รายการอ้างอิง

[1] OMG. The Common Object Request Broker: Architecture and Specification. Revision 2.2. (n.p.): 1998.

[2] Pope, A. The CORBA Reference Guide: Understanding The Common Object Request Broker Architecture. U.S.A: Addison Wesley Longman, 1998.

[3] Meyer, B. et al. Enabling Internetworking between Heterogeneous Distributed Platforms. In Distributed Platforms. (n.p.): Chapman and Hall, 1996.

[4] สรยุทธ อังคณานุกิจ. การออกแบบและพัฒนาคลังชนิดของบริการที่รองรับความสัมพันธ์แบบเท่าเทียมกัน. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย, 2543.

[5] Mccaffery, M., and Scott, B. The Official Visibroker for Java Handbook. U.S.A: Sams publishing, 1999.

[6] Orfali, R., and Harkey, D. Client/Server Programming with JAVA and CORBA. U.S.A: John Wiley and Sons, 1998.

[7] Pedrick, D., Weedon, J., Goldberg, J., and Blefield, E. Programming with Visibroker. U.S.A: John Wiley and Sons, 1998.

[8] ITU/ISO. Tutorial on ODP Trading Function. (n.p.): 1997.

[9] Senivongse, T. Evolution Transparency for Distributed Service Types. Ph.D. Thesis, University of Kent, UK., January 1997.

[10] OMG. Common Object Services (COS): Trading Service Specification. (n.p.): 1995.

[11] Flanagan, D. Java Examples in A Nutshell. U.S.A: O'Reilly and Associates, 1997.

[12] Guyot, J. BNF Index of Java language grammar. Available from: http://cui.unige.ch/db-research/Enseignement/analyseinfo/JAVA/BNFindex.html.

[13] Object Oriented Concept. ORBacus for C++ and Java. Available from : http://www.orbacus.com/ob.

[14] Berg, D.,Fritzinger, J. Advanced Techniques for Java Developers. U.S.A: John Wiley, 1997.

[15] Weiss, M. Data Structures and Algorithm Analysis in C. U.S.A: Addison Wesley Longman, 1996.

[16] Chan, P., Lee, R., and Kramer, D. The Java Class Libraries, Volume 1. U.S.A: Addison Wesley Longman, 1998.

ภาคผนวก

ภาคผนวก ก
การพัฒนาโปรแกรมประยุกต์บนคอร์บาด้วยซอฟต์แวร์วิสิโบรคเกอร์

การพัฒนาโปรแกรมประยุกต์ตามข้อกำหนดของคอร์บาเป็นการพัฒนาโปรแกรมบนสถาปัตยกรรมระบบกระจายเชิงวัตถุ (Distributed Object Architecture) ซึ่งผู้พัฒนาซอฟต์แวร์จะต้องทำการพัฒนาโปรแกรมประยุกต์ทั้งในฝั่งของผู้ให้และผู้รับบริการโดยใช้ออร์บของคอร์บาทำหน้าที่เป็นตัวกลางในการรับส่งข้อมูลระหว่างกลุ่มของวัตถุที่ทำงานร่วมกัน อนึ่งวัตถุต่างๆเหล่านี้อาจทำงานอยู่ภายในโปรแกรมเดียวกัน ต่างโปรแกรมกันแต่ทำงานอยู่ภายในเครื่องคอมพิวเตอร์เดียวกัน หรือทำงานอยู่ต่างเครื่องกันเลยก็ได้

ข้อกำหนดคอร์บาของโอเอ็มจีถูกออกแบบให้มีความยืดหยุ่นสูงในการพัฒนาโปรแกรมในทางปฏิบัติ โดยโปรแกรมของผู้รับบริการที่ถูกพัฒนาขึ้นด้วยภาษาคอมพิวเตอร์ภาษาหนึ่งจะสามารถเรียกใช้งานบริการของผู้ให้บริการที่ถูกพัฒนาขึ้นบนอีกแพลตฟอร์ม (Platform) หนึ่งหรืออีกภาษาหนึ่งได้ การใช้งานโปรแกรมประยุกต์บนสถาปัตยกรรมระบบคอมพิวเตอร์แบบกระจายจะไม่เกิดข้อจำกัดในเรื่องของภาษาที่ใช้ในการพัฒนาโปรแกรมตลอดจนแพลตฟอร์มที่ใช้งานโปรแกรมประยุกต์ที่จะต้องเหมือนกันอีกต่อไป คุณสมบัติที่เป็นจุดเด่นในลักษณะดังกล่าวนี้ถูกเรียกว่าความเป็นอิสระต่อภาษาและแพลตฟอร์ม (Language and Platform Independence) มีส่วนช่วยผลักดันให้โปรแกรมประยุกต์ต่างๆของคอร์บาเป็นที่นิยมใช้งานอย่างแพร่หลาย

