การพัฒนาบริบทประกอบเชิงกระบวนการโดยอิงสถาปัตยกรรมอิงแบบจำลอง

นายเอกวิชญ์ กุลธวัชวิชัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2553
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A DEVELOPMENT OF PROCESS-BASED COMPOSITE CONTEXTS BASED ON MODEL DRIVEN ARCHITECTURE

Mr. Ekawit Gultawatvichai

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2010

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาบริบทประกอบเชิงกระบวนการโดยอิงสถาปัตยกรรมอิงแบบจำลอง
โดย	นายเอกวิชญ์ กุลธวัชวิชัย
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(อาจารย์ ดร.ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์

(รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

..... กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.เบญจพร ลิ้มธรรมาภรณ์)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

เอกวิชญ์ กุลธวัชวิชัย : การพัฒนาบริบทประกอบเชิงกระบวนการโดยอิงสถาปัตยกรรมอิงแบบจำลอง. (A DEVELOPMENT OF PROCESS-BASED COMPOSITE CONTEXTS BASED ON MODEL DRIVEN ARCHITECTURE) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ. ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา, 90 หน้า

เทคโนโลยีคอมพิวเตอร์ได้ค่อย ๆ เข้ามาเป็นส่วนหนึ่งในชีวิตประจำวันของเรา จนทำให้เกิดแนวคิดที่เรียกว่า ระบบคอมพิวเตอร์แบบพบได้ทั่วไป กล่าวคือคอมพิวเตอร์ถูกใช้งานอยู่ทุกหนทุกแห่ง โดยที่ผู้ใช้อาจไม่ได้รู้สึกว่ากำลังใช้ระบบคอมพิวเตอร์อยู่ ซึ่งในวงการระบบคอมพิวเตอร์แบบพบได้ทั่วไปนั้นแอปพลิเคชันแบบล่วงรู้บริบทกำลังได้รับความสนใจมากขึ้นเรื่อย ๆ เนื่องจากแอปพลิเคชันเหล่านี้สามารถปรับเปลี่ยนพฤติกรรมการทำงานได้ตามบริบทโดยรอบและสภาพแวดล้อม โดยบริบทหมายถึงข้อมูลใด ๆ ที่สามารถบ่งบอกลักษณะของเอนทิตีที่สนใจได้ ซึ่งเอนทิตีอาจหมายถึงตัวบุคคล สถานที่ วัตถุสิ่งของ หรือแม้กระทั่งแอปพลิเคชัน ข้อมูลบริบทอาจเป็นได้ทั้ง (1) ค่าที่ได้รับหรือวัดได้จากฮาร์ดแวร์หรือซอฟต์แวร์ (2) ค่าที่ระบุไว้โดยตรงโดยผู้ใช้ หรือ (3) ค่าที่ได้รับมาจากบริบทอื่น ๆ เช่นจากการคำนวณ หรือจากการประกอบกันของบริบทเข้าด้วยกันเป็นบริบทประกอบ

งานวิจัยนี้นำเสนอแนวคิดของ บริบทประกอบเชิงกระบวนการ ซึ่งคือ บริบทประกอบที่สามารถสร้างได้จากบริบทย่อย ๆ ในรูปแบบของกระบวนการทำงาน ในการพัฒนาบริบทดังกล่าวยังได้อาศัยแนวคิดของสถาปัตยกรรมอิงแบบจำลอง โดยงานวิจัยนี้ได้กำหนดแบบจำลองเมตาของแบบจำลองระดับพีไอเอ็ม พีเอสเอ็ม และโค้ด รวมถึงกฎการแปลงระหว่างแบบจำลองเมตาแต่ละระดับ ซึ่งจะทำให้บริบทประกอบเชิงกระบวนที่พัฒนาได้สามารถนำไปปรับใช้กับแพลตฟอร์มต่าง ๆ ได้โดยใช้แบบจำลองระดับพีไอเอ็มแบบเดียวเท่านั้น นอกจากนั้นยังได้นำเสนอการพัฒนาบริบทประกอบเชิงกระบวนการในสองแพลตฟอร์ม ซึ่งได้แก่ โทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ และโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ โดยใช้ภาษาการแปลงเอทีแอล

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต เอกวิชญ์ กุลธวัชวิชัย

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ. ที่ปรึกษาวิทยานิพนธ์หลัก ทวิตีย์ เสนีวงศ์ ณ อยุธยา

ปีการศึกษา 2553

5170545721 : MAJOR COMPUTER ENGINEERING

KEYWORD : CONTEXT / COMPOSITE CONTEXT / WORKFLOW / UML PROFILE / MDA

EKAWIT GULTAWATVICHAI : A DEVELOPMENT OF PROCESS-BASED COMPOSITE CONTEXTS BASED ON MODEL DRIVEN ARCHITECTURE. ADVISOR : ASSOC. PROF. TWITTIE SENIVONGSE, Ph.D., 90 pp.

Computer technology has gradually become part of our daily life. This leads to a concept called ubiquitous computing or pervasive computing where computers are used around us seamlessly and, without notice, users may not realize that they are using computing systems. Within pervasive computing community, context-aware applications are gaining more interests since these applications can adapt their behavior based on surrounding context and environment. By definition, contexts refer to any information that characterizes the situation of an entity of interest. An entity can be a person, place, object, or application. Context information can be either (1) retrieved or sensed by hardware or software, (2) explicitly provided by the users, or (3) derived from other context information, e.g. by means of computation or of composition of other contexts into a composite context.

This research proposes a concept of Process-Based Composite Contexts in which a composite context can be composed from other contexts in the form of a process. To develop such a context with Model Driven Architecture, PIM-, PSM-, and code-level metamodels as well as transformation rules between the meta-models are defined. The composite context can then be ported to different platforms using only one PIM model. Also, a development of the composite context for two platforms - Windows Mobile and Android - using ATL transformation language is presented.

Department: Computer Engineering Student's Signature Ekawit Gultawatvichai

Field of Study: Computer Engineering Advisor's Signature Twittie Senivongse

Academic Year: 2010

กิตติกรรมประกาศ

ขอขอบพระคุณรองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา ที่ให้คำปรึกษา คำแนะนำ ข้อคิด และความช่วยเหลือต่าง ๆ มาโดยตลอดระยะเวลาการศึกษาและการวิจัย รวมทั้งสนับสนุนอุปกรณ์ต่าง ๆ ในการดำเนินการวิจัย ทำให้งานวิจัยนี้สำเร็จลุล่วงได้ด้วยดี

ขอขอบพระคุณอาจารย์ ดร.ยรรยง เต็งอำนวย และผู้ช่วยศาสตราจารย์ ดร.เบญจพร ลิ้มธรรมาภรณ์ คณะกรรมการสอบวิทยานิพนธ์ที่ให้ข้อชี้แนะในการปรับปรุงงานวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ทุกท่านที่ อบรม สั่งสอน ให้ความรู้ต่าง ๆ มากมายจนมีวันนี้

ขอกราบขอบพระคุณคุณพ่อ คุณแม่ ที่ให้ความรัก ความห่วงใย ทำให้มีความสุขทั้งกายและใจ และเป็นกำลังใจในการดำเนินชีวิตมาโดยตลอด

ขอขอบพระคุณญาติ ๆ ขอบคุณคุณแอม ขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ทุกคนที่เป็นกำลังใจให้กันมาเสมอมา เพื่อน ๆ พี่ ๆ ที่ห้องปฏิบัติการวิศวกรรมระบบสารสนเทศ และที่ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกคน ที่ร่วมทุกข์ร่วมสุข แบ่งปันความรู้ แง่คิดต่าง ๆ ตลอดระยะเวลาที่ดำเนินการวิจัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฏ

บทที่

1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 3

1.3 ขอบเขตของการวิจัย 4

1.4 ประโยชน์ที่คาดว่าจะได้รับ 5

1.5 วิธีดำเนินการวิจัย 5

1.6 ผลงานตีพิมพ์ 5

2 เอกสารและงานวิจัยที่เกี่ยวข้อง 7

2.1 แนวคิดและทฤษฎี 7

2.1.1 บริบท 7

2.1.2 ยูเอ็มแอล และ ยูเอ็มแอลโปรไฟล์ 7

2.1.3 เอ็มดีเอ 9

2.1.4 การพัฒนาแอปพลิเคชันโดยอิงสถาปัตยกรรมอิงแบบจำลองโดยใช้โปรแกรม Eclipse 10

หน้า

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง 12
2.2.1 งานวิจัยที่เกี่ยวข้องทางด้านการเขียนแบบจำลองสำหรับบริบท 12
2.2.2 งานวิจัยที่เกี่ยวข้องทางด้านการสร้างแอปพลิเคชันแบบล่วงรู้บริบทสำหรับอุปกรณ์เคลื่อนที่ 14
2.2.3 งานวิจัยที่เกี่ยวข้องกับระบบคอมพิวเตอร์แบบพบได้ทั่วไปและโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟน 14
3 การออกแบบ 16
3.1 แบบจำลองเมตา 17
3.1.1 แบบจำลองเมตาสำหรับแบบจำลองระดับพีไอเอ็ม 17
3.1.2 แบบจำลองเมตาสำหรับแบบจำลองระดับพีเอสเอ็ม 20
3.1.3 แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ด 22
3.2 กฎการแปลง 25
3.2.1 กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม 25
3.2.2 กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง บริบทประกอบเชิงกระบวนการระดับโค้ด 27
4 การทดสอบ 32
4.1 การพัฒนาแบบจำลองบริบทต้นแบบระดับพีไอเอ็ม 33
4.2 การพัฒนาแบบจำลองเมตา 37
4.3 การพัฒนากฎการแปลง 40
4.4 การพัฒนาแอปพลิเคชันต้นแบบ 45

4.4.1 การพัฒนาแอปพลิเคชันต้นแบบสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ 58

4.4.2 การพัฒนาแอปพลิเคชันต้นแบบสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ 62

5 บทสรุป 66

5.1 สรุปผลการวิจัย 66

5.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย 67

5.3 ข้อเสนอแนะ 68

รายการอ้างอิง 69

ภาคผนวก 74

ภาคผนวก ก คู่มือการใช้งานแอปพลิเคชัน MDA Development Assistant Tool 75

ภาคผนวก ข คู่มือการใช้งานแอปพลิเคชัน G! Movies for Windows Mobile 80

ภาคผนวก ค คู่มือการใช้งานแอปพลิเคชัน G! Movies for Android 85

ประวัติผู้เขียนวิทยานิพนธ์ 90

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ หน้า

3.1 ไวยากรณ์ของข้อความในแท็ก detail ของ Stereotype ระดับพีไอเอ็ม 19

3.2 ความสัมพันธ์ระหว่าง Stereotype ระดับพีไอเอ็ม และระดับพีเอสเอ็ม 26

3.3 ความสัมพันธ์ระหว่างส่วนของข้อความของแท็ก detail ของ Stereotype ระดับพีไอเอ็ม และแท็กของ Stereotype ระดับพีเอสเอ็ม 26

3.4 ความสัมพันธ์ระหว่าง Stereotype ระดับพีเอสเอ็ม และ element ระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ 27

3.5 ความสัมพันธ์ระหว่างแท็กของ Stereotype ระดับพีเอสเอ็ม และ attribute ของ Stereotype ระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์..... 28

3.6 ความสัมพันธ์ระหว่าง Stereotype ระดับพีเอสเอ็ม และ element ระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ 29

3.7 ความสัมพันธ์ระหว่างแท็กของ Stereotype ระดับพีเอสเอ็ม และ attribute ของ Stereotype ระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์..... 30

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปภาพที่ หน้า

2.1 การใช้ Eclipse Modeling Tools ในขั้นตอนการเตรียมไฟล์ 11

2.2 การใช้ Eclipse Modeling Tools ในขั้นตอนการพัฒนาแอปพลิเคชัน 11

3.1 โครงสร้างแบบจำลองบริบทประกอบเชิงกระบวนการตามแนวคิดเอ็มดีเอ 16

3.2 ความสัมพันธ์ระหว่างแบบจำลอง แบบจำลองเมตา และกฎการแปลง 17

3.3 ยูเอ็มแอลโปรไฟล์สำหรับบริบทประกอบเชิงกระบวนการ 18

3.4 ยูเอ็มแอลโปรไฟล์สำหรับบริบทประกอบเชิงกระบวนการสำหรับอุปกรณ์ประเภทโทรศัพท์เคลื่อนที่ 21

3.5 สกีมาของแบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ 22

3.6 สกีมาของแบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ 24

4.1 ภาพรวมของการทดสอบ 33

4.2 กระบวนการทำงานภายในบริบทประกอบ Recommended Nearby Cinema 34

4.3 แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็มของบริบทต้นแบบ 36

4.4 ไฟล์ ecore ของแบบจำลองเมตาระดับพีไอเอ็ม 38

4.5 ไฟล์ ecore ของแบบจำลองเมตาระดับพีเอสเอ็ม 39

4.6 ไฟล์ ecore ของแบบจำลองเมตาระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ 39

4.7 ไฟล์ ecore ของแบบจำลองเมตาระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ 40

4.8 ตัวอย่างส่วนหนึ่งของไฟล์ ATL ของกฎการแปลงสำหรับแปลงแบบจำลองระดับพีไอเอ็มไปยังแบบจำลองระดับพีเอสเอ็มสำหรับแอนดรอยด์ 42

หน้า

4.9 ตัวอย่างส่วนหนึ่งของไฟล์ ATL ของกฎการแปลงสำหรับแปลงแบบจำลองระดับพีเอสเอ็มไปยังแบบจำลองระดับโค้ดสำหรับแอนดรอยด์ 43

4.10 ภาพรวมของการทดสอบเมื่อใช้แอปพลิเคชันเพื่ออำนวยความสะดวกช่วยในการจัดการไฟล์ 45

4.11 หน้าจอแอปพลิเคชันส่วนจัดรูปแบบไฟล์ก่อนเข้าเครื่องประมวลผล 46

4.12 หน้าจอแอปพลิเคชันส่วนจัดรูปแบบไฟล์แบบจำลองระดับโค้ดให้เป็นโค้ด 46

4.13 โครงสร้างไฟล์แบบจำลองบริบทระดับพีไอเอ็มที่ถูกจัดรูปแบบก่อนเข้าเครื่องประมวลผล 47

4.14 ตัวอย่างการปรับแก้โค้ดส่วน import ของโค้ดในภาษาจาวาด้วยโปรแกรม Eclipse 48

4.15 หน้าจอแสดงผลรายชื่อโรงภาพยนตร์แบบแผนที่ 50

4.16 หน้าจอแสดงผลรายชื่อโรงภาพยนตร์แบบรายการ 51

4.17 หน้าจอแสดงรายการภาพยนตร์ที่ฉายอยู่ของแต่ละโรงภาพยนตร์ 51

4.18 หน้าจอสำหรับระบุความพึงใจของผู้ใช้ 52

4.19 ตัวอย่างหน้าเว็บ Google Movies โดยใช้คำค้น “Bangkok” 53

4.20 ส่วนประกอบของระบบเว็บเซอร์วิซ “G! Movies Webservice” 54

4.21 ตัวอย่างของ Request Object สำหรับใช้เรียกเว็บเซอร์วิซที่พัฒนาขึ้น 54

4.22 ตัวอย่างของ Response Object ที่ได้จากเว็บเซอร์วิซที่พัฒนาขึ้น 55

4.23 แผนภาพอีอาร์ของฐานข้อมูลของระบบเว็บเซอร์วิซ “G! Movies Webservice” 56

4.24 หน้าจอระหว่างการทำงานของแอปพลิเคชันสำหรับดึงข้อมูลมาเก็บไว้ในฐานข้อมูล 57

4.25 หน้าจอการตั้งค่าของแอปพลิเคชันสำหรับดึงข้อมูลมาเก็บไว้ในฐานข้อมูล 57

4.26 คลาสเครื่องประมวลผลภาษาสำหรับบริบทเชิงกระบวนการ 58

4.27 แผนภาพภายในเอพีไอ 59

หน้า

4.28 ตัวอย่างโค้ดการเรียกใช้งานคลาสเครื่องประมวลผล 59

4.29 หน้าจอแบบแผนที่ของแอปพลิเคชันต้นแบบบนวินโดวส์โมบายล์ 60

4.30 หน้าจอแบบรายการของแอปพลิเคชันต้นแบบบนวินโดวส์โมบายล์..... 60

4.31 หน้าจอแสดงรายการภาพยนตร์ที่ฉายอยู่ของแต่ละโรงภาพยนตร์ของแอปพลิเคชันต้นแบบบนวินโดวส์โมบายล์..... 61

4.32 หน้าจอตั้งค่าของแอปพลิเคชันต้นแบบบนวินโดวส์โมบายล์ 61

4.33 หน้าจอแบบแผนที่ของแอปพลิเคชันต้นแบบบนแอนดรอยด์..... 63

4.34 หน้าจอแบบรายการของแอปพลิเคชันต้นแบบบนแอนดรอยด์ 63

4.35 หน้าจอแสดงรายการภาพยนตร์ที่ฉายอยู่ของแต่ละโรงภาพยนตร์ของแอปพลิเคชันต้นแบบบนแอนดรอยด์..... 64

