอมพิวเตอร์ แต่ยังรวมถึง PDA, โทรศัพท์มือถือ หุ่นยนต์ หรือเครื่องใช้ไฟฟ้า เช่น ตู้เย็ น เครื่องปรับอากาศ อีกด้วย หากสิ่งดังกล่าวสามารถทำงานโปรแกรมภาษาจาวาได้

จาวาแพลตฟอร์ม

จาวาแพลีตฟอร์คือคีย์ที่ทำให้เกิดการทำงานแบบ Write Once, Run Anywhere เนื่องจากทุกๆ ระบบที่ต้องการใช้งานโปรแกรมภาษาจาวาจะต้องมีกลไกการทำงานย่อยที่ เรียกว่าจาวาเวอร์ชวล แม็ชชีน (Java Virtual Machine) เรียกสั้น ๆ ว่า JVM

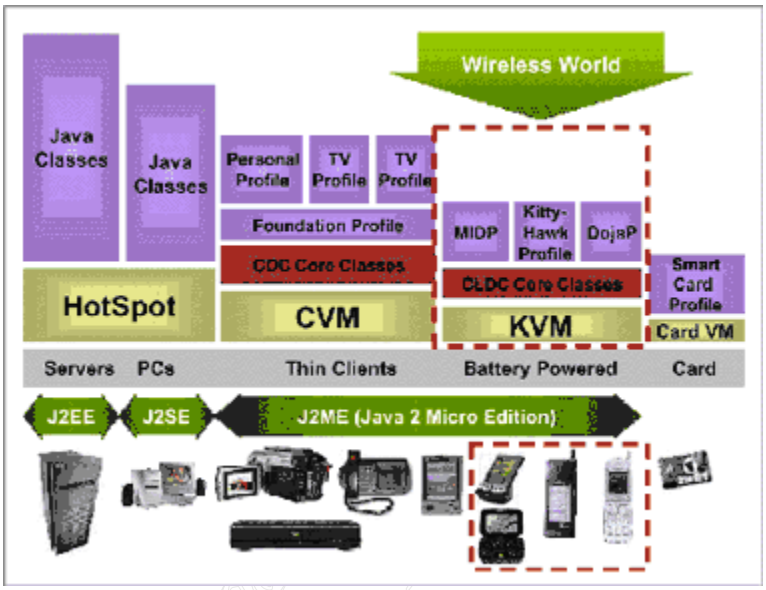
โปรแกรมภาษาจาวา คือโปรแกรมที่ได้จากภาษาจาวาที่ผ่านการคอมไพล์แล้ว การคอมไพล์ภาษาจาวาจะได้ภาษากลางที่เรียกว่าไบต์โค้ด (Bytecode) และไบต์โค้ดนี้เองคือโปรแกรมภาษาจาวา

JVM ทำหน้าที่แปลงไบต์โค้ดไปเป็นภาษาเครื่องสำหรับระบบปฏิบัติการนั้น ๆ เช่นเมื่อโปรแกรมนำมาใช้งานในระบบปฏิบัติการวินโดวส์ ไบต์โค้ดของโปรแกรมจะถูกแปลงเป็นภาษาเครื่องสำหรับระบบปฏิบัติการวินโดวส์ผ่านทาง JVM หากไบต์โค้ดตัวเดียวกันมาใช้งานบนระบบปฏิบัติการลีนุกซ์ JVM ก็จะแปลงเป็นภาษาเครื่องสำหรับลีนุกซ์นั้นเอง ดังนั้นไบต์โค้ดมักจะถูกเรียกว่าเป็นภาษากลางที่สามารถนำไปใช้งานได้ในทุก ๆ ระบบผ่านทาง JVM

เครื่องมือในการพั ฒนาโปรแกรม บริษัทซันไมโครซิสเท็มได้แบ่งสภาพการทำงานของจาวาออกเป็น 3 รูปแบบดังนี้