งานวิจัยนี้เลือกใช้ซอฟต์แวร์วิสิโบรคเกอร์สำหรับภาษาจาวา (Visibroker for Java) รุ่น 3.3 ในการพัฒนาโปรแกรมประยุกต์ทั้งในฝั่งของผู้ให้บริการและผู้รับบริการตลอดจนโปรแกรมในส่วนอื่นๆอาทิเช่น โปรแกรมที่ให้บริการค้นหาและตัวดำเนินการแปลงต่างๆเป็นต้น (รายละเอียดการพัฒนาโปรแกรมต่างๆเหล่านี้อยู่ในบทที่ 5) ขั้นตอนการพัฒนาโปรแกรมประยุกต์ด้วยซอฟต์แวร์วิสิโบรคเกอร์สำหรับภาษาจาวามีขั้นตอนการทำงานตามรายละเอียดต่างๆดังต่อไปนี้

1. ผู้พัฒนาโปรแกรมในฝั่งของผู้ให้บริการจะทำการกำหนดรายละเอียดส่วนต่อประสานของบริการที่จะถูกสร้างขึ้นภายในโปรแกรมประยุกต์โดยใช้ภาษาไอดีแอล (IDL[22]) รายละเอียดของส่วนต่อประสานที่กล่าวถึงนี้จะถูกจัดเก็บลงในแฟ้มข้อมูลที่มีนามสกุลเป็น .idl อาทิเช่นตัวอย่างการทดสอบในหัวข้อที่ 5.1จะใช้แฟ้มข้อมูล account_by_id.idl ในการจัดเก็บรายละเอียดของส่วนต่อประสาน ::service_by_id::account_by_id เป็นต้น

2. ผู้พัฒนาโปรแกรมประยุกต์ในฝั่งของผู้ให้บริการทำการตกลงและจัดส่งแฟ้มข้อมูลไอดีแอลที่สร้างขึ้นตามข้อที่ 1 ไปยังผู้พัฒนาโปรแกรมประยุกต์ในฝั่งของผู้รับบริการเพื่อใช้เป็นข้อมูลในการสร้าง

---

[22] IDL ย่อมาจาก Interface Definition Language เป็นภาษาที่ถูกสร้างขึ้นโดยโอเอ็มจีเพื่อใช้ในการประกาศรายละเอียดส่วนต่อประสานของบริการต่างๆในคอร์บา

คลาสสตับ (Client Stub) สำหรับใช้ในการรับส่งข้อมูล (Data Marshaling) ระหว่างโปรแกรมของผู้รับบริการกับอินสแตนซ์จริงของบริการซึ่งทำงานอยู่ภายในโปรแกรมของผู้ให้บริการ

3. ผู้พัฒนาโปรแกรมประยุกต์ในฝั่งผู้ให้บริการทำการคอมไพล์แฟ้มข้อมูลไอดีแอลเพื่อสร้างคลาสสเคเลตัน (Server Skeleton) ขึ้นจากข้อมูลส่วนต่อประสานโดยเรียกใช้โปรแกรมไอดีแอลคอมไพเลอร์ (IDL Compiler) ซึ่งในซอฟต์แวร์วิสิโบรคเกอร์จะใช้การเรียกผ่านคำสั่ง idl2java ตัวอย่างเช่นถ้าแฟ้มข้อมูลไอดีแอลที่ใช้จัดเก็บส่วนต่อประสานของบริการมีชื่อ account_by_id.idl ผู้พัฒนาโปรแกรมจะทำการคอมไพล์แฟ้มข้อมูลไอดีแอลด้วยการเรียกใช้คำสั่ง idl2java ดังนี้

prompt> idl2java account_by_id.idl

4. ผู้พัฒนาโปรแกรมประยุกต์ในฝั่งผู้รับบริการทำการคอมไพล์แฟ้มข้อมูลไอดีแอลเพื่อสร้างคลาสสตับ (Client Stub) จากข้อมูลส่วนต่อประสานโดยเรียกใช้โปรแกรมไอดีแอลคอมไพเลอร์เช่นเดียวกันกับข้อ 3