4.36 หน้าจอตั้งค่าของแอปพลิเคชันต้นแบบบนแอนดรอยด์ 64

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีต่าง ๆ ได้เข้ามามีบทบาทแทบทุกส่วนในการดำรงชีวิตของมนุษย์ จนทำให้เกิดแนวคิดเกี่ยวกับระบบคอมพิวเตอร์แบบพบได้ทั่วไป (Ubiquitous/Pervasive Computing) กล่าวคือการทำงานของระบบคอมพิวเตอร์จะกระจัดกระจายและพบได้ทั่วไปในชีวิตประจำวัน โดยที่ผู้ใช้อาจจะไม่รู้ตัวว่า ณ ขณะนั้นได้ใช้งานระบบคอมพิวเตอร์อยู่ และนอกจากนี้ยังมักสามารถปรับหรือเสนอพฤติกรรมการทำงานได้อย่างอัตโนมัติตามบริบท (Context) หรือสถานะของสภาพแวดล้อมในการใช้งาน ระบบคอมพิวเตอร์เช่นนี้ใช้หลักการของการคำนวณแบบล่วงรู้บริบท (Context-Aware Computing) [1] และได้รับความสนใจเป็นอย่างมากทั้งในแวดวงวิจัยและการนำไปใช้จริงในชีวิตประจำวัน

การคำนวณแบบล่วงรู้บริบทโดยทั่วไปจะกล่าวถึงองค์ประกอบสามส่วน ได้แก่

1) บริบท
2) ฮาร์ดแวร์หรือซอฟต์แวร์ที่สามารถให้ค่าบริบท
3) ระบบคอมพิวเตอร์หรือแอปพลิเคชันที่นำค่าบริบทไปใช้

ทั้งนี้บริบทหมายถึงข้อมูลใด ๆ ที่บ่งบอกถึงลักษณะสถานการณ์ของเอนทิตี (Entity) ซึ่งเอนทิตีอาจเป็นตัวบุคคลหรือวัตถุที่เกี่ยวข้องในปฏิสัมพันธ์ระหว่างผู้ใช้กับแอปพลิเคชัน และยังหมายรวมถึงตัวผู้ใช้กับแอปพลิเคชันเองด้วย [2] ตัวอย่างเช่น หากบริบทคือตำแหน่งที่อยู่ของผู้ใช้โทรศัพท์มือถือ และ ค่าบริบทนี้ผู้ให้บริการโทรศัพท์มือถือสามารถทราบได้จากตำแหน่งของเครือข่ายโทรศัพท์มือถือที่ผู้ใช้ปรากฏตัวอยู่ ผู้ให้บริการโทรศัพท์มือถือสามารถพัฒนาแอปพลิเคชันแจ้งข่าวสารผ่านโทรศัพท์มือถือที่สามารถล่วงรู้บริบทตำแหน่งที่อยู่ของผู้ใช้ และแจ้งข่าวสารที่เกี่ยวข้องกับบริเวณตำแหน่งของเครือข่ายโทรศัพท์มือถือนั้นให้ผู้ใช้ทราบเมื่อตรวจพบว่าผู้ใช้เดินทางเข้าสู่บริเวณ เช่น การแจ้งข่าวสารของสถานที่ท่องเที่ยวในบริเวณดังกล่าว บริบทนั้นมีความหลากหลาย ขึ้นอยู่กับว่าแอปพลิเคชันที่พัฒนาขึ้นจะให้ความสนใจกับข้อมูลสถานะหรือสภาพแวดล้อมตัวใด ตัวอย่างบริบทอื่น ๆ เช่น วันเวลา การปรากฏตัวของผู้ใช้ ความชอบ กิจกรรมที่ผู้ใช้ทำอยู่ บุคคลอื่นที่อยู่ใกล้เคียง เป็นต้น

งานวิจัยโดยมากจะพิจารณาบริบทเป็นคู่ของชื่อบริบทและค่าของบริบท (Name-Value Pair) โดยมีการแบ่งประเภทไว้หลายแง่มุม โดยหากมองในแง่มุมความซับซ้อน ค่าบริบทต่าง ๆ จะสามารถนำมาพิจารณาร่วมกันเพื่อกำหนดค่าบริบทประเภทอื่นซึ่งมีความซับซ้อนมากขึ้น [1] งานวิจัยของ Sheng และ Benatallah [3] และงานวิจัยของ Han และคณะ [4] ได้แบ่งประเภทของบริบทโดยพิจารณาจากความซับซ้อนออกเป็น

1) บริบทอะตอมมิก (Atomic Context) หรือบริบทอย่างง่าย (Simple Context) ซึ่งให้ค่าพื้นฐาน เช่น บริบทอุณหภูมิ บริบทโอกาสเกิดฝนตก
2) บริบทประกอบ (Composite Context) ซึ่งได้จากการพิจารณาค่าบริบทอะตอมมิก/บริบทอย่างง่าย หรือค่าบริบทประกอบอื่น ๆ หลายประเภทประกอบกัน เพื่อตีความได้เป็นบริบทระดับสูงที่ซับซ้อนขึ้น เช่น บริบทสภาพอากาศเลวร้าย ได้จากการพิจารณาค่าบริบทอุณหภูมิและบริบทโอกาสเกิดฝนตกประกอบกัน

นอกจากนั้นบริบทยังสามารถแบ่งประเภทตามการได้มาของบริบทได้ โดยงานวิจัยของ Mostéfaoui และคณะ [1] ได้แบ่งประเภทของบริบทโดยพิจารณาจากการได้มา ดังนี้

1) บริบทที่สามารถวัดค่าได้โดยตรง (Sensed Context) ซึ่งอ่านค่าได้จากฮาร์ดแวร์หรือซอฟต์แวร์เซ็นเซอร์โดยอัตโนมัติ เช่น บริบทอุณหภูมิ บริบทแรงดัน บริบทระดับความสว่าง และบริบทระดับเสียง
2) บริบทที่ได้รับมา (Derived Context) ซึ่งใช้การประเมิน คำนวณ หรือแปลความโดยอัตโนมัติจากค่าที่วัดได้ เช่น บริบทวัน-เวลา
3) บริบทที่ได้จากการระบุค่าไว้โดยตรง (Explicitly Provided Context) เช่น บริบทความพึงใจ (Preference) ซึ่งผู้ใช้อาจระบุไว้เป็นข้อมูลเกี่ยวกับตัวผู้ใช้เองอยู่ในอุปกรณ์เคลื่อนที่

โทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟน (Smart Phone) เป็นระบบคอมพิวเตอร์แบบพบได้ทั่วไประบบแรกที่ถูกนำมาใช้ในชีวิตประจำวัน [5, 6, 7] โดยที่ผู้ใช้งานอุปกรณ์มักไม่รู้สึกว่า ณ ขณะนั้นได้ใช้งานระบบคอมพิวเตอร์อยู่ นอกจากนั้นอุปกรณ์ดังกล่าวมีประสิทธิภาพในการทำงานที่สูง มีความสามารถในการเชื่อมต่อกับระบบเครือข่ายความเร็วสูง เช่น ไวไฟ (Wi-Fi) หรือ สามจี (3G) มีตัวรับรู้ (Sensors) มากมาย และผู้ใช้สามารถพกติดตัวไปได้ตลอดส่งผลให้ข้อมูลภายในมีแนวโน้มที่จะถูกเปลี่ยนแปลงเพื่อให้สอดคล้องกับสถานการณ์ปัจจุบันของผู้ใช้อยู่เรื่อย ๆ ซึ่งทำให้โทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟนมีสภาพแวดล้อมที่เหมาะกับการทำงานของแอปพลิเคชันแบบล่วงรู้บริบท

จากการที่การพัฒนาแอปพลิเคชันในปัจจุบันมีแนวโน้มเข้าสู่การพัฒนาเชิงกระบวนการ (Process-Oriented Development) [8] โดยการสร้างแบบจำลองกระบวนการธุรกิจ (Business Process Model) ซึ่งแสดงกระบวนการทำงานของธุรกิจที่ต้องการ จากนั้นทำการแปลงแบบจำลองดังกล่าวไปเป็นภาษาสำหรับประมวลผลกระแสงาน (Workflow Execution Language) ซึ่งมีโครงสร้างสอดคล้องกับกระบวนการธุรกิจ แล้วประมวลผลโดยใช้เครื่องประมวลผล (Execution Engine) ที่สนับสนุนภาษาประมวลผลกระแสงานนั้น การพัฒนาในลักษณะนี้ทำได้อย่างรวดเร็ว เนื่องจากผู้ใช้หรือนักวิเคราะห์ธุรกิจ (Business Analyst) ที่มีความรู้เกี่ยวกับกระบวนการทำงานและโดเมนงานของแอปพลิเคชันสามารถสร้างแบบจำลองกระบวนการธุรกิจได้เองอย่างรวดเร็ว และอาศัยเครื่องมือแปลงภาษากระแสงานและเครื่องประมวลผลสำหรับประมวลผลแอปพลิเคชันได้อย่างสะดวก ตัวอย่างการพัฒนาในลักษณะนี้ได้แก่การพัฒนาแอปพลิเคชันโดยแปลงแบบจำลองกระบวนการธุรกิจไปเป็นภาษาดับเบิลยูเอส-บีเพล (WS-BPEL) [9] ซึ่งเป็นภาษาประมวลผลกระแสงานของเทคโนโลยีเว็บเซอร์วิซ โดยการแทนที่งาน (Task) ในกระบวนการธุรกิจด้วยการเรียกใช้เว็บเซอร์วิซ และใช้เครื่องประมวลผลบีเพล (BPEL Execution Engine) สำหรับประมวลผลกระแสงานนั้น การพัฒนาในลักษณะนี้สนับสนุนการปรับเปลี่ยนแบบจำลองกระบวนการธุรกิจรวมทั้ง การปรับเปลี่ยนเว็บเซอร์วิซภายในกระแสงานได้อย่างสะดวกรวดเร็ว ผู้วิจัยสนใจการนำหลักการพัฒนาเชิงกระบวนการมาใช้ในการสร้างข้อมูลบริบท โดยเห็นว่าบริบทต่าง ๆ สามารถนำมาประกอบเข้าด้วยกันในรูปกระแสงานในลักษณะเชิงกระบวนการ ซึ่งในที่นี้จะเรียกว่า บริบทประกอบเชิงกระบวนการ (Process-Based Composite Context) โดยที่บริบทประกอบเชิงกระบวนการนี้จะเป็นบริบทระดับสูงที่มีความซับซ้อนขึ้นและประเมินค่าบริบทประกอบได้จากการพิจารณาค่าบริบทย่อยอื่น ๆ ที่ประกอบกันอยู่ในลักษณะเชิงกระบวนการ

จากแนวคิดเกี่ยวกับการคำนวณแบบล่วงรู้บริบท และ การพัฒนาเชิงกระบวนการ งานวิจัยนี้จึงนำเสนอการสร้างแบบจำลองบริบทประกอบในรูปแบบของบริบทเชิงกระบวนการ เพื่อทำให้การพัฒนาบริบทประกอบมีประสิทธิภาพมากขึ้นโดยบริบทประกอบสามารถถูกสร้างขึ้นโดยการเชื่อมโยงการประมวลผลค่าบริบทอื่น ๆ ในลักษณะเชิงกระบวนการ และ นำแนวคิดของเอ็มดีเอหรือสถาปัตยกรรมอิงแบบจำลอง (MDA: Model Driven Architecture) มาประยุกต์เพื่อทำให้แบบจำลองบริบทประกอบที่ได้สามารถใช้ได้กับแพลตฟอร์มที่หลากหลาย โดยงานวิจัยนี้จะสนใจบริบทในโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟนเป็นหลัก

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อนำเสนอแนวคิดเกี่ยวกับบริบทประกอบเชิงกระบวนการ

1.2.2 เพื่อนำเสนอแนวทางในการนำเอ็มดีเอมาใช้ในการออกแบบและพัฒนาแบบจำลองบริบทประกอบเชิงกระบวนการสำหรับแพลตฟอร์มที่หลากหลาย

1.3 ขอบเขตของการวิจัย

1.3.1 กำหนดยูเอ็มแอลโปรไฟล์ของแบบจำลองระดับพีไอเอ็มสำหรับบริบทประกอบเชิงกระบวนการ และ ยูเอ็มแอลโปรไฟล์ของแบบจำลองระดับพีเอสเอ็มสำหรับบริบทประกอบเชิงกระบวนการสำหรับอุปกรณ์เคลื่อนที่ประเภทโทรศัพท์มือถือ โดยกำหนดเป็นส่วนขยายของแผนภาพกิจกรรม ซึ่งครอบคลุมโครงสร้างพื้นฐาน ได้แก่ Assign, Receive, Reply, Exit, Invoke, If, While, RepeatUntil, Sequence เป็นอย่างน้อย

1.3.2 กำหนดกฎการแปลงของแบบจำลองระดับพีไอเอ็มไปยังแบบจำลองระดับพีเอสเอ็ม

1.3.3 กำหนดกฎการแปลงของแบบจำลองระดับพีเอสเอ็มไปยังแบบจำลองระดับโค้ด

1.3.4 แบบจำลองระดับโค้ด นำเสนอโดยการพัฒนาตัวอย่างบนสองแพลตฟอร์ม ดังนี้

1.3.4.1 โทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ โดยใช้ภาษาสำหรับบริบทเชิงกระบวนการซึ่งผู้วิจัยได้พัฒนาขึ้นมา

1.3.4.2 โทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

1.3.5 พัฒนาแอปพลิเคชันสำหรับแปลงแบบจำลองระดับโค้ดไปยังโค้ด เพื่ออำนวยความสะดวกในการพัฒนาเอพีไอสำหรับให้ค่าบริบทประกอบเชิงกระบวนการ

1.3.6 พัฒนาภาษาสำหรับบริบทเชิงกระบวนการและเครื่องประมวลผลสำหรับวินโดวส์โมบายล์

1.3.7 ทดสอบและประเมินผลโดยการแสดงตัวอย่างเพื่อแสดงให้เห็นว่าแบบจำลองและกฎการแปลงต่างๆ ที่นำเสนอ สามารถทำงานได้อย่างถูกต้องตามความต้องการเชิงหน้าที่

1.3.8 พัฒนาไลบรารีที่ใช้สร้าง จัดเก็บ ปรับปรุง และเรียกค่าบริบทที่เป็นค่านามธรรมเพิ่มเติมสำหรับแต่ละแพลตฟอร์มที่ใช้ในการทดสอบ

1.3.9 พัฒนาแอปพลิเคชันแบบล่วงรู้บริบทต้นแบบที่มีการเรียกใช้เอพีไอที่ให้ค่าบริบทเชิงประกอบสำหรับแต่ละแพลตฟอร์มที่ใช้ในการทดสอบ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้แนวทางในการพัฒนาแบบจำลองบริบทประกอบเชิงกระบวนการ

1.4.2 ได้แนวทางในการพัฒนาแบบจำลองบริบทประกอบเชิงกระบวนการสำหรับหลายแพลตฟอร์มตามแนวคิดเอ็มดีเอ โดยใช้ยูเอ็มแอลโปรไฟล์ในระดับพีไอเอ็มและพีเอสเอ็ม และ กฎการแปลง

1.5 วิธีดำเนินการวิจัย

1.5.1 ศึกษาความรู้เกี่ยวกับยูเอ็มแอล และ ยูเอ็มแอลโปรไฟล์

1.5.2 ศึกษาความรู้เกี่ยวกับเอ็มดีเอ

1.5.3 พัฒนายูเอ็มแอลโปรไฟล์สำหรับแบบจำลองบริบทประกอบเชิงกระบวนการสำหรับหลายแพลตฟอร์มตามแนวคิดเอ็มดีเอ และ พัฒนากฎการแปลง

1.5.4 ออกแบบบริบทต้นแบบเพื่อใช้ในการทดสอบและประเมินผล

1.5.5 พัฒนาแบบจำลองบริบทประกอบเชิงกระบวนการสำหรับหลายแพลตฟอร์มตามแนวคิดเอ็มดีเอของบริบทต้นแบบ

1.5.6 พัฒนาเอพีไอสำหรับให้ค่าบริบทต้นแบบด้วยกระบวนการแบบเอ็มดีเอ

1.5.7 พัฒนาแอปพลิเคชันต้นแบบเพื่อทดสอบการใช้งานเอพีไอที่ได้พัฒนาขึ้น

1.5.8 ปรับปรุงแก้ไขงานวิจัย

1.5.9 สรุปผลงานวิจัย และ จัดทำวิทยานิพนธ์

1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์และนำเสนอในการประชุมวิชาการดังนี้

1.6.1 บทความชื่อ *"A Framework for Context-Flow Composite Contexts for Mobile Phone Applications"* [10]

1.6.1.1 ชื่อผู้แต่ง Ekawit Gultawatvichai และ Twittie Senivongse

1.6.1.2 ตีพิมพ์และนำเสนอในงานประชุมวิชาการชื่อ *The 12th National Computer Science and Engineering Conference (NCSEC2008)* ซึ่งจัดขึ้นในวันที่ 20-21 พฤศจิกายน 2551 ณ จ.ชลบุรี ประเทศไทย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

1.6.2 บทความชื่อ *"A Development of Process-Based Composite Contexts for Mobile Device Platforms Based on Model Driven Architecture"* [11]