1. J2SE (Java 2 Standard Edition) ซึ่งใช้เป็นการใช้งานภาษาจาวาบนเครื่องคอมพิวเตอร์ปกติ หรือโดยทั่วไปไม่ว่าจะเป็นการใช้งานแบบเครื่องเดียว (Standalone) หรือเรียกว่าจาวาแอปพลิเคชั่ น (Java Application) การใช้ผ่านระบบเน็ตเวิร์ค (Network) การใช้งานผ่านเว็บที่เรียกว่าจาวาแอปเพล็ต (Java Applet) หากพัฒนาโปรแกรมในแบบ J2SE ต้องมีเครื่องมือ J2SDK ติดตั้งอยู่บนเครื่อง ดาวน์โหลด ได้จาก http://java.sun.com/j2se
2. J2EE (Java 2 Enterprise Edition) สำหรับใช้งานในรูปแบบการทำงานแบบไคลแอ็นต์/เซิร์ฟเวอร์ ซึ่งโปรแกรมภาษาจาวานี้จะทำงานบนเครื่องระดับเซิร์ฟเวอร์แทน เพื่อตอบสนองการทำงานร่วมกับโปรแกรมอื่น ๆ ที่ทำงานบนเครื่องระดับไคลแอนท์ โปรแกรมภาษาจาวาที่ทำงานบนเซิร์ฟเวอร์ นี้มีรูปแบบการทำงานเฉพาะ เรียกว่า จาวาเซิร์ฟเล็ต (Java Servlet) และเจเอสพี (JSP, Java Server Page) หากพัฒนาโปรแกรมในแบบ J2EE ต้องมีเครื่องมือ J2SDK ติดตั้งอยู่บนเครื่อง พร้อมกับกลุ่มคลาสที่รองรับการทำงานของ J2EE ดาวน์โหลดเพิ่มเติมได้จาก http://java.sun.com/j2ee
3. J2ME (Java 2 Micro Edition) เป็นรูปแบบการทำงานบนอุปกรณ์ขนาดเล็ก เช่น PDA, โทรศัพท์มือถือ เครื่องใช้ไฟฟ้า ซึ่งมีข้อจำกัดทางด้านทรัพยากรการทำงาน เช่น หน่วย

ประมวลผล หน่วยความจำ จอแสดงผล แต่โปรแกรมภาษาจาวาสามารถเข้าไปทำงานบนอุปกรณ์เหล่านั้นได้ โปรแกรมในกลุ่มนี้ บางครั้งเรียกว่าจาวาเม็ดเล็ ต (Java Medlet) หากพัฒนาโปรแกรมในแบบ J2ME ต้องมีเครื่องมือ J2SDK ติดตั้งอยู่บนเครื่อง พร้อมกลุ่มคลาสที่รองรับการทำงานของ J2ME และโปรแกรมจำลองการทำงานของอุปกรณ์ขนาดเล็กบนเครื่องคอมพิวเตอร์ที่เรียกว่าอีมูเลเตอร์ (Emulator) ดาวน์โหลดเพิ่มเติมได้จาก http://java.sun.com/j2me



รูปที่ 2.1 แสดงโครงสร้างของ Java Technologies

เทคโนโลยีจาวาที่ใช้ในงานวิจัย

1. JDBC (Java Database Connectivity)

JDBC คือ API (Application Programming Interface) หรือ library ในจาวาที่ใช้สำหรับติดต่อกับฐานข้อมูลที่เป็นแบบ relational อย่างเช่น MS SQL, Oracle, MySQL, DB2, Informix เป็นต้น JDBC จะช่วยให้สามารถเพิ่ม แก้ไข ลบ หรือเรียกดูข้อมูลที่เก็บไว้ในฐานข้อมูลจากโปรแกรมจาวาที่เขียนขึ้นได้