โปรแกรมไอดีแอลคอมไพเลอร์จะทำการคอมไพล์ส่วนต่อประสานภายในแฟ้มข้อมูลไอดีแอลเพื่อจัดสร้างคลาสต่างๆเป็นภาษาจาวา อาทิเช่นคลาส _account_by_idImplBase ซึ่งเป็นคลาสสเคเลตันที่ถูกสร้างขึ้นสำหรับทำหน้าที่เป็นคลาสแม่ (Parent Class) ให้โปรแกรมประยุกต์ในฝั่งของผู้ให้บริการทำการสืบทอดคุณสมบัติจากคลาสนี้หรือคลาส _st_account_by_id ซึ่งทำงานเป็นคลาสสตับสำหรับถูกเรียกใช้งานโดยโปรแกรมของผู้รับบริการในการรับส่งข้อมูลระหว่างโปรแกรมผู้รับบริการกับอินสแตนซ์ของบริการในโปรแกรมของผู้ให้บริการผ่านทางออร์บของคอร์บา เป็นต้น

นอกจากคลาสสตับและคลาสสเคเลตันที่กล่าวถึงไปแล้วนั้น โปรแกรมไอดีแอลคอมไพเลอร์ยังได้จัดสร้างคลาสอื่นๆขึ้นเพื่อใช้อำนวยความสะดวกในการพัฒนาโปรแกรมประยุกต์อีกด้วยอาทิเช่นคลาสตัวช่วย (Helper Class) สำหรับใช้รองรับการเรียกคำสั่ง bind เมื่อโปรแกรมของผู้รับบริการต้องการค้นหาข้อมูลอ้างอิงของบริการต่างๆจากสมาร์ทเอเจนท์ เป็นต้น

5. ผู้พัฒนาโปรแกรมประยุกต์ทั้ง 2 ฝั่งทำการพัฒนาโปรแกรมในฝั่งของตนด้วยภาษาโปรแกรมที่เลือกไว้ (ไม่จำเป็นต้องเป็นภาษาเดียวกัน) ในงานวิจัยนี้จะใช้การพัฒนาโปรแกรมประยุกต์ทั้ง 2 ฝั่งด้วยภาษาจาวา จากนั้นเมื่องานพัฒนาโปรแกรมแล้วเสร็จผู้พัฒนาโปรแกรมประยุกต์จะทำการคอมไพล์โปรแกรมที่ได้ด้วยคอมไพเลอร์ของภาษานั้นๆต่อไป ตัวอย่างเช่นถ้าโปรแกรมของผู้ให้บริการถูกพัฒนาขึ้นด้วยภาษาจาวาโดยใช้ชื่อ servicebyid.java ผู้พัฒนาโปรแกรมจะต้องคอมไพล์แฟ้มข้อมูลนี้ด้วยจาวาคอมไพเลอร์โดยใช้การเรียกไปยังคำสั่ง vbjc ดังนี้

prompt> vbjc servicebyid.java

จาวาคอมไพเลอร์จะทำการแปลงโปรแกรมต้นฉบับภายในแฟ้มข้อมูลที่ระบุมาไปเป็นจาวาไบท์โค้ด (Java Bytecode) ที่พร้อมจะถูกเรียกใช้งานในจาวาเวอร์ชวลแมชชีน (Java Virtual Machine) โดยไบท์

โค้ดที่ได้จากการคอมไพล์จะถูกจัดเก็บอยู่ในแฟ้มข้อมูลที่มีนามสกุลเป็น .class อาทิเช่นแฟ้มข้อมูล servicebyid.class เป็นต้น