1.6.2.1 ชื่อผู้แต่ง Ekawit Gultawatvichai และ Twittie Senivongse

1.6.2.2 ตีพิมพ์และนำเสนอในงานประชุมวิชาการชื่อ *The 8th International Joint Conference on Computer Science and Software Engineering (JCSSE2011)* ซึ่งจัดขึ้นในวันที่ 11-13 พฤษภาคม 2554 ณ จ.นครปฐม ประเทศไทย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎี

2.1.1 บริบท

คำนิยามของบริบท (Context) [12]

"บริบท คือสารสนเทศต่าง ๆ ที่สามารถนำมาบ่งบอกถึงลักษณะสถานการณ์ของเอนทิตี (Entity) ซึ่งเอนทิตีคือ บุคคล สถานที่ หรือวัตถุ ที่พิจารณาแล้วว่ามีความเกี่ยวข้องกับการปฏิสัมพันธ์ระหว่างผู้ใช้และแอปพลิเคชัน โดยเอนทิตีนั้นรวมไปถึงผู้ใช้และตัวแอปพลิเคชันเองด้วย"

ประเภทของบริบทสามารถจำแนกได้เป็น 4 ประเภทที่สำคัญคือ

1) เอกลักษณ์ (Identity) คือการกำหนดตัวระบุจำเพาะ (Unique Identifier) ให้กับเอนทิตี
2) สถานที่ เป็นข้อมูลตำแหน่งสามมิติ ซึ่งบอกทิศทางและความสูง รวมไปถึงข้อมูลที่จะบอกความสัมพันธ์ระหว่างเอนทิตี เช่น อยู่ในสถานที่เดียวกัน หรือมีระยะห่างระหว่างกัน
3) กิจกรรมหรือสถานะ บ่งบอกลักษณะอันแท้จริงของเอนทิตีที่สามารถวัดค่า (Sense) ได้ เช่น สถานที่ อาจจะบอกถึงอุณหภูมิ สภาพแสงโดยรอบ หรือความดังของเสียง ถ้าเป็นบุคคล อาจจะบอกถึงสถานะทางร่างกายและจิตใจ เช่น ความเหนื่อย หรือกิจกรรมที่ทำอยู่ เช่น อ่านหนังสือ หรือพูดคุย
4) เวลา เป็นบริบทที่จะช่วยในการบอกลักษณะของสถานการณ์ ซึ่งจะช่วยเพิ่มคุณค่าของข้อมูลที่ได้บันทึกไว้ เวลาจะถูกนำไปพิจารณากับบริบทอื่น ๆ ในรูปแบบของการบันทึกเวลา (Timestamp) หรือช่วงเวลา หรือในบางกรณีใช้เพื่อให้รู้ถึงลำดับของการเกิดเหตุการณ์ต่าง ๆ

2.1.2 ยูเอ็มแอล และ ยูเอ็มแอลโปรไฟล์

ยูเอ็มแอล (UML : Unified Modeling Language) [13, 14, 15] ถูกพัฒนาขึ้นโดยโอเอ็มจี (OMG : Object Management Group) เป็นภาษาสำหรับเขียนแบบจำลองซอฟต์แวร์ เพื่อใช้ใน

การวิเคราะห์ ออกแบบ และ พัฒนาระบบ ในยูเอ็มแอลรุ่น 2.2 ประกอบด้วยแผนภาพ (Diagram) 14 แบบ เพื่อใช้เขียนแบบจำลองของระบบ ซึ่งสามารถแบ่งได้เป็น 2 กลุ่มหลัก ๆ คือ กลุ่มแผนภาพทางด้านโครงสร้าง เช่น แผนภาพคลาสของยูเอ็มแอล (UML Class Diagram) เป็นต้น และ กลุ่มแผนภาพทางด้านพฤติกรรม เช่น แผนภาพกิจกรรมของยูเอ็มแอล (UML Activity Diagram) เป็นต้น

ยูเอ็มแอลเป็นภาษาที่มีความยืดหยุ่นมาก สามารถนำไปปรับใช้กับงานได้หลากหลายรูปแบบ แต่ในงานบางประเภทที่มีความเฉพาะเจาะจงเป็นอย่างมาก จำเป็นต้องมีการระบุรายละเอียดข้อมูลบางส่วนเพิ่มเติมขึ้นจากยูเอ็มแอลมาตรฐาน เพื่อให้สามารถอธิบายรายละเอียดและการทำงานของส่วนต่างๆ ของระบบได้ชัดเจนขึ้น จึงจำเป็นต้องมีเครื่องมือที่ใช้ในการขยายความสามารถของยูเอ็มแอลเพื่อรองรับความต้องการดังกล่าว ซึ่งในยูเอ็มแอลได้มีการจัดเตรียมเครื่องมือนี้ไว้ให้เป็นแพคเกจ (Package) หนึ่งในยูเอ็มแอลชื่อว่าโปรไฟล์ (Profile) โดยมีเครื่องมือต่าง ๆ ทั้ง Stereotype, Tagged Values และ Constraints โดยเมื่อทำการเขียนโปรไฟล์เป็นชุดเครื่องมือไว้ล่วงหน้าเพื่อใช้กับงานรูปแบบหนึ่ง ชุดเครื่องมือนั้นจะถูกเรียกว่า ยูเอ็มแอลโปรไฟล์ (UML Profile)

โอเอ็มจีได้มีการนิยามความหมายของยูเอ็มแอลโปรไฟล์ไว้ในเอกสารของยูเอ็มแอลรุ่น 1.3 ดังนี้

“ยูเอ็มแอลโปรไฟล์ คือ ชุดของ Stereotype, Tagged Values และ Constraints ที่ได้มีการกำหนดรายละเอียดไว้ล่วงหน้า เพื่อใช้ในการทำให้ยูเอ็มแอลมีความเฉพาะเจาะจง และ มีความสามารถเพิ่มขึ้นสำหรับโดเมนหรือกระบวนการที่เฉพาะเจาะจงหนึ่ง ๆ เช่น โปรไฟล์สำหรับยูนิไฟด์โปรเซส (Profile for Unified Process) โดยที่โปรไฟล์จะไม่มีการเพิ่มแนวคิดหรือหลักการพื้นฐานใหม่ ๆ แต่จะนำเสนอแนวทางสำหรับการนำยูเอ็มแอลไปประยุกต์ใช้ และ ทำให้เฉพาะเจาะจงกับสภาพแวดล้อมหรือโดเมนหนึ่ง ๆ”

ยูเอ็มแอลโปรไฟล์สามารถนำไปใช้งานได้หลากหลายสถานการณ์ ตัวอย่างเช่น

- ใช้เพื่อปรับปรุงให้ยูเอ็มแอลมีความเฉพาะเจาะจงสำหรับการนำไปใช้งานกับเทคโนโลยีหรือภาษาหนึ่ง ๆ เช่น ยูเอ็มแอลโปรไฟล์สำหรับภาษาจาวา เพื่อให้ยูเอ็มแอลมีความสอดคล้องกับภาษาจาวามากขึ้น

- ใช้เพื่อปรับปรุงให้ยูเอ็มแอลมีความเฉพาะเจาะจงสำหรับโดเมนหนึ่ง ๆ เช่น ยูเอ็มแอลโปรไฟล์สำหรับการเขียนแบบจำลองคิวโอเอส (QoS) หรือ การเขียนแบบจำลองที่ทนต่อความผิดพร่อง (Fault Tolerance)
- ใช้เพื่อปรับปรุงให้ยูเอ็มแอลมีความเฉพาะเจาะจงสำหรับโปรแกรมเฉพาะทาง เช่น ยูเอ็มแอลโปรไฟล์สำหรับโปรแกรมทางด้านภูมิศาสตร์ที่ต้องมีการระบุค่าความสัมพันธ์ระหว่างวัตถุบางอย่างซึ่งไม่มีในโปรแกรมอื่น ๆ
- ใช้เพื่อปรับปรุงให้ยูเอ็มแอลมีความเฉพาะเจาะจงกับการเขียนแบบจำลองของกระบวนการพัฒนาโปรแกรมหนึ่ง ๆ เช่น ยูเอ็มแอลโปรไฟล์สำหรับกระบวนการพัฒนาโปรแกรมในองค์กร

2.1.3 เอ็มดีเอ

แนวคิดของเอ็มดีเอหรือสถาปัตยกรรมอิงแบบจำลอง (MDA: Model Driven Architecture) [16] ถูกนำเสนอโดยโอเอ็มจี เป็นสถาปัตยกรรมที่ใช้แบบจำลองเป็นพื้นฐานในการพัฒนาระบบ ทำให้เอกสารแบบจำลองที่ถูกเขียนขึ้นในขั้นตอนของการวิเคราะห์และออกแบบได้นำมาใช้ประโยชน์ต่อในขั้นตอนของการพัฒนาโปรแกรม ซึ่งจะส่งผลให้ใช้เวลาในการพัฒนาโปรแกรมน้อยลง และ โปรแกรมและเอกสารมีความสอดคล้องกันมากขึ้นง่ายต่อการแก้ไขปรับปรุงในอนาคต

แบบจำลองที่ใช้ในการพัฒนาโปรแกรมแบบเอ็มดีเอ สามารถแบ่งได้เป็น 3 ระดับ ได้แก่

1) แบบจำลองระดับพีไอเอ็ม (PIM) เป็นแบบจำลองที่มีความเป็นนามธรรมมากที่สุด รายละเอียดในแบบจำลองนี้ส่วนใหญ่เป็นรายละเอียดเกี่ยวกับกระบวนการทางธุรกิจใช้เพื่ออธิบายรายละเอียดและขั้นตอนการทำงาน
2) แบบจำลองระดับพีเอสเอ็ม (PSM) เป็นแบบจำลองที่มีรายละเอียดเกี่ยวกับเทคโนโลยีที่เป็นเป้าหมายเพิ่มขึ้นกว่า แบบจำลองระดับพีไอเอ็ม
3) แบบจำลองระดับโปรแกรมหรือโค้ด (Code) เป็นแบบจำลองที่มีรายละเอียดตามเทคโนโลยีที่เป็นเป้าหมายที่จะพัฒนาโปรแกรม ซึ่งโค้ดถือว่าเป็นแบบจำลองระดับนี้รูปแบบหนึ่ง

ส่วนประกอบสำคัญที่ใช้ในการสร้างแบบจำลอง และแปลงแบบจำลองระหว่างแต่ละระดับตามแนวคิดเอ็มดีเอ ได้แก่

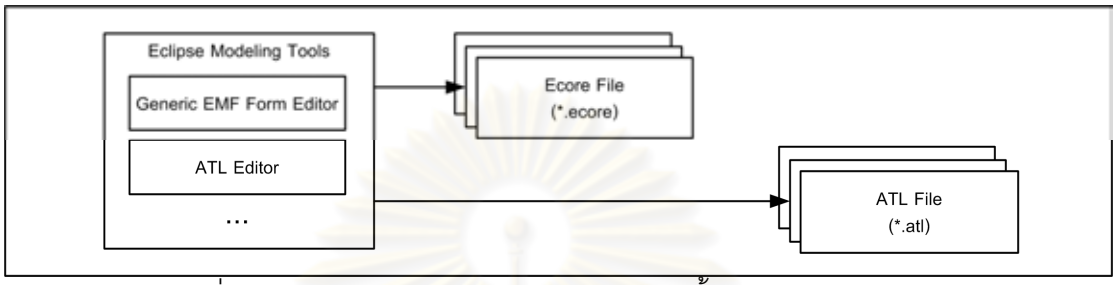
1) แบบจำลองเมตา (Metamodel) ของแบบจำลองทั้งระดับพีไอเอ็ม พีเอสเอ็ม และโค้ด ซึ่งใช้เป็นข้อกำหนดโครงร่างในการสร้างแบบจำลองระดับต่าง ๆ
2) กฎการแปลง (Transformation Rule) ที่ใช้เป็นเครื่องมือในการสร้างแบบจำลองด้วยการแปลงจากแบบจำลองของระดับหนึ่งไปสู่แบบจำลองอีกระดับหนึ่ง เช่น การแบบจำลองระดับพีไอเอ็มไปยังแบบจำลองระดับพีเอสเอ็ม เป็นต้น โดยกระบวนการแปลงสามารถทำได้ทั้งแบบอัตโนมัติ หรือ ทำโดยผู้พัฒนาเองก็ได้

2.1.4 การพัฒนาแอปพลิเคชันโดยอิงสถาปัตยกรรมอิงแบบจำลองโดยใช้โปรแกรม Eclipse

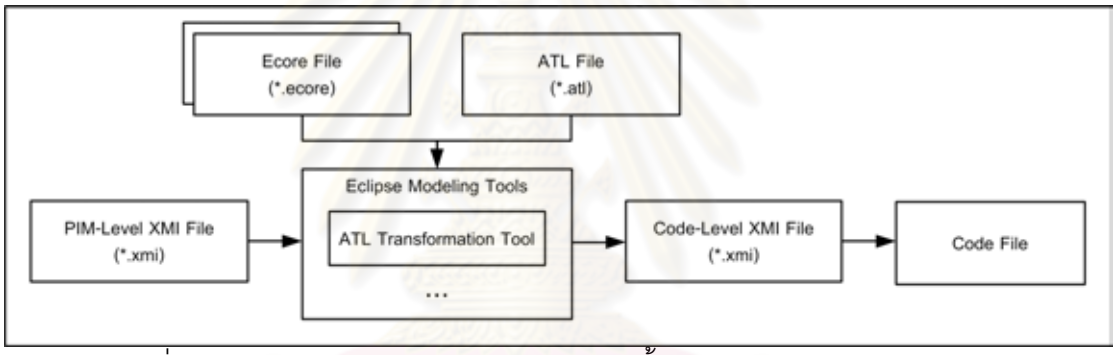
โปรแกรม Eclipse เป็นชุดเครื่องมือสำหรับใช้ในการพัฒนาแอปพลิเคชันที่ได้รับความนิยมอย่างมากชุดหนึ่ง ถูกพัฒนาขึ้นโดยบริษัทไอบีเอ็มเมื่อปี ค.ศ.2001 ปัจจุบัน Eclipse อยู่ภายใต้การดูแลของ Eclipse Foundation ซึ่งเป็นองค์กรไม่หวังผลกำไร โดย Eclipse เป็นโปรแกรมแบบเปิดเผยโค้ด (Open Source Software) ซึ่งมีชุมชนนักพัฒนาร่วมกันปรับปรุงและพัฒนาโค้ดส่วนต่าง ๆ ของโปรแกรมอย่างต่อเนื่อง Eclipse ถูกพัฒนาขึ้นโดยประกอบด้วยส่วนแกนหลักซึ่งเป็นส่วนการทำงานพื้นฐานของโปรแกรม และส่วนเสริม (Plug-In) เพื่อรองรับความสามารถเพิ่มเติม เช่น ส่วนเสริมสำหรับการพัฒนาแอปพลิเคชันในภาษาต่าง ๆ เช่น Java, C, C++, PHP, Perl, Python, Ruby ส่วนเสริมสำหรับการจัดการฐานข้อมูล ส่วนเสริมสำหรับการพัฒนาแบบจำลอง เป็นต้น ซึ่งส่วนเสริมที่กล่าวถึงนี้จะอยู่ภายใต้การดูแลของโครงการต่าง ๆ ภายในชุมชนนักพัฒนาโปรแกรม Eclipse โดยตั้งแต่ปี ค.ศ.2006 เป็นต้นมา Eclipse Foundation ตั้งเป้าไว้ว่าจะออก Eclipse เวอร์ชันใหม่ในทุกเดือนมิถุนายนของทุกปี โดยจะมีทั้งโปรแกรม Eclipse แบบพื้นฐานซึ่งผู้ใช้สามารถไปเพิ่มส่วนเสริมภายหลังได้ และโปรแกรม Eclipse แบบรวมส่วนเสริมเฉพาะทางมาแล้ว เช่น Eclipse Modeling Tools (Includes Incubating Components) [17] ซึ่งเป็นชุดโปรแกรม Eclipse แบบรวมส่วนเสริมที่มีความสามารถด้านการพัฒนาแบบจำลองจากโครงการ Eclipse Modeling Project [18] เข้าไว้ด้วยแล้ว

การพัฒนาแอปพลิเคชันโดยอิงสถาปัตยกรรมอิงแบบจำลองด้วยชุดโปรแกรม Eclipse ใช้ Eclipse ชุดแพคเกจ Eclipse Modeling Tools (Includes Incubating Components) [17] และลงส่วนเสริมที่เกี่ยวกับ ATL [19] เพิ่มเติม ซึ่งในชุดโปรแกรม Eclipse และส่วนเสริมดังกล่าวประกอบด้วยเครื่องมือจำนวนมาก ในที่นี้ใช้เครื่องมือ ดังนี้

- Generic EMF Form Editor เป็นเครื่องมือสำหรับสร้างและแก้ไขไฟล์ที่มีพื้นฐานตาม EMF (Eclipse Modeling Framework)
- ATL Editor เป็นเครื่องมือสำหรับสร้างและแก้ไขไฟล์ภาษา ATL
- ATLTransformation Tool เป็นเครื่องมือสำหรับประมวลผลไฟล์ภาษา ATL



รูปที่ 2.1 การใช้ Eclipse Modeling Tools ในขั้นตอนการเตรียมไฟล์



รูปที่ 2.2 การใช้ Eclipse Modeling Tools ในขั้นตอนการพัฒนาแอปพลิเคชัน

กระบวนการพัฒนาแอปพลิเคชันโดยอิงสถาปัตยกรรมอิงแบบจำลองด้วยชุดโปรแกรม Eclipse สามารถแสดงได้ดังรูปที่ 2.1 - รูปที่ 2.2 โดยมีรายละเอียดดังนี้