1.1 โครงสร้างของ JDBC

JDBC ทำงานโดยใช้ชุดของ Java Interface ซึ่งแต่ละ interface จะถูก implement โดยผู้ผลิตฐานข้อมูลแต่ละรายแตกต่างกันไป ชุดของ class ที่ implement JDBC interface สำหรับฐานข้อมูลแต่ละตัวนี้เรียกว่า JDBC Driver ในการสร้าง application นั้นไม่ต้องคำนึงถึงว่า class เหล่านี้ถูก implement อย่างไร จุดมุ่งหมายของ JDBC คือ ซ่อนความ

แตกต่างเฉพาะตัวของแต่ละฐานข้อมูลไว้ทำให้เราคำนึงถึงเฉพาะส่วนของ application เท่านั้น

1.2 JDBC Drivers

ในการติดต่อกับฐานข้อมูลโดยใช้ JDBC นั้นจำเป็นจะต้องมี JDBC Driver ซึ่งจะทำหน้าที่เป็นตัวกลางระหว่าง application และฐานข้อมูล ซึ่งทางบริษัทซันได้แบ่ง JDBC Driver ออกเป็น 4 ชนิด ซึ่งผู้ผลิตฐานข้อมูลแต่ละราย อาจจะจัดเตรียม JDBC Driver ไว้ไม่ครบทั้ง 4 ชนิดก็เป็นได้

1.2.1 JDBC Bridge Driver

Driver ประเภทนี้ ใช้เทคโนโลยี bridge เชื่อมต่อ client กับ third - party API เช่น Oracle Database Connectivity (ODBC) . JDBC - ODBC bridge ของ sun ก็เป็นอีกตัวอย่างหนึ่งของ driver ประเภทนี้

1.2.2 Native API

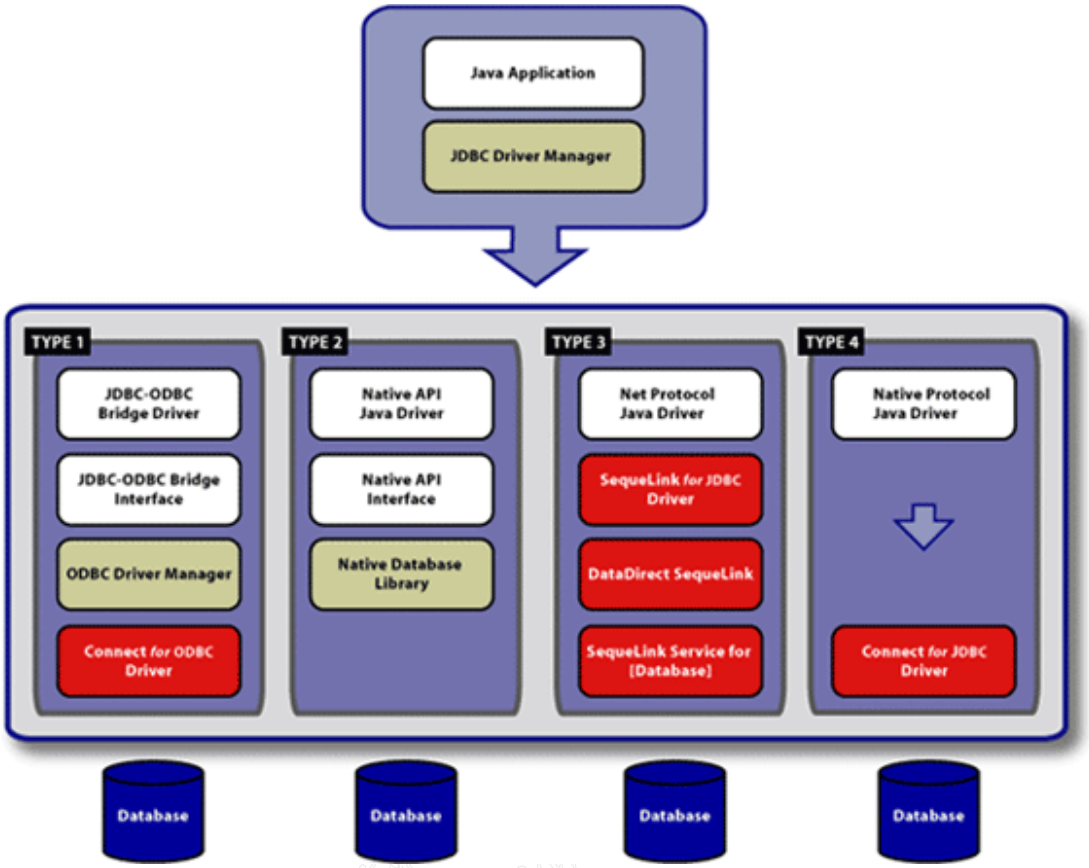
Driver แบบนี้จะใช้ java class ห่อหุ้ม native API ไว้ หมายความว่า ใช้ java code เรียกใช้ method ของ native c เหนือ c++ ซึ่งผู้ผลิตฐานข้อมูลแต่ละรายได้จัดเตรียมไว้ให้