6. ผู้ใช้งานโปรแกรมประยุกต์ทั้ง 2 ฝั่งต้องทำการเรียกใช้สมาร์ทเอเจนท์ก่อนที่จะทำการเรียกใช้โปรแกรมประยุกต์ผ่านทางคำสั่ง vbj ดังตัวอย่างต่อไปนี้

ฝั่งผู้ให้บริการ (Server Side)

prompt> osagent

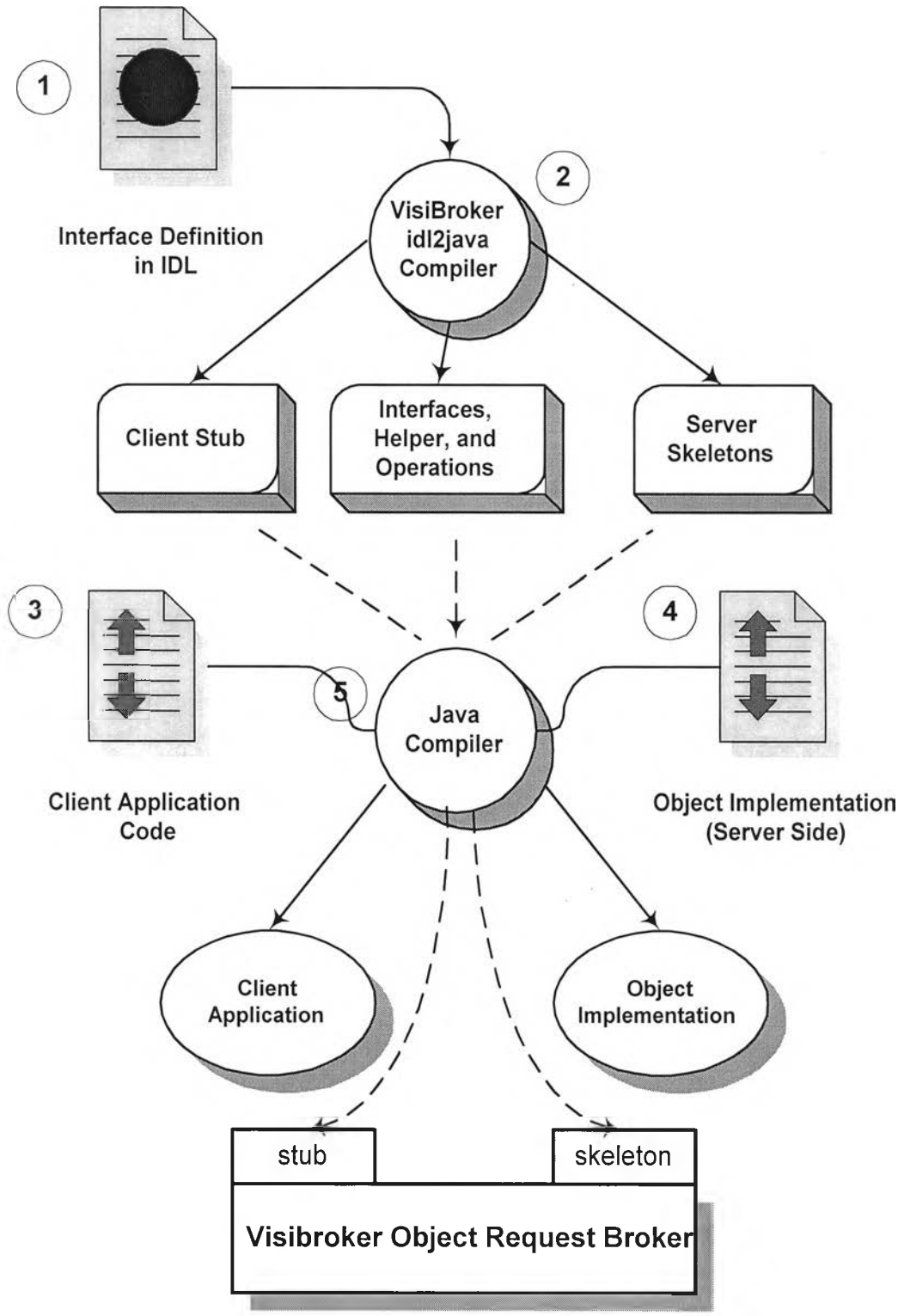
prompt> vbj servicebyid

ฝั่งผู้รับบริการ (Client Side)