1) ขั้นตอนการเตรียมไฟล์เพื่อใช้ในกระบวนการพัฒนาแอปพลิเคชัน โดยไฟล์ที่เตรียมไว้เหล่านี้สามารถใช้สำหรับการพัฒนาแอปพลิเคชันใด ๆ บนแพลตฟอร์มที่ไฟล์เหล่านี้ออกแบบไว้ให้รองรับ แสดงเป็นแผนภาพได้ดังรูปที่ 2.1
   1.1) พัฒนาแบบจำลองเมตาในรูปแบบของไฟล์ ecore เพื่อเป็นข้อกำหนดในการเขียนแบบจำลองของแอปพลิเคชันบนแพลตฟอร์มที่ต้องการ โดยสร้างไฟล์ ecore ของแบบจำลองเมตาแต่ละแบบทั้งในระดับพีไอเอ็ม พีเอสเอ็ม และโค้ด เครื่องมือในชุดโปรแกรม Eclipse Modeling Tools ที่ใช้ ได้แก่ Generic EMF Form Editor

1.2) พัฒนากฎการแปลงในรูปแบบของไฟล์ ATL เพื่อใช้เป็นข้อกำหนดในการแปลงแบบจำลอง โดยสร้างไฟล์ ATL ของกฎการแปลงทั้งจากแบบจำลองระดับพีไอเอ็มไปยังแบบจำลองระดับพีเอสเอ็ม และแบบจำลองระดับพีเอสเอ็มไปยังแบบจำลองระดับโค้ด เครื่องมือในชุดโปรแกรม Eclipse Modeling Tools ที่ใช้ได้แก่ ATL Editor

2) ขั้นตอนการใช้งานไฟล์ที่เตรียมไว้เพื่อพัฒนาแอปพลิเคชันโดยอิงสถาปัตยกรรมอิงแบบจำลอง แสดงเป็นแผนภาพได้ดังรูปที่ 2.2

2.1) พัฒนาแบบจำลองระดับพีไอเอ็มของแอปพลิเคชันที่ต้องการในรูปแบบของไฟล์ XMI โดยอ้างอิงแบบจำลองเมตาซึ่งอยู่ในรูปแบบของไฟล์ ecore ที่ได้พัฒนาไว้ในขั้นตอนที่ 1.1 เครื่องมือในชุดโปรแกรม Eclipse Modeling Tools ได้แก่ Generic EMF Form Editor

2.2) นำไฟล์ XMI ของแบบจำลองระดับพีไอเอ็มที่พัฒนาไว้ในขั้นตอนที่ 2.1 เข้าสู่กระบวนการแปลงเพื่อสร้างแบบจำลองระดับโค้ด โดยใช้โปรแกรม Eclipse ร่วมกับไฟล์ ecore ของแบบจำลองเมตาแต่ละระดับ และไฟล์ ATL ของกฎการแปลงที่ได้พัฒนาไว้ เครื่องมือในชุดโปรแกรม Eclipse Modeling Tools ได้แก่ ATL Transformation Tool

2.3) ได้ไฟล์ผลของการแปลงซึ่งอยู่ในรูปของไฟล์ XMI ของแบบจำลองระดับโค้ดเพื่อแปลงเป็นโค้ด เพื่อใช้ในการพัฒนาแอปพลิเคชันต่อไป

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยที่เกี่ยวข้องทางด้านการเขียนแบบจำลองสำหรับบริบท

งานวิจัยที่เกี่ยวข้องทางด้านการเขียนแบบจำลองสำหรับบริบทสามารถจัดกลุ่มได้ ดังนี้

2.2.1.1 กลุ่มงานวิจัยที่เสนอการเขียนแบบจำลองสำหรับบริบทโดยปรับปรุงแบบจำลองเมตาของยูเอ็มแอล

งานวิจัย [20] ของ Henrickson และคณะ นำเสนอส่วนขยายของยูเอ็มแอล เพื่อแสดงลักษณะเฉพาะของข้อมูลบริบทซึ่งมีผลต่อแอปพลิเคชัน ได้แก่ การแยกประเภทของความสัมพันธ์ (Association) ระหว่างตัวบริบทกับคลาสของแอปพลิเคชันตามแง่มุมความซับซ้อน และแง่มุมการรวบรวมค่า และการระบุความขึ้นต่อกัน (Dependency) ระหว่างค่าบริบทต่าง ๆ ที่มีความสัมพันธ์กัน โดยแบบจำลองของบริบทในงานวิจัยนี้จะอยู่ในรูปแบบส่วนขยายของแผนภาพคลาส

งานวิจัย [3] ของ Sheng และ Benatallah นำเสนอแบบจำลองเมตาของยูเอ็มแอล โดยอิงจากแบบจำลองเมตาของแผนภาพคลาสของยูเอ็มแอลเช่นกัน เพื่อใช้สำหรับเว็บเซอร์วิซที่ล่วงรู้บริบทได้ โดยในแบบจำลองสามารถระบุรายละเอียดเพิ่มเติม เช่น บริบทที่ใช้ในแอปพลิเคชัน แหล่งที่มาของค่าข้อมูลบริบท (Context Source) และบริบทที่โอเปอเรชันของเว็บเซอร์วิซมีความขึ้นต่อกันอยู่หรือสามารถปรับพฤติกรรมตาม ทั้งนี้บริบทในแบบจำลองจะแบ่งเป็นบริบทอะตอมมิกและบริบทประกอบ โดยบริบทประกอบจะถูกกำหนดในรูปของแอกกรีเกชัน (Aggregation)

2.2.1.2 กลุ่มงานวิจัยที่เสนอการเขียนแบบจำลองสำหรับบริบทโดยนำเสนอยูเอ็มแอลโปรไฟล์

งานวิจัย [21] ของ Ayed และ Berbers นำเสนอยูเอ็มแอลโปรไฟล์สำหรับการเขียนแบบจำลองของแอปพลิเคชันที่ขึ้นกับค่าบริบท โดยในยูเอ็มแอลโปรไฟล์ที่นำเสนอประกอบด้วย 2 ส่วน

- ยูเอ็มแอลโปรไฟล์สำหรับการเขียนแบบจำลองของบริบท เป็นส่วนขยายของแผนภาพคลาสของยูเอ็มแอล ใช้เพื่อเขียนแบบจำลองของบริบทที่มีผลกระทบต่อแอปพลิเคชันโดยไม่สนใจตัวแอปพลิเคชันที่จะใช้งานบริบทนั้น ในแบบจำลองสามารถระบุรายละเอียดที่เกี่ยวข้องกับบริบท ได้แก่ กระบวนการเก็บค่าบริบท คุณภาพของค่าบริบท ความสัมพันธ์ระหว่างบริบท สถานะของบริบท
- ยูเอ็มแอลโปรไฟล์สำหรับการเขียนแบบจำลองของแอปพลิเคชัน ประกอบด้วยแบบจำลองแบ่งเป็น 2 แง่มุม ได้แก่ แง่มุมสแตติก (Static Aspect) เพื่อใช้ระบุการปรับเปลี่ยนของแอปพลิเคชันตามข้อมูลบริบททางด้านโครงสร้างและสถาปัตยกรรม และ แง่มุมไดนามิก (Dynamic Aspect) เพื่อใช้ระบุการปรับเปลี่ยนของแอปพลิเคชันตามข้อมูลบริบททางด้านพฤติกรรม

2.2.1.3 กลุ่มงานวิจัยที่เสนอการเขียนแบบจำลองสำหรับบริบทในรูปแบบของส่วนโปรแกรม

งานวิจัย [22] ของ Zimmer และ Beigl นำเสนอแบบจำลองสำหรับการออกแบบแอปพลิเคชันที่ล่วงรู้บริบทที่มีคุณสมบัติมอดูลาร์หรือมีส่วนของโปรแกรมต่าง ๆ ที่มาประกอบกัน โดยยกตัวอย่างการออกแบบแอปพลิเคชันของห้องประชุม โดยแบ่งออกเป็นส่วนย่อย ๆ เพื่อการประมวลผลเพื่อให้สามารถล่วงรู้บริบทบางชนิด เช่น ส่วนย่อยที่ล่วงรู้สถานะของปากกาไวท์บอร์ดกับส่วนย่อยที่ล่วงรู้สถานะของเก้าอี้ ซึ่งเมื่อนำบริบทส่วนย่อย ๆ มาประกอบกัน ก็จะสามารถได้ค่า

บริบทที่ซับซ้อนขึ้น เช่น เมื่อนำบริบทสถานะของปากกาไวท์บอร์ดที่ถูกใช้งานอยู่ รวมกับ บริบทสถานะของเก้าอี้ที่ถูกนั่ง ก็จะสามารถระบุได้ว่าห้องประชุมกำลังถูกใช้งานอยู่

2.2.2 งานวิจัยที่เกี่ยวข้องทางด้านการสร้างแอปพลิเคชันแบบล่วงรู้บริบทสำหรับอุปกรณ์เคลื่อนที่

งานวิจัย [23] ของ Fahy และคณะ และ งานวิจัย [24] ของ Costa และคณะ นำเสนอการสร้างแอปพลิเคชันแบบล่วงรู้บริบทสำหรับอุปกรณ์เคลื่อนที่ (Mobile Device) เช่น พีดีเอ โดยการสร้างตัวกลางบนเครื่องบริการ (Server) โดยในงานวิจัยของ Fahy และคณะ นำเสนอตัวกลางในรูปแบบของมิดเดิลแวร์ ในขณะที่งานวิจัยของ Costa และคณะ นำเสนอแพลตฟอร์มเซอร์วิซ โดยทั้งสองแนวทางมีจุดประสงค์เพื่อให้ตัวกลางทำหน้าที่ประมวลผลบริบทที่ได้รับมาทั้งจากตัวรับรู้ภายนอก และจากอุปกรณ์เคลื่อนที่ ทำให้อุปกรณ์เคลื่อนที่ดังกล่าวมีความสามารถในการประมวลบริบทที่ซับซ้อนยิ่งขึ้นได้ และ สามารถปรับเปลี่ยนขั้นตอนวิธี (Algorithm) การประมวลผลบริบทได้ง่ายขึ้นเนื่องจากไม่ยึดติดกับตัวโค้ดโปรแกรมบนอุปกรณ์เคลื่อนที่

2.2.3 งานวิจัยที่เกี่ยวข้องกับระบบคอมพิวเตอร์แบบพบได้ทั่วไปและโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟน

งานวิจัย [5] ของ Abowd และคณะ งานวิจัย [6] ของ Ballagas และคณะ และงานวิจัย [7] ของ Raento และคณะ ได้กล่าวว่าโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟนถือว่าเป็นแพลตฟอร์มแรกสำหรับระบบคอมพิวเตอร์แบบพบได้ทั่วไปที่ถูกนำมาใช้งานจริงในชีวิตประจำวัน ซึ่งถือเป็นก้าวสำคัญของการมาถึงของยุคระบบคอมพิวเตอร์แบบพบได้ทั่วไป โดยโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟนถือมีลักษณะเด่นคือมักจะมีข้อมูลที่เกี่ยวข้องกับผู้ใช้ และ มีการนำบริบทต่าง ๆ ไปคำนวณแบบล่วงรู้บริบท

งานวิจัยของ Ballagas และคณะ [6] ได้วิเคราะห์ถึงการใช้งานโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟนเป็นอุปกรณ์รับเข้า (Input Device) ในสภาพแวดล้อมระบบคอมพิวเตอร์แบบพบได้ทั่วไป ในแง่มุมของคำสั่งย่อยทางด้านกราฟฟิก ได้แก่ การเปลี่ยนตำแหน่ง (Position) การกำหนดทิศทาง (Orient) การเลือก (Select) การกำหนดเส้นทาง (Path) การกำหนดจำนวน (Quantify) การพิมพ์ข้อความ (Text) เพื่อเป็นแนวทางสำหรับการออกแบบการปฏิสัมพันธ์ของแอปพลิเคชันในโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟน

งานวิจัยของ Raento และคณะ [7] ได้นำเสนอ ContextPhone ซึ่งเป็นแพลตฟอร์มของซอฟต์แวร์บนโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟนบนแพลตฟอร์มซิมเบียน ประกอบด้วย 4 มอดูล ได้แก่ มอดูลตัวรับรู้ มอดูลการสื่อสาร มอดูลเซอร์วิซของระบบ และมอดูลแอปพลิเคชันที่ปรับแต่งได้ เพื่อเป็นแนวทางให้สามารถพัฒนาแอปพลิเคชันแบบล่วงรู้บริบทบนอุปกรณ์ดังกล่าวได้ง่ายและรวดเร็วขึ้น และได้แสดงต้นแบบแอปพลิเคชันแบบล่วงรู้บริบทที่ได้ใช้ ContextPhone เพื่ออำนวยความสะดวกในการพัฒนา

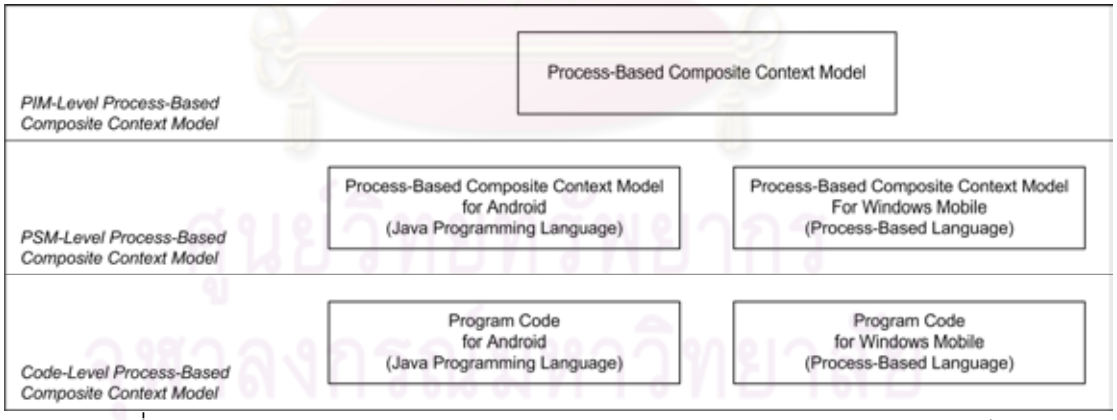
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การออกแบบ

จากการที่งานวิจัยที่เกี่ยวข้องต่าง ๆ นำเสนอการสร้างแบบจำลองสำหรับบริบทประกอบและ แอปพลิเคชันแบบล่วงรู้บริบทโดยการเพิ่มส่วนขยายหรือโปรไฟล์สำหรับแบบจำลองของคลาส แต่ยังไม่มีงานใดกล่าวถึงการสร้างแบบจำลองสำหรับบริบทประกอบในรูปแบบเชิงกระบวนการ กล่าวคือบริบทประกอบสามารถสร้างขึ้นโดยการนำบริบทอื่น ๆ มาประกอบกันเข้าและเชื่อมโยงการประมวลผลค่าบริบทในลักษณะเชิงกระบวนการ

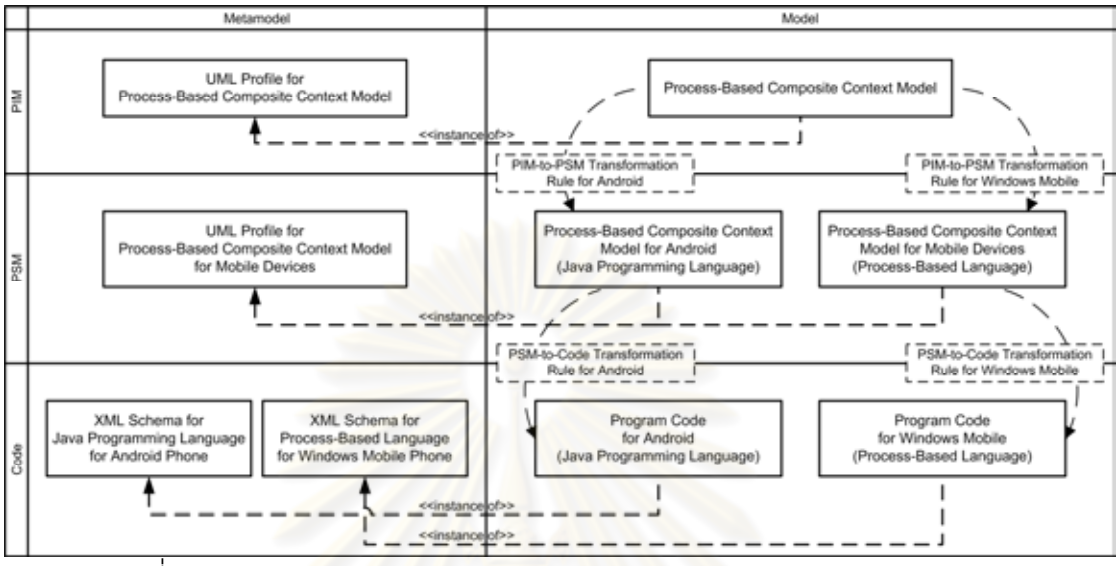
งานวิจัยนี้จึงได้นำเสนอแนวทางในการสร้างแบบจำลองบริบทประกอบในรูปแบบของบริบทเชิงกระบวนการโดยการสร้างได้อิงตามแนวคิดเอ็มดีเอด้วยการสร้างแบบจำลองบริบทประกอบตามระดับต่าง ๆ ของเอ็มดีเอซึ่งสามารถนำไปใช้ในการพัฒนาบริบทประกอบในรูปแบบของบริบทเชิงกระบวนการได้ในหลายแพลตฟอร์ม โดยในงานวิจัยนี้ได้แสดงตัวอย่างการสร้างแบบจำลองบริบทบนอุปกรณ์ประเภทโทรศัพท์เคลื่อนที่ 2 แพลตฟอร์ม ได้แก่ โทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ และโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ การสร้างแบบจำลองบริบทประกอบในรูปแบบของบริบทเชิงกระบวนการอิงตามแนวคิดเอ็มดีเอ สามารถแบ่งแบบจำลองได้เป็น 3 ระดับ ดังรูปที่ 3.1