1.2.3 Network Protocol

Driver ประเภทนี้ จะใช้ network protocol ติดต่อไปยัง middle-tier server ซึ่งจะเป็นผู้ติดต่อไปยังฐานข้อมูลอีกต่อหนึ่ง

1.2.4 Native Protocol

เป็น 100 % pure java driver ซึ่งจะติดต่อโดยตรงกับฐานข้อมูล



รูปที่ 2.2 แสดงประเภทของ JDBC

2. XML

XML เป็นเทคโนโลยีที่นิยมกันแพร่หลายในขณะนี้ เพราะว่าสามารถเก็บข้อมูลในรูปแบบของ text file ที่มีการนิยามความหมายไว้ ทำให้ง่ายในการอ่าน, การจัดการ และการสร้างขึ้นมาใหม่ตามเงื่อนไขที่ได้กำหนดตกลงเอาไว้ ซึ่ง XML ใช้สำหรับอธิบายข้อมูล ในขณะที่ HTML ใช้ในการจัดรูปแบบข้อมูลเพื่อนำไปแสดงผล และ XML สามารถส่งผ่านข้อมูล cross plateform เพราะ รูปแบบของ XML เป็นลักษณะ plain text ธรรมดา เหมาะสำหรับการแลกเปลี่ยนข้อมูลของระบบที่ไม่ compatible กัน

2.1 XML Syntax

XML มีลักษณะเป็น case sensitive นั่นคือตัวพิมพ์เล็กและตัวพิมพ์ใหญ่มีความแตกต่างกัน มี tag เปิดและปิด ถ้า empty tag ใช้

2.2 ตัวอย่าง XML

```
1: <?xml version="1.0" ?>
2: <! DOCTYPE TroubleTicket SYSTEM "TroubleTicket.dtd">
3: <TroubleTicket ticketID="T7426284">
4: <Description> Customer product </Description>
5: </TroubleTicket>
6: </xml>
```

งานวิจัยที่เกี่ยวข้องกับลายพิมพ์นิ้วมือ

1. ทฤษฎีการรู้จำไบโอเมตริกซ์ (Biometrics Recognition)

ไบโอเมตริกซ์ (Biometrics) มาจากภาษากรีก 2 คำ คือ “Bios” (life) หมายถึงชีวิตหรือความมีชีวิต และ “Metro” (Measure) หมายถึงการวัดหรือมาตรวัด ดังนั้น Biometrics จึงหมายความว่า วิธีการหรือเทคนิคในการตรวจสอบ, แยกแยะสิ่งมีชีวิต โดยวัดจากคุณลักษณะของสิ่งมีชีวิตนั้น ๆ แล้วนำมาเปรียบเทียบกับคุณลักษณะนั้น ๆ ที่ได้มีการบันทึกไว้ก่อนหน้านี้ในฐานข้อมูล เพื่อวัตถุประสงค์ในการแยกแยะ (Recognize)