prompt> osagent

prompt> vbj client

ขั้นตอนการทำงานทั้งหมดในการพัฒนาโปรแกรมประยุกต์บนคอร์บาเป็นไปตามรูปที่ ก1



รูปที่ ก1 ขั้นตอนการพัฒนาโปรแกรมประยุกต์ของคอร์บาด้วยซอฟต์แวร์วิสิโบรคเกอร์

ภาคผนวก ข
การใช้งานซอฟต์แวร์เจทีบีและจาวาซีซีในการสร้างจาวาพาร์สเซอร์

งานวิจัยนี้ใช้ซอฟต์แวร์เจทีบี (JTB[23]) และจาวาซีซี (JavaCC[24]) ในการสร้างจาวาพาร์สเซอร์ (Java Parser) สำหรับใช้ในการตรวจสอบความถูกต้องของโปรแกรมผู้รับบริการที่ถูกพัฒนาขึ้นด้วยภาษาจาวา ให้มีรูปแบบของคำสั่ง (Syntax) ที่สอดคล้องตรงกับหลักไวยากรณ์ของภาษาจาวา (Java Grammar) ก่อนที่จะนำโปรแกรมต้นฉบับนั้นไปทำการแทรกคำสั่งในขั้นตอนอื่นโดยโปรแกรมพรีโพรเซสเซอร์ต่อไป (ดังแอ็คทิวิตีไดอะแกรมในรูปที่ 3.9)

จาวาพาร์สเซอร์ที่ถูกสร้างขึ้นโดยซอฟต์แวร์เจทีบีและจาวาซีซีจะมีความสามารถในการสร้างซินแท็กซ์ทรีขึ้นในหน่วยความจำภายหลังการทำพาร์สซิงโปรแกรมต้นฉบับของผู้รับบริการ ซินแท็กซ์ทรีนี้จะถูกสร้างขึ้นเพื่อแทนองค์ประกอบต่างๆของโปรแกรมต้นฉบับด้วยโหนดต่างๆของซินแท็กซ์ทรี ส่งผลให้โปรแกรมพรีโพรเซสเซอร์สามารถจัดรูปแบบของโปรแกรมต้นฉบับขึ้นนั้นใหม่ได้โดยใช้การท่องไปยังโหนดต่างๆของซินแท็กซ์ทรีที่ถูกสร้างขึ้น นอกจากคลาสต่างๆที่ทำงานเป็นจาวาพาร์สเซอร์แล้ว งานวิจัยนี้ยังได้ใช้ซอฟต์แวร์เจทีบีและจาวาซีซีในการสร้างคลาส TreeFormatter และ TreeDumper เพื่อช่วยให้การท่องโหนดต่างๆของซินแท็กซ์ทรีง่ายขึ้น โปรแกรมต้นฉบับใหม่ที่ผ่านการจัดรูปแบบให้ง่ายต่อการวิเคราะห์แล้วจะถูกจัดเก็บลงในแฟ้มข้อมูลชั่วคราวเพื่อรอการนำไปผ่านขั้นตอนการวิเคราะห์และแทรกคำสั่งด้วยโปรแกรมพรีโพรเซสเซอร์ต่อไป

ขั้นตอนการสร้างจาวาพาร์สเซอร์และคลาส TreeFormatter ตลอดจนคลาส TreeDumper ด้วยซอฟต์แวร์เจทีบีและจาวาซีซีของงานวิจัยนี้มีขั้นตอนต่างๆดังต่อไปนี้