รูปที่ 3.1 โครงสร้างแบบจำลองบริบทประกอบเชิงกระบวนการตามแนวคิดเอ็มดีเอ

นอกจากนั้นยังได้นำเสนอแนวทางในการสร้างแบบจำลองเมตา และกฎการแปลง สำหรับแพลตฟอร์มตัวอย่างทั้ง 2 แพลตฟอร์ม เพื่อใช้ในการสร้าง และแปลงแบบจำลองระหว่างแบบจำลองแต่ละระดับ จากแบบจำลองระดับพีไอเอ็ม ไปยังแบบจำลองระดับพีเอสเอ็ม และไปยังแบบจำลองระดับโค้ด ตามลำดับ โดยที่ความสัมพันธ์ระหว่างแบบจำลองทั้งสามระดับ แบบจำลอง

เมตาของแบบจำลองระดับต่างๆ และกฎการแปลงสามารถแสดงได้ ดังรูปที่ 3.2 โดยมีรายละเอียดของแบบจำลองเมตา และกฎการแปลงดังนี้



รูปที่ 3.2 ความสัมพันธ์ระหว่างแบบจำลอง แบบจำลองเมตา และกฎการแปลง

3.1 แบบจำลองเมตา

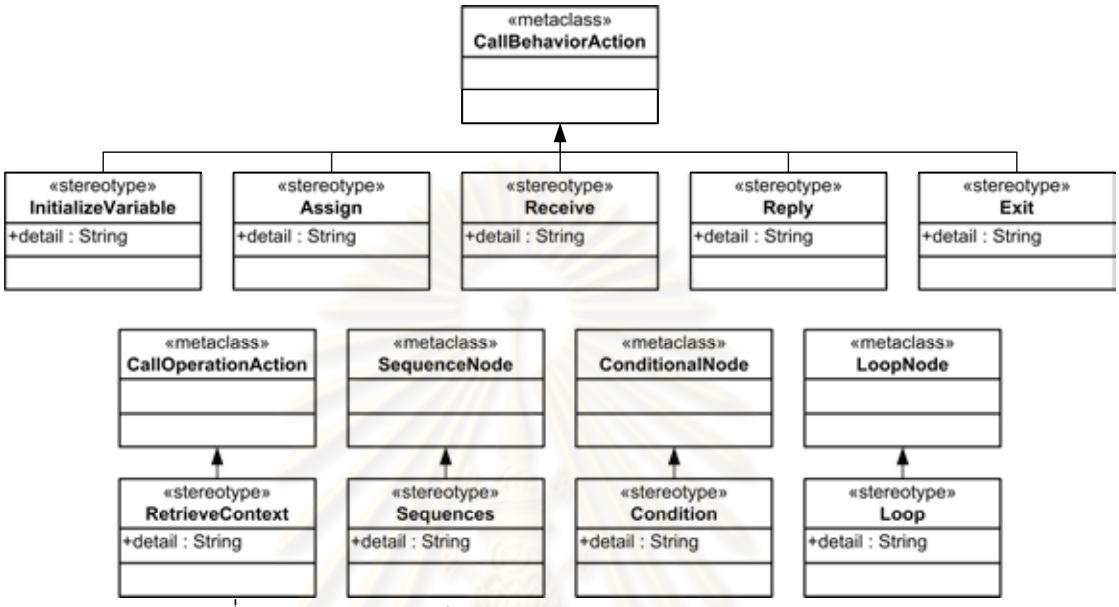
แบบจำลองเมตาเป็นข้อกำหนดในการเขียนแบบจำลองที่อธิบายถึงโครงสร้างและรายละเอียดของการสร้างแบบจำลองหนึ่งๆขึ้นมา เปรียบเสมือนไวยากรณ์ของแบบจำลอง แบบจำลองเมตาในงานวิจัยนี้ประกอบด้วย

1) แบบจำลองเมตาสำหรับแบบจำลองระดับพีไอเอ็ม (Metamodel for PIM)
2) แบบจำลองเมตาสำหรับแบบจำลองระดับพีเอสเอ็ม (Metamodel for PSM)
3) แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ด (Metamodel for Code) โดยแบ่งเป็น
   - แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์
   - แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

3.1.1 แบบจำลองเมตาสำหรับแบบจำลองระดับพีไอเอ็ม

แบบจำลองเมตาสำหรับแบบจำลองระดับพีไอเอ็มในงานวิจัยนี้นำเสนอในรูปแบบของยูเอ็มแอลโปรไฟล์ (UML Profile) โดยเป็นส่วนขยายของแผนภาพกิจกรรม (Activity Diagram) ของยูเอ็มแอลที่มีรายละเอียดที่เกี่ยวข้องกับการเขียนแบบจำลองบริบทประกอบในรูปแบบของ

บริบทเชิงกระบวนการเพิ่มเติมในรูปแบบของ Stereotype ต่าง ๆ โดยได้ถือเอาภาษาดับเบิลยูเอส-บีเพล [9] เป็นแนวทางในการออกแบบ ซึ่งสามารถเขียนเป็นยูเอ็มแอลโปรไฟล์สำหรับบริบทประกอบเชิงกระบวนการ (UML Profile for Process-Based Composite Context) ได้ดังรูปที่ 3.3



รูปที่ 3.3 ยูเอ็มแอลโปรไฟล์สำหรับบริบทประกอบเชิงกระบวนการ

Stereotype ของแบบจำลองเมตาสำหรับแบบจำลองระดับพีไอเอ็ม สามารถแบ่งได้เป็น 2 กลุ่ม ได้แก่ กลุ่มการทำงาน และ กลุ่มโครงสร้าง

Stereotype กลุ่มการทำงาน เป็นส่วนที่ใช้ระบุการทำงานของบริบทประกอบ เช่น การกำหนดค่าตัวแปร การเรียกค่าบริบทที่เป็นส่วนประกอบ เป็นต้น โดย Stereotype ในกลุ่มนี้เป็นส่วนขยายของคลาส CallBehaviorAction และ CallOperationAction ดังแสดงในรูปที่ 3.3 โดยมีรายละเอียดของ Stereotype ต่าง ๆ ดังนี้

- InitializeVariable ใช้สำหรับกำหนดค่าตัวแปรก่อนเริ่มต้นการทำงาน
- Assign ใช้สำหรับกำหนดค่าตัวแปรระหว่างการทำงาน
- Receive ใช้สำหรับรับค่าข้อมูลเข้าสู่กระบวนการทำงาน
- Reply ใช้สำหรับคืนค่าข้อมูลออกจากกระบวนการทำงาน ซึ่งในที่นี้คือบริบทประกอบในรูปแบบของบริบทเชิงกระบวนการ
- Exit ใช้สำหรับหยุดการทำงานทันที
- RetrieveContext ใช้สำหรับเรียกค่าบริบทที่เป็นส่วนประกอบ

Stereotype กลุ่มโครงสร้าง เป็นส่วนที่ใช้ระบุโครงสร้างการทำงานภายในแบบต่าง ๆ ของบริบทประกอบ เช่น การทำงานเป็นลำดับ การทำงานแบบมีเงื่อนไข เป็นต้น โดย Stereotype ในกลุ่มนี้เป็นส่วนขยายของคลาส SequenceNode, ConditionalNode, LoopNode ดังแสดงในรูปที่ 3.3 โดยมีรายละเอียดของ Stereotype ต่าง ๆ ดังนี้

- Sequences ใช้สำหรับการทำงานเป็นลำดับ
- Condition ใช้สำหรับการกำหนดเงื่อนไข
- Loop ใช้สำหรับการวนซ้ำ

ในการเขียนแท็ก (Tag) "detail" ของ Stereotype ในระดับพีไอเอ็มนี้มีการกำหนดข้อบังคับ (Constraint) ด้านไวยากรณ์ของข้อความในแท็กในรูปแบบของ EBNF-like ตามที่แสดงในตารางที่ 3.1 เพื่อที่จะทำให้การแปลงจากแบบจำลองระดับพีไอเอ็มไปยังแบบจำลองระดับพีเอสเอ็มเป็นไปได้โดยง่าย

ตารางที่ 3.1 ไวยากรณ์ของข้อความในแท็ก detail ของ Stereotype ระดับพีไอเอ็ม

PIM Stereotype with Tag Definition 'detail'	*Format Constraint*
InitializeVariable	"New" $VariableType $VariableName [$VariableValue|($VariableValue $VariableNamespace)]
Assign	"Assign" $VariableName "=" $VariableValue
Receive	"Receive" $VariableName
Reply	"Reply" $VariableName
Exit	"Exit"
RetrieveContext	"Retrieve" $VariableOutputName "From" $ContextSource ["Where" $VariableInputName {$VariableInputName}]
Sequences	"Sequences" $ActivityName {$ActivityName}
Condition	"Check" $ConditionDetail
Loop	"Loop" "While"|"RepeatUntil" $ConditionDetail

จากตารางที่ 3.1 เครื่องหมาย […] หมายถึง ส่วนของข้อความภายในเครื่องหมายนี้จะมีหรือไม่มีก็ได้ เครื่องหมาย (…) หมายถึง ส่วนของข้อความภายในเครื่องหมายนี้อยู่เป็นชุดเดียวกัน เครื่องหมาย {…} หมายถึง ส่วนของข้อความภายในเครื่องหมายนี้มีซ้ำได้ตั้งแต่ 0 ถึง ไม่จำกัด เครื่องหมาย "…" หมายถึง ส่วนของข้อความภายในเครื่องหมายนี้ต้องเขียนตามข้อความที่กำหนดไว้ภายในเครื่องหมาย เครื่องหมาย | หมายถึง ให้เลือกส่วนของข้อความที่มาก่อนหน้าหรือ

ตามหลังเครื่องหมายนี้อันใดอันหนึ่ง ในขณะที่ข้อความที่นำหน้าด้วยเครื่องหมาย $ หมายถึงข้อมูลต่าง ๆ ดังนี้

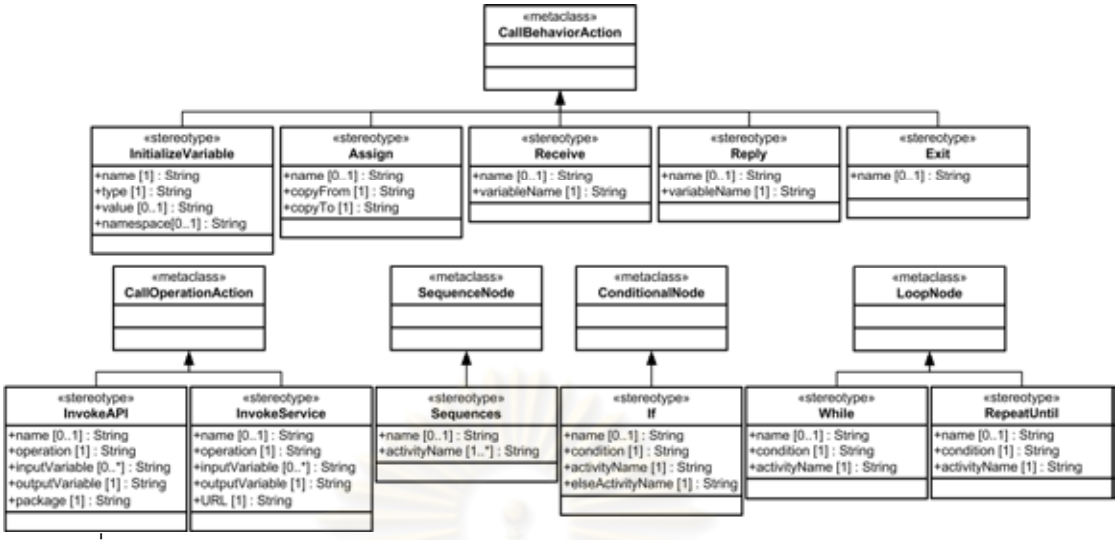
- $VariableType ชนิดของตัวแปร
- $VariableName ชื่อของตัวแปร
- $VariableValue ค่าของตัวแปร
- $VariableNamespace เนมสเปซของตัวแปร
- $VariableInputName ชื่อของตัวแปรขาเข้า
- $VariableOutputName ชื่อของตัวแปรขาออก
- $ContextSource ชื่อคำสั่งที่เรียกทำงาน
- $Activityname ชื่อกิจกรรมการทำงาน
- $ConditionDetail รายละเอียดเงื่อนไข

3.1.2 แบบจำลองเมตาสำหรับแบบจำลองระดับพีเอสเอ็ม

แบบจำลองเมตาสำหรับแบบจำลองระดับพีเอสเอ็มในงานวิจัยนี้จะนำเสนอในรูปแบบของยูเอ็มแอลโปรไฟล์ (UML Profile) โดยเป็นส่วนขยายของแผนภาพกิจกรรมของยูเอ็มแอลเช่นเดียวกับระดับพีไอเอ็ม ซึ่งมีรายละเอียดที่เกี่ยวข้องกับการเขียนแบบจำลองบริบทประกอบเชิงกระบวนการสำหรับอุปกรณ์ประเภทโทรศัพท์เคลื่อนที่เพิ่มเติมในรูปแบบของ Stereotype ต่าง ๆ โดยสามารถเขียนเป็นยูเอ็มแอลโปรไฟล์สำหรับบริบทประกอบเชิงกระบวนการสำหรับอุปกรณ์ประเภทโทรศัพท์เคลื่อนที่ (UML Profile for Process-Based Composite Context for Mobile Devices) ได้โดยมีรายละเอียดดังรูปที่ 3.4

Stereotype ในระดับพีเอสเอ็มนี้มีโครงสร้างใกล้เคียงกับ Stereotype ในระดับพีไอเอ็ม ในแต่ละ Stereotype มีแท็กเพิ่มขึ้นเพื่อระบุรายละเอียดแต่ละส่วนโดยชัดเจน และมี Stereotype บางส่วนเปลี่ยนแปลงเพื่อให้ระบุรายละเอียดได้ชัดเจนขึ้น ดังนี้

- RetrieveContext ในระดับพีไอเอ็ม สามารถแยกออกได้เป็น InvokeAPI และ InvokeService ในระดับพีเอสเอ็ม
- Condition ในระดับพีไอเอ็ม เปลี่ยนเป็น If ในระดับพีเอสเอ็ม
- Loop ในระดับพีไอเอ็ม สามารถแยกออกได้เป็น While และ RepeatUntil



รูปที่ 3.4 ยูเอ็มแอลโปรไฟล์สำหรับบริบทประกอบเชิงกระบวนการสำหรับอุปกรณ์ประเภทโทรศัพท์เคลื่อนที่

Stereotype ของแบบจำลองเมตาสำหรับแบบจำลองระดับพีเอสเอ็มมีรายละเอียดดังนี้

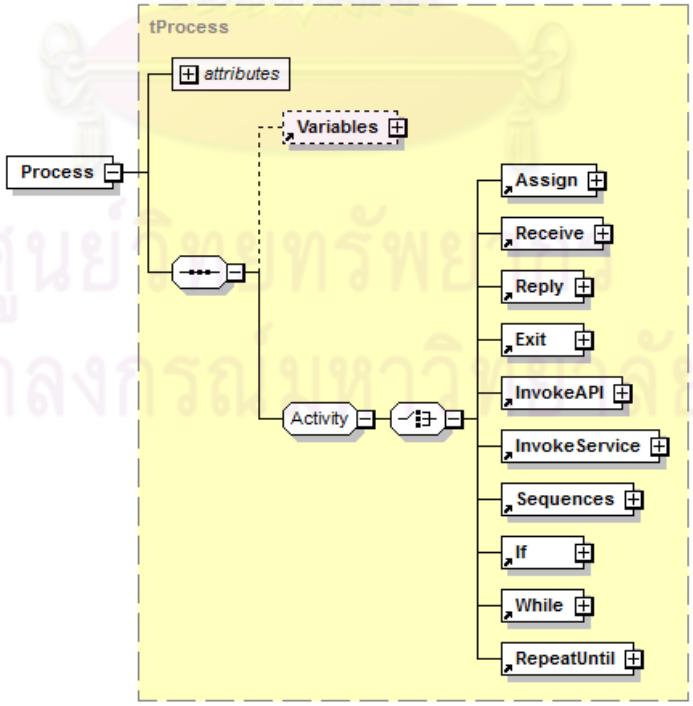
- InitializeVariable ใช้สำหรับกำหนดค่าตัวแปรก่อนเริ่มต้นการทำงาน
- Assign ใช้สำหรับกำหนดค่าตัวแปรระหว่างการทำงาน
- Receive ใช้สำหรับรับค่าข้อมูลเข้าสู่กระบวนการทำงาน
- Reply ใช้สำหรับคืนค่าข้อมูลออกจากกระบวนการทำงาน ซึ่งในที่นี้คือบริบทประกอบในรูปแบบของบริบทเชิงกระบวนการ
- Exit ใช้สำหรับหยุดการทำงานทันที
- InvokeAPI ใช้สำหรับเรียกค่าบริบทที่เป็นส่วนประกอบ จากเอพีไอที่ให้ค่าบริบทภายในอุปกรณ์
- InvokeService ใช้สำหรับเรียกค่าบริบทที่เป็นส่วนประกอบ จากผู้ให้บริการภายนอกที่ให้ค่าบริบท
- Sequences ใช้สำหรับการทำงานเป็นลำดับ
- If ใช้สำหรับการกำหนดเงื่อนไข
- While ใช้สำหรับการวนซ้ำ โดยการตรวจสอบเงื่อนไขจะทำก่อนการทำงานแต่ละรอบ
- RepeatUntil ใช้สำหรับการวนซ้ำ โดยการตรวจสอบเงื่อนไขจะทำหลังการทำงานแต่ละรอบ