ไบโอเมตริกซ์สามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ ๆ คือ

1. ไบโอเมตริกซ์ทางกายภาพ (Physiology Biometrics) เช่น ลายพิมพ์นิ้วมือ (Fingerprint) โครงหน้า (Face) รูม่านตา (Iris) เป็นต้น
2. ไบโอเมตริกซ์ทางพฤติกรรม (Behavioral Biometrics) เช่น เสียง (Voice) ลายเซ็นต์ (Signature) เป็นต้น

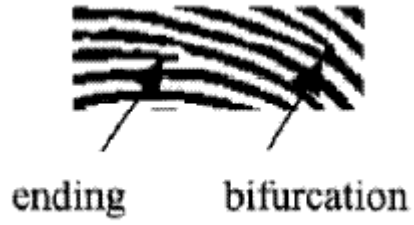
การใช้งานไบโอเมตริกซ์ แบ่งออกเป็น 2 ประเภทใหญ่ คือ

1. เพื่อยืนยันตัวบุคคล (Verification) การใช้งานในกรณีนี้ จะมีการเปรียบเทียบข้อมูลไบโอเมตริกซ์ที่เก็บได้ใหม่ ณ จุดใช้งาน กับข้อมูลของบุคคลนั้นที่ได้เคยลงทะเบียนไว้เพื่อพิสูจน์ว่าบุคคลนั้น ๆ เป็นตัวจริง เป็นการเปรียบเทียบแบบหนึ่งต่อหนึ่ง
2. เพื่อระบุตัวบุคคล (Identification) กรณีนี้จะต้องเทียบข้อมูลไบโอเมตริกซ์ที่เก็บใหม่ ณ จุดใช้งาน กับข้อมูลไบโอเมตริกซ์ทั้งหมดที่มีอยู่ในฐานข้อมูล เพื่อพิสูจน์ว่าบุคคลที่เป็นเจ้าของข้อมูลที่เก็บ ณ จุดใช้งานนั้นเป็นใครในฐานข้อมูลถือว่าเป็นการเปรียบเทียบหนึ่งต่อหลาย ๆ ต้นแบบ

2. การรู้จำลายนิ้วมือ (Fingerprint Recognition)

ลายนิ้วมือ (Fingerprint) ในแต่ละบุคคลนั้นจะมีรูปแบบเฉพาะตัว และไม่มีการเปลี่ยนแปลงรูปแบบตามกาลเวลา (แต่อาจเปลี่ยนขนาดได้) ซึ่งปลอมแปลงได้ยาก จึงมีความน่าเชื่อถือสูง

ลายนิ้วมือของแต่ละคน ประกอบด้วยสันและร่อง ซึ่งลักษณะนี้มีคุณลักษณะที่สำคัญ 2 ประการ คือ ไม่มีการเปลี่ยนแปลง รูปแบบตามกาลเวลา และมีรูปแบบเฉพาะในแต่ละบุคคล จุดลักษณะสำคัญบนลายนิ้วมือหรือที่เรียกว่า “ไมนูเทีย” (Minutia) ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 แสดงลักษณะลายนิ้วมือ

ไมนูเทียในลายพิมพ์นิ้วมือเป็นสิ่งที่สำคัญ ที่จะนำไปใช้ในการพิสูจน์ว่าเป็นลายพิมพ์นิ้วมือของบุคคลใด โดยที่ระบบการพิสูจน์ลายพิมพ์นิ้วมือประกอบด้วยส่วนที่สำคัญดังนี้

Image Capture จะเป็นส่วนของการรับลายพิมพ์นิ้วมือจากเครื่องสแกนลายนิ้วมือ

Image Enhancement จะเป็นการปรับปรุงคุณภาพของลายพิมพ์นิ้วมือให้มีความชัดเจนมากขึ้น

Feature Extraction จะเป็นการหาไมนูเทีย

Verification เป็นส่วนของการพิสูจน์ว่าลายพิมพ์นิ้วมือนี้เป็นลายพิมพ์นิ้วนี้เป็นลายพิมพ์เดียวกันกับลายพิมพ์นิ้วมือที่ต้องการพิสูจน์หรือไม่