1. เรียกใช้ซอฟต์แวร์เจทีบีโดยส่งผ่านชื่อแฟ้มข้อมูลที่ใช้จัดเก็บหลักไวยากรณ์ภาษาจาวา (งานวิจัยนี้ใช้แฟ้มข้อมูลชื่อ Java1.1.jj ของบริษัทซัน ไมโครซิสเต็มซ์) เป็นค่าพารามิเตอร์ให้กับโปรแกรมด้วยคำสั่งต่อไปนี้

prompt> jtb –printer Java1.1.jj

ซอฟต์แวร์เจทีบีจะทำการวิเคราะห์แฟ้มข้อมูล Java1.1.jj เพื่อจัดสร้างแฟ้มข้อมูลหลักไวยากรณ์ใหม่โดยเพิ่มเติมการทำงานในส่วนของการสร้างซินแท็กซ์ทรีเข้าไปยังไวยากรณ์เดิม แฟ้มข้อมูลที่ใช้จัดเก็บหลักไวยากรณ์ใหม่นี้จะใช้ชื่อว่า jtb.out.jj โดยแฟ้มข้อมูลนี้จะถูกนำไปใช้ในการสร้างจาวาพาร์สเซอร์ด้วย

---

[23] JTB ย่อมาจาก Java Tree Builder งานวิจัยนี้ใช้ซอฟต์แวร์เจทีบีรุ่น 1.2.1 ซึ่งดาวน์โหลดมาจาก http://www.cs.purdue.edu/jtb/index.html

[24] JavaCC ย่อมาจาก Java Compiler Compiler งานวิจัยนี้ใช้ซอฟต์แวร์จาวาซีซีรุ่น 1.1 ของบริษัทซัน ไมโครซิสเต็มซ์ ซึ่งดาวน์โหลดมาจาก http://www.metamata.com/javacc

ซอฟต์แวร์จาวาซีซีในขั้นตอนต่อไป นอกจากแฟ้มข้อมูล jtb.out.jj แล้วซอฟต์แวร์เจทีบียังได้จัดสร้างไดเรคทอรีขึ้นอีก 2 ไดเรคทอรีคือ

- ไดเรคทอรี syntaxtree ที่ใช้จัดเก็บคลาสต่างๆของจาวาที่จะใช้แทนโหนดต่างๆของซินแท็กซ์ทรี อาทิเช่นคลาส ClassBodyDeclaration และคลาส MethodDeclaration เป็นต้น

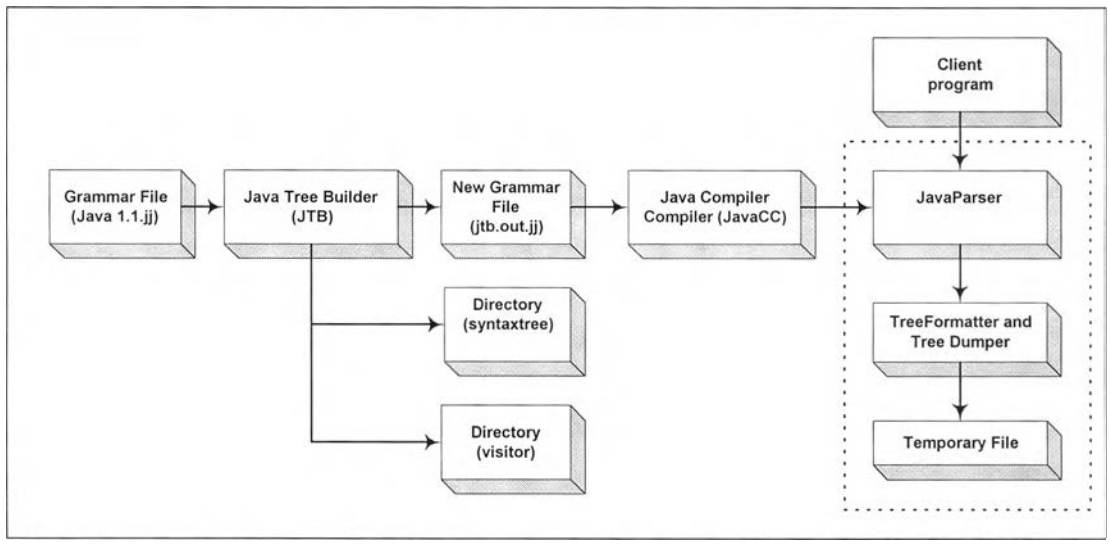
- ไดเรคทอรี visitor ที่ใช้จัดเก็บคลาสและส่วนต่อประสานต่างๆของวิสิเตอร์ อาทิเช่นคลาส DepthFirstVisitor และส่วนต่อประสาน Visitor เป็นต้น นอกจากนั้นภายในไดเรคทอรีนี้ยังใช้จัดเก็บคลาส TreeFormatter และ TreeDumper ที่กล่าวถึงไปแล้วอีกด้วย