3.1.3 แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ด

แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดในงานวิจัยนี้อยู่ในรูปแบบของเอ็กซ์เอ็มแอลสกีมา (XML Schema) โดยเป็นแบบจำลองที่มีรายละเอียดเหมือนกับโค้ด แตกต่างกันแค่เพียงรูปแบบการแสดงผลเท่านั้น การพัฒนาแบบจำลองเมตาในระดับโค้ดจะแยกกันระหว่างแต่ละแพลตฟอร์ม ประกอบไปด้วย *แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์* และ *แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์*

3.1.3.1 แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์

เป็นแบบจำลองเมตาเพื่อใช้เขียนแบบจำลองของ ภาษาสำหรับบริบทเชิงกระบวนการ (Process-Based Context Language) [10] ซึ่งผู้วิจัยได้พัฒนาขึ้นมา โดยภาษาดังกล่าวเป็นภาษาสำหรับประมวลผลกระแสงานมีโครงสร้างของภาษาในรูปแบบของเอ็กซ์เอ็มแอล (XML) ประมวลผลโดยการใช้ส่วนแปลภาษาและเครื่องประมวลผลที่ผู้วิจัยได้พัฒนาขึ้น สกีมาของแบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์สามารถเขียนเป็นรูปได้ดังรูปที่ 3.5



รูปที่ 3.5 สกีมาของแบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์

จากรูปที่ 3.5 สกีมามีรากคือ element “Process” ประกอบด้วย attribute และ element ซึ่ง element ประกอบด้วย element “Variables” ใช้สำหรับประกาศตัวแปรก่อนเริ่มต้นการทำงาน และ element กลุ่ม Activity โดยใน element กลุ่ม Activity มีรายละเอียดดังนี้

- Assign ใช้สำหรับกำหนดค่าตัวแปรระหว่างการทำงาน
- Receive ใช้สำหรับรับค่าข้อมูลเข้าสู่กระบวนการทำงาน
- Reply ใช้สำหรับคืนค่าข้อมูลออกจากกระบวนการทำงาน ซึ่งในที่นี้คือบริบทประกอบในรูปแบบของบริบทเชิงกระบวนการ
- Exit ใช้สำหรับหยุดการทำงานทันที
- InvokeAPI ใช้สำหรับเรียกค่าบริบทที่เป็นส่วนประกอบ จากเอพีไอที่ให้ค่าบริบทภายในอุปกรณ์
- InvokeService ใช้สำหรับเรียกค่าบริบทที่เป็นส่วนประกอบ จากผู้ให้บริการภายนอกที่ให้ค่าบริบท
- Sequences ใช้สำหรับการทำงานเป็นลำดับ
- If ใช้สำหรับการกำหนดเงื่อนไข
- While ใช้สำหรับการวนซ้ำ โดยการตรวจสอบเงื่อนไขจะทำก่อนการทำงานแต่ละรอบ
- RepeatUntil ใช้สำหรับการวนซ้ำ โดยการตรวจสอบเงื่อนไขจะทำหลังการทำงานแต่ละรอบ

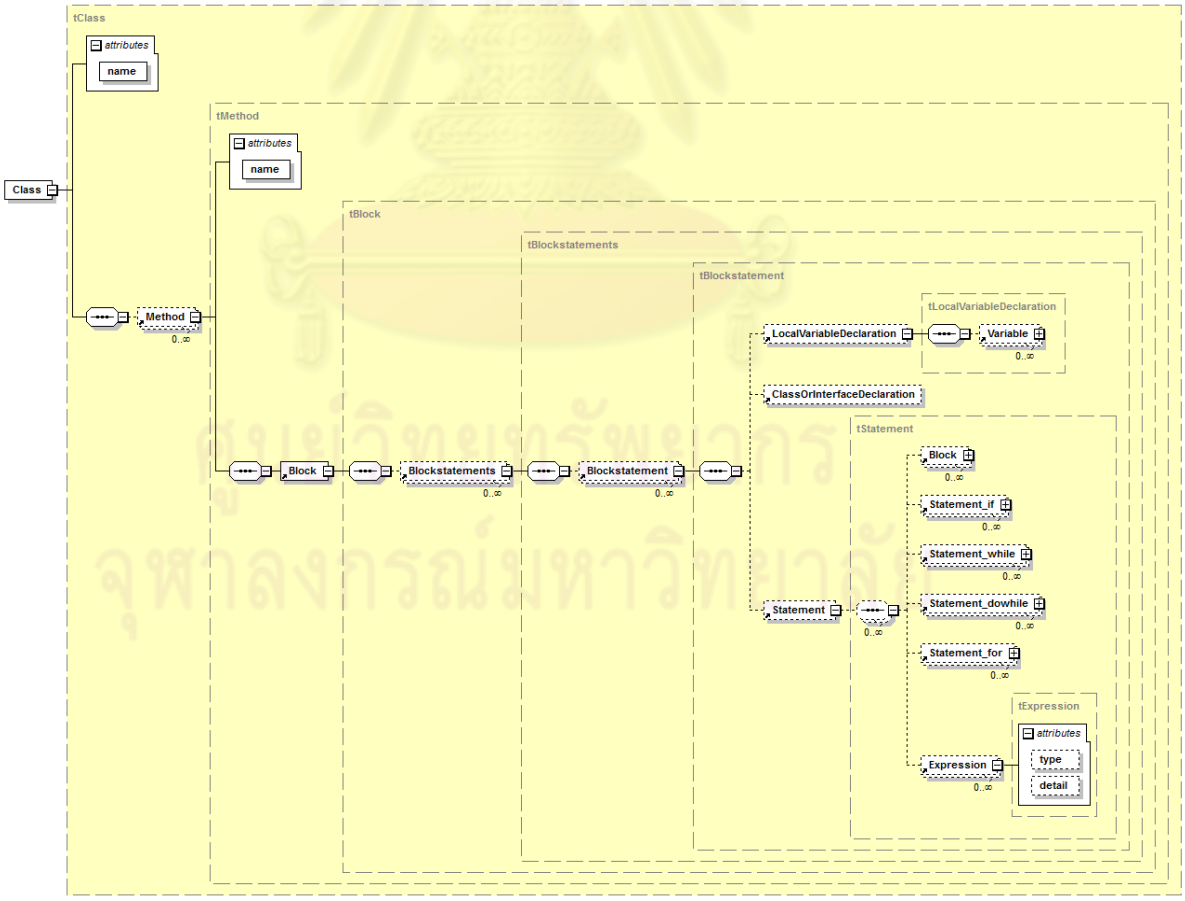
3.1.3.2 แบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

เป็นแบบจำลองเมตาเพื่อใช้เขียนแบบจำลองของภาษาจาวาซึ่งเป็นภาษาหลักในการพัฒนาแอปพลิเคชันบนแพลตฟอร์มแอนดรอยด์ โดยโครงสร้างของแบบจำลองเมตานี้อ้างอิงไวยากรณ์ของภาษาจาวา [25] แต่นำมาเฉพาะส่วนที่จำเป็นสำหรับงานวิจัยนี้เท่านั้น สกีมาของแบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์สามารถเขียนเป็นรูปได้ดังรูปที่ 3.6

จากรูปที่ 3.6 สกีมามีรากคือ element “Class” โดยมีส่วนรายละเอียดสำคัญอยู่ใน element “Blockstatement” ซึ่งประกอบด้วย element “LocalVariableDeclaration” ใช้สำหรับประกาศตัวแปรก่อนเริ่มต้นการทำงาน และ element กลุ่ม Statement ซึ่งประกอบไปด้วย

- Statement_if ใช้สำหรับการกำหนดเงื่อนไข

- Statement_while ใช้สำหรับการวนซ้ำ โดยการตรวจสอบเงื่อนไขจะทำก่อนการทำงานแต่ละรอบ
- Statement_dowhile ใช้สำหรับการวนซ้ำ โดยการตรวจสอบเงื่อนไขจะทำหลังการทำงานแต่ละรอบ
- Statement_for ใช้สำหรับการวนซ้ำแบบรู้จำนวนรอบ
- Expression ใช้สำหรับการทำงานต่าง ๆ โดยการทำงานแต่ละแบบจะต้องกำหนดค่า attribute “type” ที่ต่างกัน ดังนี้
  - parameter-to-method สำหรับรับค่าเข้าเมท็อด
  - return-from-method สำหรับคืนค่าออกจากเมท็อด
  - method-invocation สำหรับเรียกการทำงานของเมท็อดอื่น
  - webservice-invocation สำหรับเรียกการทำงานของเว็บเซอร์วิซ
  - value-assignment สำหรับกำหนดค่าตัวแปร
  - exit สำหรับจบการทำงานทันที



รูปที่ 3.6 สกีมาของแบบจำลองเมตาสำหรับแบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

3.2 กฎการแปลง

การแปลงระหว่างแบบจำลองแต่ละระดับตามโครงสร้างที่ได้นำเสนอในรูปที่ 3.2 นั้นจะอาศัยกฎการแปลง (Transformation Rule) โดยในงานวิจัยนี้นำเสนอกฎการแปลงดังนี้

3.2.1 กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม

กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็ม ไปยังแบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม มีดังนี้

1) Stereotype ของระดับพีไอเอ็ม 1 คลาส แปลงไปยัง Stereotype ของระดับพีเอสเอ็ม ได้ 1 คลาส ตามความสัมพันธ์ในตารางที่ 3.2 โดยที่ Stereotype “RetrieveContext” และ “Loop” มีรายละเอียดเพิ่มเติมดังนี้
   - RetrieveContext จะถูกแปลงไปยัง InvokeAPI ที่ใช้เรียกเอพีไอในระบบ หรือ InvokeService ที่ใช้เรียกเซอร์วิซแบบ SOAP ขึ้นอยู่กับว่าในแพลตฟอร์มนั้น ๆ มีเอพีไอที่ถูกระบุไว้ในแท็ก detail ใน Stereotype ระดับพีไอเอ็มหรือไม่
     - ถ้ามีจะให้แปลงไปยัง InvokeAPI
     - ถ้าไม่มีจะให้แปลงไปยัง InvokeService
   - Loop จะถูกแปลงไปยัง While หรือ RepeatUntil ขึ้นอยู่กับการระบุในแท็ก detail ตามไวยากรณ์ที่ระบุในตารางที่ 3.1
2) การแปลงส่วนของแท็กของ Stereotype ระดับพีเอสเอ็ม ให้อ้างอิงจากไวยากรณ์ของข้อความในแท็ก detail ของ Stereotype ระดับพีไอเอ็มที่ระบุในตารางที่ 3.1 โดยการตัดคำด้วยช่องว่าง แล้วนำส่วนของข้อความที่ได้ไปใส่ให้แก่แท็กต่าง ๆ ของ Stereotype ระดับพีเอสเอ็มโดยมีรายละเอียดดังตารางที่ 3.3

ในงานวิจัยนี้กฎการแปลงในระดับนี้สำหรับแต่ละแพลตฟอร์มมีรายละเอียดการทำงานเหมือนกัน เนื่องจากใช้แปลงจากแบบจำลองระดับพีไอเอ็มไปยังแบบจำลองระดับพีเอสเอ็มที่มีแบบจำลองเมตาร่วมกัน โดยกฎการแปลงมีส่วนที่แตกต่างกันหลัก ๆ คือ ส่วนที่ใช้แปลง RetrieveContext ในระดับพีไอเอ็ม ไปเป็น InvokeAPI หรือ InvokeService ในระดับพีเอสเอ็ม เนื่องจากมีการระบุถึงที่อยู่ของเอพีไอที่ขึ้นกับแพลตฟอร์มนั้น ๆ หรือระบุตำแหน่งของข้อมูลบริบทภายนอกแล้วแต่กรณี

ตารางที่ 3.2 ความสัมพันธ์ระหว่าง Stereotype ระดับพีไอเอ็ม และระดับพีเอสเอ็ม

PIM Stereotype	*PSM Stereotype*
InitializeVariable	InitializeVariable
Assign	Assign
Receive	Receive
Reply	Reply
Exit	Exit
RetrieveContext	InvokeAPI InvokeService
Sequences	Sequences
Condition	If
Loop	While RepeatUntil

ตารางที่ 3.3 ความสัมพันธ์ระหว่างส่วนของข้อความของแท็ก detail ของ Stereotype ระดับพีไอเอ็ม และแท็กของ Stereotype ระดับพีเอสเอ็ม

PIM Stereotype	*Part of "detail" tag*	*PSM Stereotype*	*PSM Stereotype tag*
InitializeVariable	$VariableType	InitializeVariable	type
	$VariableName		name
	$VariableValue		value
	$VariableNamespace		namespace
Assign	$VariableName	Assign	copyTo
	$VariableValue		copyFrom
Receive	$VariableName	Receive	variableName
Reply	$VariableName	Reply	variableName
Exit	-	Exit	-
RetrieveContext	$ContextSource	InvokeAPI InvokeService	operation
	$VariableInputName		inputVariable
	$VariableOutputName		outputVariable
Sequences	$ActivityName	Sequences	activityName
Condition	$ConditionDetail	If	condition
Loop	$ConditionDetail	While RepeatUntil	condition

3.2.2 กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง บริบทประกอบเชิงกระบวนการระดับโค้ด

กฎการแปลงในระดับนี้เป็นกฎการแปลงที่มีการทำงานเฉพาะเจาะจงสำหรับแต่ละแพลตฟอร์มเนื่องจากใช้แปลงแบบจำลองระดับพีเอสเอ็มไปยังแบบจำลองระดับโค้ดที่มีแบบจำลองเมตาต่างกัน สามารถแบ่งได้เป็น *กฎการแปลงสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์* และ*กฎการแปลงสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์*

3.2.2.1 กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง แบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์

กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง แบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ มีดังนี้

1) Stereotype ของระดับพีเอสเอ็ม 1 คลาสแปลงไปยัง element ของสกีมาระดับโค้ด 1 element โดยมีความสัมพันธ์ตามที่แสดงในตารางที่ 3.4
2) แท็กของ Stereotype ของระดับพีเอสเอ็ม แปลงไปยัง attribute ของสกีมาระดับโค้ด โดยมีความสัมพันธ์ตามที่แสดงในตารางที่ 3.5

ตารางที่ 3.4 ความสัมพันธ์ระหว่าง Stereotype ระดับพีเอสเอ็ม และ element ระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์

PSM Stereotype	*Code Element (Windows Mobile)*
InitializeVariable	Variable
Assign	Assign
Receive	Receive
Reply	Reply
Exit	Exit
InvokeAPI	InvokeAPI
InvokeService	InvokeService
Sequences	Sequences
If	If
While	While
RepeatUntil	RepeatUntil

ตารางที่ 3.5 ความสัมพันธ์ระหว่างแท็กของ Stereotype ระดับพีเอสเอ็ม และ attribute ของ Stereotype ระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์

PSM Stereotype	*tag*	*Code Element (Windows Mobile)*	*attribute*
InitializeVariable	type	Variable	messageType
	name		name
	value		value
	namespace		namespace
Assign	copyTo	Assign	to
	copyFrom		from
Receive	variableName	Receive	variable
Reply	variableName	Reply	variable
Exit	-	Exit	-
InvokeAPI	operation	InvokeAPI	operation
	inputVariable		inputVariable
	outputVariable		outputVariable
	package		package
InvokeService	operation	InvokeService	operation
	inputVariable		inputVariable
	outputVariable		outputVariable
	URL		url
Sequences	activityName	Sequences	activity
If	condition	If	condition
While	condition	While	condition
RepeatUntil	condition	RepeatUntil	condition

3.2.2.2 กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง แบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง แบบจำลองระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ มีดังนี้

1) Stereotype ของระดับพีเอสเอ็ม 1 คลาสแปลงไปยัง element ของสกีมาระดับโค้ด 1 element โดยมีความสัมพันธ์ตามที่แสดงในตารางที่ 3.6
2) แท็กของ Stereotype ของระดับพีเอสเอ็ม แปลงไปยัง attribute ของสกีมาระดับโค้ด โดยมีความสัมพันธ์ตามที่แสดงในตารางที่ 3.7 โดยที่ Stereotype “Assign”, “Receive”, “Reply”, “Exit”, “InvokeAPI”, “InvokeService” จะแปลงไปยัง element “expression” เหมือนกันแต่แตกต่างกันที่รายละเอียดของ attribute “type” ซึ่งใช้ระบุชนิดของการทำงาน และรายละเอียดของ attribute “detail” ซึ่งใช้ระบุรายละเอียดการทำงาน โดยมีรายละเอียดดังแสดงในตารางที่ 3.7 เช่นเดียวกัน

ตารางที่ 3.6 ความสัมพันธ์ระหว่าง Stereotype ระดับพีเอสเอ็ม และ element ระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

PSM Stereotype	*Code Element (Android)*
InitializeVariable	Variable
Assign	Expression
Receive	Expression
Reply	Expression
Exit	Expression
InvokeAPI	Expression
InvokeService	Expression
Sequences	Blockstatement
If	Statement_if
While	Statement_while
RepeatUntil	Statement_dowhile

ผลลัพธ์ที่ได้ในการแปลงในระดับโค้ดอยู่ในรูปของแบบจำลองของโค้ดที่มีหน้าตาและรายละเอียดใกล้เคียงกับโค้ดของแต่ละแพลตฟอร์มมาก การนำไปใช้เพียงแค่นำแบบจำลองนี้ไปจัดรูปแบบใหม่แปลงเป็นโค้ด ที่สามารถใช้งานได้กับแพลตฟอร์มนั้น ๆ โดยกระบวนการนี้อาจทำโดยสร้างเครื่องมืออัตโนมัติหรือทำด้วยมือก็ได้

ตารางที่ 3.7 ความสัมพันธ์ระหว่างแท็กของ Stereotype ระดับพีเอสเอ็ม และ attribute ของ Stereotype ระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

PSM Stereotype	*tag*	*Code Element (Android)*	*attribute*
InitializeVariable	type	Variable	type
	name		name
	value		value
	namespace		namespace
Assign	copyTo copyFrom	Expression	type = “value-assignment"
			detail = “*copyTo* = *copyFrom*”
Receive	variableName	Expression	type = “parameter-to-method"
			detail = “parameter *variableName*”
Reply	variableName	Expression	type = “return-from-method"
			detail = “return *variableName*”
Exit	-	Expression	type = “exit"
			detail = “exit”
InvokeAPI	operation inputVariable outputVariable package	Expression	type = “method-invocation"
			detail = “*outputVariable* = *package*. *operation*(*inputVariable*, …)”
InvokeService	operation inputVariable outputVariable URL	Expression	type = “webservice-invocation"
			detail = “*outputVariable* = call(*URL*, *operation*, { *inputVariable*, …})”
Sequences	activityName	Blockstatement	-
If	condition	Statement_if	condition
While	condition	Statement_while	condition
RepeatUntil	condition	Statement_dowhile	condition

ในบทนี้ได้แสดงการออกแบบส่วนประกอบสำคัญ 2 ส่วนซึ่งใช้ในการพัฒนาบริบทประกอบเชิงกระบวนการโดยอิงสถาปัตยกรรมอิงแบบจำลอง ได้แก่ (1) แบบจำลองเมตา ที่ใช้เป็นข้อกำหนดในการเขียนแบบจำลอง และ (2) กฎการแปลง ที่ใช้ในการแปลงระหว่างแบบจำลองแต่ละระดับ โดยแสดงตัวอย่างบนโทรศัพท์เคลื่อนที่ 2 แพลตฟอร์ม ได้แก่ โทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ และโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ การออกแบบแบบจำลองระดับพีไอเอ็ม และพีเอสเอ็มนำเสนอในรูปแบบของยูเอ็มแอลโปรไฟล์ ในขณะที่แบบจำลองระดับโค้ดนำเสนอในรูปแบบของสกีมาของแต่ละภาษา และได้กำหนดไวยากรณ์สำหรับเขียนคำอธิบายในระดับพีไอเอ็มเพื่อให้สามารถแปลงไปยังแบบจำลองระดับอื่น ๆ ได้โดยง่าย ส่วนกฎการแปลงถูกออกแบบไว้ในรูปแบบของข้อความ โดยระบุขั้นตอนการแปลงระหว่างแบบจำลองระดับพีไอเอ็ม ไปยังแบบจำลองระดับพีเอสเอ็ม และไปยังแบบจำลองระดับโค้ดตามลำดับ รวมถึงความสัมพันธ์ระหว่างแบบจำลองเมตาแต่ละระดับ ซึ่งทั้งแบบจำลองเมตา และกฎการแปลง จะถูกนำไปใช้ในการพัฒนาเอพีไอสำหรับบริบทประกอบเชิงกระบวนการโดยอิงสถาปัตยกรรมอิงแบบจำลองต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การทดสอบ

การทดสอบทำโดยทดสอบการใช้งานแบบจำลองเมตา และกฎการแปลงที่ได้ออกแบบโดยอิงสถาปัตยกรรมอิงแบบจำลองดังแสดงไว้ในบทที่ 3 มาทำการสร้างเอพีไอต้นแบบเพื่อให้ค่าบริบทประกอบเชิงกระบวนการสำหรับโทรศัพท์เคลื่อนที่ 2 ประเภท ได้แก่ โทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ และโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ โดยประเมินผลว่าเอพีไอที่สร้างได้สามารถทำงานได้ถูกต้องตามความต้องการเชิงหน้าที่ (Functional Requirement) ที่ได้ออกแบบไว้หรือไม่ กล่าวคือ เอพีไอต้นแบบต้องสามารถให้ค่าบริบทประกอบเชิงกระบวนการได้เมื่อถูกเรียกใช้งาน

ในการทดสอบนี้ได้สร้างเอพีไอต้นแบบเพื่อให้ค่าบริบทประกอบเชิงกระบวนการที่ชื่อว่า “Recommended Nearby Cinema” ซึ่งเป็นบริบทสถานที่ที่ให้ค่าโรงภาพยนตร์ที่น่าสนใจซึ่งมีภาพยนตร์ประเภทที่ผู้ใช้ชื่นชอบและอยู่บริเวณใกล้เคียงผู้ใช้ โดยบริบทนี้เป็นบริบทประกอบซึ่งประกอบด้วยบริบทย่อยได้แก่ พิกัดแผนที่ของตำแหน่งผู้ใช้ พิกัดแผนที่ของตำแหน่งโรงภาพยนตร์ต่าง ๆ รายชื่อภาพยนตร์ที่โรงภาพยนตร์นั้น ๆ ฉายอยู่ วันเวลา และความพึงใจของผู้ใช้ ซึ่งในที่นี้กำหนดให้ความพึงใจของผู้ใช้ประกอบไปด้วยประเภทของภาพยนตร์ที่ชื่นชอบและระยะทางที่สามารถยอมรับได้ รายละเอียดกระบวนการทำงานของบริบทประกอบนี้อยู่ในหัวข้อที่ 4.1

บริบทย่อยที่นำมาใช้ในการทดสอบในงานวิจัยนี้สามารถจัดกลุ่มตามการได้มา [1] ดังนี้

1) บริบทที่สามารถวัดค่าได้โดยตรง ได้แก่ พิกัดแผนที่ของตำแหน่งผู้ใช้
2) บริบทที่ได้รับมา ได้แก่ พิกัดแผนที่ของตำแหน่งโรงภาพยนตร์ต่าง ๆ วันเวลา และรายชื่อของภาพยนตร์ที่โรงภาพยนตร์นั้น ๆ ฉายอยู่ ซึ่งได้มาจากเว็บเซอร์วิซที่ผู้วิจัยพัฒนาขึ้น
3) บริบทที่ได้จากการระบุค่าโดยตรงโดยผู้ใช้งาน ได้แก่ ความพึงใจของผู้ใช้ โดยในส่วนนี้ผู้วิจัยได้พัฒนาไลบรารีเพื่อใช้ในการสร้าง จัดเก็บ ปรับปรุง และเรียกค่าบริบทดังกล่าวสำหรับแต่ละแพลตฟอร์มไว้แล้ว

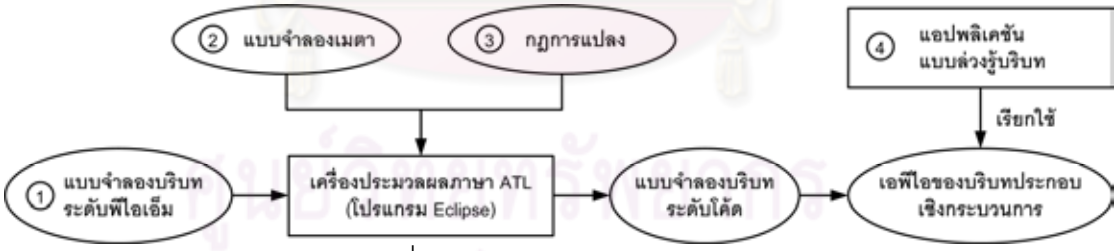
นอกจากนั้นบริบทย่อยที่นำมาใช้ในการทดสอบนี้ยังสามารถจัดกลุ่มตามแหล่งกำเนิดได้ดังนี้

1) บริบทที่เป็นค่าพื้นฐานทางกายภาพ (Native Physical Context) คือ บริบทที่เป็นค่าที่วัดได้จากฮาร์ดแวร์ของเครื่อง ในกรณีนี้ได้แก่ พิกัดแผนที่ของตำแหน่งผู้ใช้

2) บริบทที่เป็นค่านามธรรม (Abstract Context) คือ บริบทที่เป็นค่ากำหนดเพิ่มเติมจากค่าพื้นฐานที่สามารถวัดได้จากฮาร์ดแวร์ โดยอยู่ในรูปแบบของซอฟต์แวร์ที่ทำหน้าที่ให้ค่าบริบทในลักษณะเดียวกับฮาร์ดแวร์ ในกรณีนี้ได้แก่ พิกัดแผนที่ของตำแหน่งโรงภาพยนตร์ต่าง ๆ รายชื่อภาพยนตร์ที่โรงภาพยนตร์นั้น ๆ ฉายอยู่ วันเวลา และ ความพึงใจของผู้ใช้ ในงานวิจัยนี้ผู้วิจัยได้จัดทำไลบรารีและเว็บเซอร์วิซ ซึ่งประกอบด้วย เอพีไอเพื่อใช้เรียกค่าบริบทที่เป็นค่านามธรรมสำหรับแต่ละแพลตฟอร์มไว้ด้วย

บริบทต้นแบบที่สร้างขึ้นอยู่ในรูปแบบของเอพีไอสำหรับให้นักพัฒนาแอปพลิเคชันเรียกใช้งานเพื่อใช้สร้างแอปพลิเคชันแบบล่วงรู้บริบทต่อไป โดยในงานวิจัยนี้ได้สร้างแอปพลิเคชันต้นแบบที่มีการเรียกค่าจากบริบทประกอบ Recommended Nearby Cinema ที่ได้ไปใช้งานด้วย ภาพรวมการทดสอบสามารถสรุปได้ดังรูปที่ 4.1 โดยมีขั้นตอนที่ต้องพัฒนาดังนี้

1) พัฒนาแบบจำลองบริบทต้นแบบระดับพีไอเอ็มของบริบท “Recommended Nearby Cinema” โดยใช้ แบบจำลองเมตาสำหรับแบบจำลองระดับพีไอเอ็ม ที่ได้ออกแบบไว้ในหัวข้อที่ 3.1.1 เป็นแบบจำลองเมตาในการพัฒนาแบบจำลอง
2) พัฒนาแบบจำลองเมตาตามที่ได้ออกแบบไว้ในหัวข้อที่ 3.1 ทั้งระดับพีไอเอ็ม พีเอสเอ็ม และโค้ด รวมทั้งหมด 4 แบบจำลองเมตา
3) พัฒนากฎการแปลงตามที่ได้ออกแบบไว้ในหัวข้อที่ 3.2 รวมทั้งหมด 4 กฎ
4) พัฒนาแอปพลิเคชันต้นแบบเพื่อทดลองเรียกใช้งานเอพีไอ

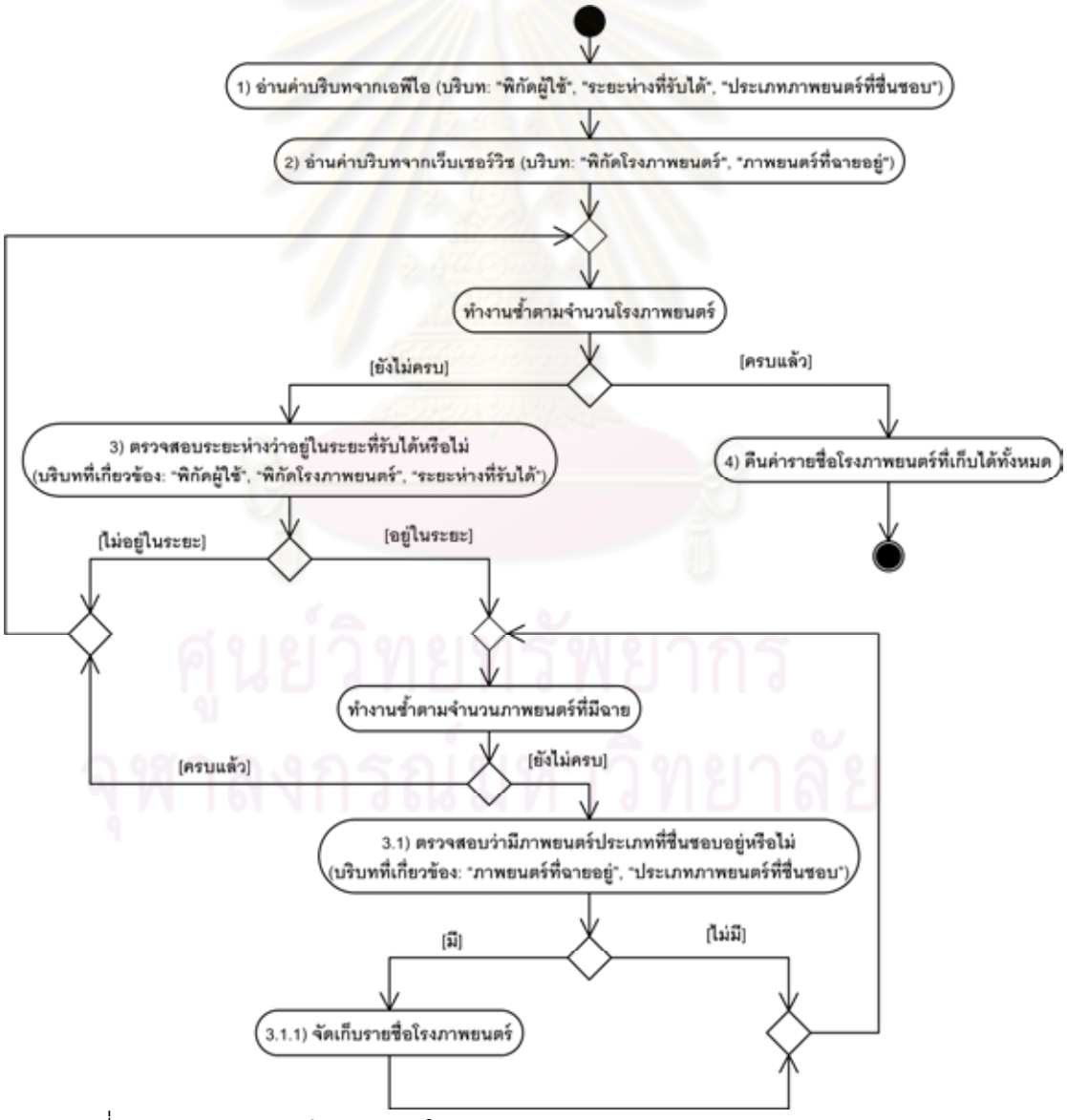


รูปที่ 4.1 ภาพรวมของการทดสอบ

4.1 การพัฒนาแบบจำลองบริบทต้นแบบระดับพีไอเอ็ม

การพัฒนาแบบจำลองบริบทต้นแบบระดับพีไอเอ็ม (PIM-Level Context Model Development) เป็นการสร้างแบบจำลองของบริบทประกอบที่ชื่อว่า Recommended Nearby Cinema ซึ่งเป็นบริบทสถานที่ที่ให้ค่าโรงภาพยนตร์ที่น่าสนใจ มีภาพยนตร์ประเภทที่ผู้ใช้ชื่นชอบและอยู่บริเวณใกล้เคียงผู้ใช้ ในงานวิจัยนี้บริบทที่เป็นส่วนประกอบของบริบทนี้ สามารถจัดกลุ่มตามความสัมพันธ์กับระบบซึ่งในที่นี้หมายถึงโทรศัพท์เคลื่อนที่ ได้ดังนี้