2. เรียกใช้ซอฟต์แวร์จาวาซีซีโดยส่งผ่านแฟ้มข้อมูล jtb.out.jj (จากข้อ 1) เป็นค่าพารามิเตอร์ให้กับโปรแกรมด้วยคำสั่งต่อไปนี้

prompt> javacc jtb.out.jj

ซอฟต์แวร์จาวาซีซีจะทำการสร้างคลาสต่างๆของจาวาที่ใช้ทำหน้าที่เป็นจาวาพาร์สเซอร์อาทิเช่น คลาส JavaParser คลาส ParseException และคลาส TokenMgrError เป็นต้น ถ้าแฟ้มข้อมูลไวยากรณ์ที่ผู้ใช้งานระบุมาไม่มีข้อผิดพลาดใดๆ

ขั้นตอนการทำงานทั้งหมดในการใช้งานซอฟต์แวร์เจทีบีและจาวาซีซีในการสร้างจาวาพาร์สเซอร์เป็นไปตามรูปที่ ข1



รูปที่ ข1 ขั้นตอนการใช้งานซอฟต์แวร์เจทีบีและจาวาซีซีในการสร้างจาวาพาร์สเซอร์

โปรแกรมพรีโพรเซสเซอร์สามารถใช้งานจาวาพาร์สเซอร์และคลาส TreeFormatter ตลอดจนคลาส TreeDumper ได้ตามคำสั่งการทำงานดังต่อไปนี้

```
//สมมติให้ตัวแปร inputfile จัดเก็บชื่อแฟ้มข้อมูลของโปรแกรมผู้รับบริการที่จะถูกแทรกคำสั่ง
JavaParser parser=null;
try { parser = new JavaParser(new java.io.FileInputStream(inputfile)); }
catch (java.io.FileNotFoundException e) {
        // ไม่มีแฟ้มข้อมูลที่มีชื่อตามตัวแปร inputfile
      System.err.println("Error: "+inputfile+" file not found.");
      System.exit(1);
}
catch(Exception e){
      System.err.println(e.getMessage());
      e.printStackTrace();     System.exit(1);
}
try{
      // ทำการพาร์สซิงโปรแกรมของผู้รับบริการ
    syntaxtree.Node root = parser.CompilationUnit():
    System.out.println("Parsing source code successfully.");
    System.out.println("Abstract Syntax Tree (AST) has been created.");
      // สมมติให้แฟ้มข้อมูลชั่วคราวใช้นามสกุลเป็น .tran
    tmp=inputfile.trim()+".tran";
      // จัดรูปแบบของโปรแกรมต้นฉบับใหม่พร้อมเขียนลงในแฟ้มข้อมูลชั่วคราวโดยใช้คลาส
      // TreeFormatter และ TreeDumper
    final visitor.TreeDumper dumper = new visitor.TreeDumper(new java.io.FileOutputStream(tmp));
    root.accept(new visitor.TreeFormatter(3,0));
    dumper.resetPosition();
    root.accept(dumper);
}
catch(ParseException e){
    System.err.println("Preprocessor encountered some syntax errors.");
    System.err.println(e.getMessage());
    e.printStackTrace();
    System.exit(1);
}
```

รูปที่ ข2 การใช้งานจาวาพาร์สเซอร์ตลอดจนคลาส TreeFormatter และTreeDumper ในโปรแกรมพรีโพรเซสเซอร์

ประวัติผู้วิจัย

นายสมบูรณ์ แซ่ลิ่ม เกิดเมื่อวันที่ 9 กรกฎาคม พ.ศ. 2515 ที่อำเภอหาดใหญ่ จังหวัดสงขลา สำเร็จการศึกษาปริญญาวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า(โทรคมนาคม) จากภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ในปี พ.ศ. 2537 และได้เข้าศึกษาต่อในหลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2541 ปัจจุบันทำงานในตำแหน่งวิศวกรระบบบอาวุโส บริษัท วิทยุการบินแห่งประเทศไทย จำกัด รัฐวิสาหกิจ สังกัดกระทรวงคมนาคม