1) ค่าบริบทที่ได้จากภายในระบบ โดยในงานวิจัยนี้หมายถึงค่าบริบทที่ได้จากเอพีไอภายในอุปกรณ์นั้นๆ ได้แก่
   - พิกัดแผนที่ของตำแหน่งผู้ใช้
   - ประเภทภาพยนตร์ที่ผู้ใช้ชื่นชอบ
   - ระยะห่างระหว่างตำแหน่งผู้ใช้กับตำแหน่งของโรงภาพยนตร์ที่ผู้ใช้รับได้
2) ค่าบริบทที่ได้จากภายนอกระบบ โดยในงานวิจัยนี้หมายถึงค่าบริบทที่ได้จากเว็บเซอร์วิซ ได้แก่
   - ข้อมูลเกี่ยวกับโรงภาพยนตร์ต่าง ๆ ซึ่งประกอบด้วยพิกัดแผนที่ของตำแหน่งโรงภาพยนตร์ และรายชื่อภาพยนตร์ที่โรงภาพยนตร์นั้น ๆ ฉายอยู่ในช่วงเวลานั้น



รูปที่ 4.2 กระบวนการทำงานภายในบริบทประกอบ Recommended Nearby Cinema

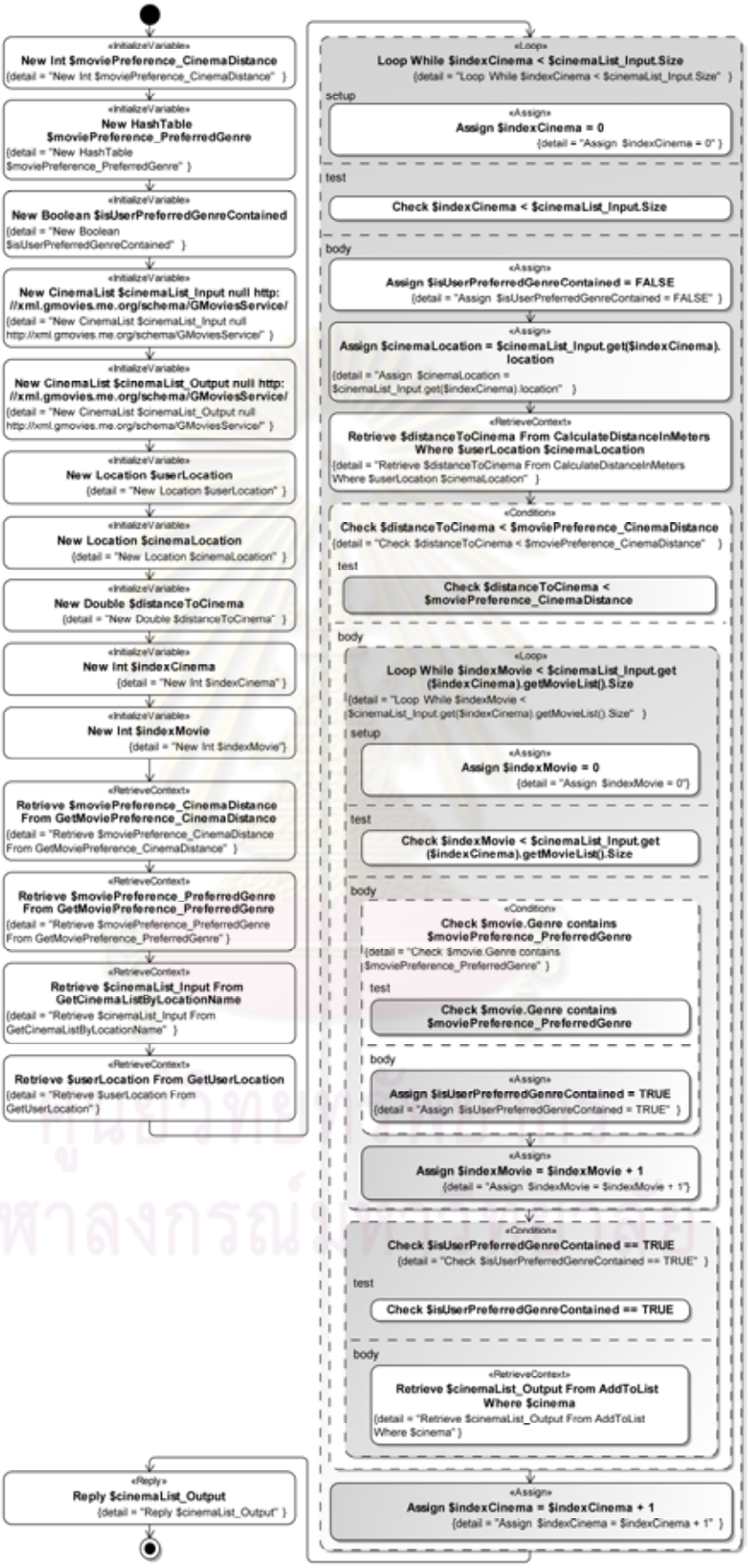
กระบวนการทำงานภายในบริบทประกอบ Recommended Nearby Cinema สามารถแสดงเป็นแผนภาพกิจกรรมของยูเอ็มแอลได้ดังรูปที่ 4.2 โดยสรุปได้ดังนี้

1) อ่านค่าบริบทจากเอพีไอ ได้แก่ “พิกัดแผนที่ของตำแหน่งผู้ใช้” “ประเภทภาพยนตร์ที่ผู้ใช้ชื่นชอบ” “ระยะห่างระหว่างตำแหน่งผู้ใช้กับตำแหน่งของโรงภาพยนตร์ที่ผู้ใช้รับได้”
2) อ่านค่าบริบทจากเว็บเซอร์วิซ ได้แก่ “ข้อมูลเกี่ยวกับโรงภาพยนตร์ต่าง ๆ ซึ่งประกอบด้วยพิกัดแผนที่ของตำแหน่งโรงภาพยนตร์ และภาพยนตร์ที่กำลังฉายอยู่ในช่วงเวลานั้น”
3) ตรวจสอบว่าระยะห่างระหว่างตำแหน่งผู้ใช้กับตำแหน่งโรงภาพยนตร์หนึ่งๆ มีระยะห่างไม่เกินระยะห่างที่ผู้ใช้รับได้หรือไม่ โดยทำการวนซ้ำโดยมีจำนวนรอบการวนซ้ำเท่ากับจำนวนโรงภาพยนตร์ที่ได้รับมาจากเว็บเซอร์วิซ
    3.1) ถ้าระยะห่างไม่เกินระยะห่างที่ผู้ใช้รับได้ ให้ตรวจสอบว่าโรงภาพยนตร์นั้น ๆ มีภาพยนตร์ประเภทที่ผู้ใช้ชื่นชอบฉายอยู่หรือไม่
        3.1.1) ถ้ามีภาพยนตร์ประเภทที่ผู้ใช้ชื่นชอบฉายอยู่ ให้จัดเก็บรายชื่อโรงภาพยนตร์นั้นไว้
4) คืนค่ารายชื่อโรงภาพยนตร์ทั้งหมดที่ตรงตามเงื่อนไข

แผนภาพกิจกรรมของยูเอ็มแอลที่แสดงในรูปที่ 4.2 นั้น ใช้อธิบายกระบวนการทำงานภายในบริบทต้นแบบเท่านั้น การพัฒนาแบบจำลองบริบทต้นแบบระดับพีไอเอ็มจำเป็นต้องเขียนแผนภาพกิจกรรมโดยระบุ Stereotype ตามยูเอ็มแอลโปรไฟล์ที่ได้ออกแบบไว้ในหัวข้อที่ 3.1.1 รวมทั้งเขียนรายละเอียดในแท็ก detail ตามไวยากรณ์ที่ได้ออกแบบไว้ในตารางที่ 3.1 เครื่องมือที่ใช้พัฒนาแบบจำลองระดับพีไอเอ็มในงานวิจัยนี้ ได้แก่ โปรแกรม MagicDraw UML เวอร์ชัน 16.6 [26] โดยมีขั้นตอนการพัฒนาดังนี้

1) พัฒนาแบบจำลองเมตาระดับพีไอเอ็มภายในโปรแกรมดังกล่าว ตามที่ได้ออกแบบไว้ในหัวข้อที่ 3.1.1
2) พัฒนาแบบจำลองของบริบทต้นแบบ โดยมีกระบวนการทำงานภายในบริบทตามที่ได้กล่าวไว้ข้างต้น

ผลลัพธ์ของการพัฒนาบริบทจากโปรแกรมดังกล่าวสามารถแสดงเป็นแผนภาพกิจกรรมได้ดังรูปที่ 4.3 จากนั้นทำการบันทึกเป็นไฟล์เอ็กซ์เอ็มแอลออกมาในรูปแบบ EMF UML2 v2.x XMI ซึ่งแสดงข้อมูลแทนแผนภาพกิจกรรม สำหรับนำไปใช้ในการพัฒนาแอปพลิเคชันแบบล่วงรู้บริบทด้วยกระบวนการแบบเอ็มดีเอในหัวข้อที่ 4.4 ต่อไป



รูปที่ 4.3 แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็มของบริบทต้นแบบ

4.2 การพัฒนาแบบจำลองเมตา

การพัฒนาแบบจำลองเมตา (Metamodel Development) ในงานวิจัยนี้ได้พัฒนาแบบจำลองเมตาในรูปแบบของไฟล์ ecore โดยได้ออกแบบให้แต่ละไฟล์เก็บข้อมูลแบบจำลองเมตา 1 แบบ สำหรับแบบจำลองเมตาในระดับพีไอเอ็มและแบบจำลองเมตาในระดับพีเอสเอ็มนั้นในไฟล์ ecore 1 ไฟล์จะประกอบด้วย

- ข้อมูลที่เกี่ยวกับยูเอ็มแอล โดยในที่นี้เรียกใช้เฉพาะคลาสที่เกี่ยวกับแผนภาพกิจกรรมของยูเอ็มแอล
- ข้อมูลที่เกี่ยวกับยูเอ็มแอลโปรไฟล์ ซึ่งใช้เก็บข้อมูล Stereotype และ Tag ที่ได้ออกแบบไว้
- ข้อมูลส่วนพื้นฐานอื่น ๆ ที่จำเป็น

สำหรับแบบจำลองเมตาในระดับโค้ดนั้น ไฟล์ ecore จะมีโครงสร้างคล้ายไฟล์ ecore ในระดับพีไอเอ็มและพีเอสเอ็ม แต่เปลี่ยนส่วนที่เป็นข้อมูลที่เกี่ยวกับยูเอ็มแอลโปรไฟล์ เป็นส่วนที่เกี่ยวกับสกีมาแทน รายละเอียดการสร้างไฟล์ ecore เฉพาะส่วนที่เกี่ยวข้องกับแบบจำลองเมตาที่ได้ออกแบบไว้มีดังนี้

1) สร้าง EPackage สำหรับเก็บข้อมูลแบบจำลองเมตาที่ได้ออกแบบไว้
2) สร้าง EClass สำหรับแต่ละ Stereotype (ของยูเอ็มแอลโปรไฟล์) หรือ Element (ของสกีมา)
3) สร้าง EAttribute สำหรับแต่ละ Tag (ของยูเอ็มแอลโปรไฟล์) หรือ Attribute (ของสกีมา)

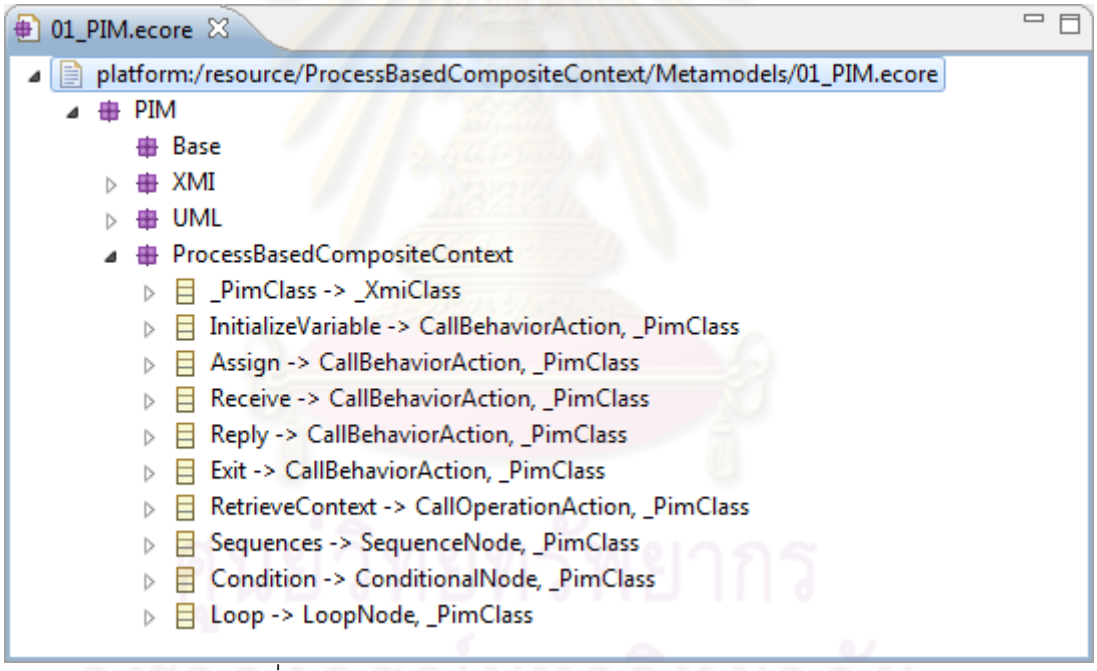
ในรูปที่ 4.4 ถึงรูปที่ 4.7 เป็นตัวอย่างไฟล์ ecore สำหรับแบบจำลองเมตาทั้ง 4 แบบที่สร้างขึ้นในงานวิจัยนี้ โดยแสดงให้เห็นเฉพาะรายละเอียดภายใน EPackage ในส่วนที่ออกแบบเพิ่มตามหัวข้อที่ 3.1

เครื่องมือหลักที่ใช้ในการพัฒนาแบบจำลองเมตาคือ Eclipse เวอร์ชัน 3.6.x (Helios) รุ่นแพ็คเกจ Eclipse Modeling Tools (Includes Incubating Components) [17] ซึ่งได้รวมชุดคอมโพเนนท์เสริมเกี่ยวกับการพัฒนาแบบจำลองจากโครงการ Eclipse Modeling Project [18] เข้าไว้เรียบร้อยแล้ว

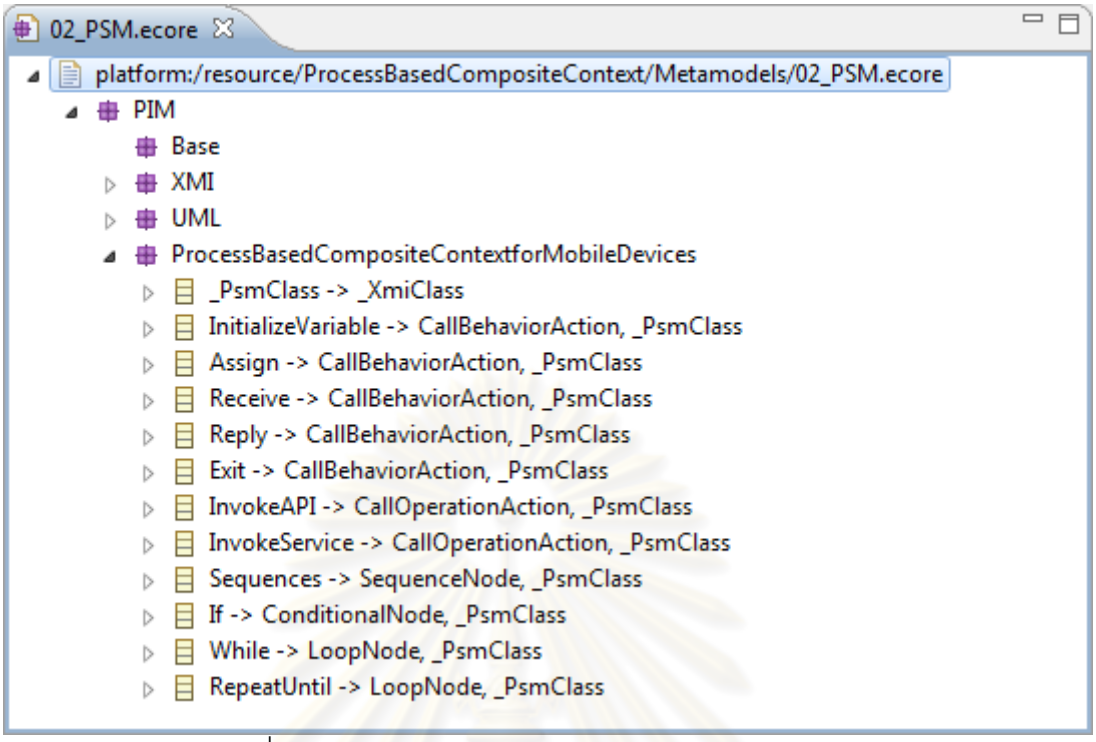
ผลลัพธ์ของการพัฒนาแบบจำลองเมตาในงานวิจัยนี้ได้ไฟล์ ecore ทั้งหมด 4 ไฟล์ ได้แก่

1) ไฟล์ ecore ของแบบจำลองเมตาระดับพีไอเอ็ม (รูปที่ 4.4)
2) ไฟล์ ecore ของแบบจำลองเมตาระดับพีเอสเอ็ม (รูปที่ 4.5)
3) ไฟล์ ecore ของแบบจำลองเมตาระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ (รูปที่ 4.6)
4) ไฟล์ ecore ของแบบจำลองเมตาระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ (รูปที่ 4.7)

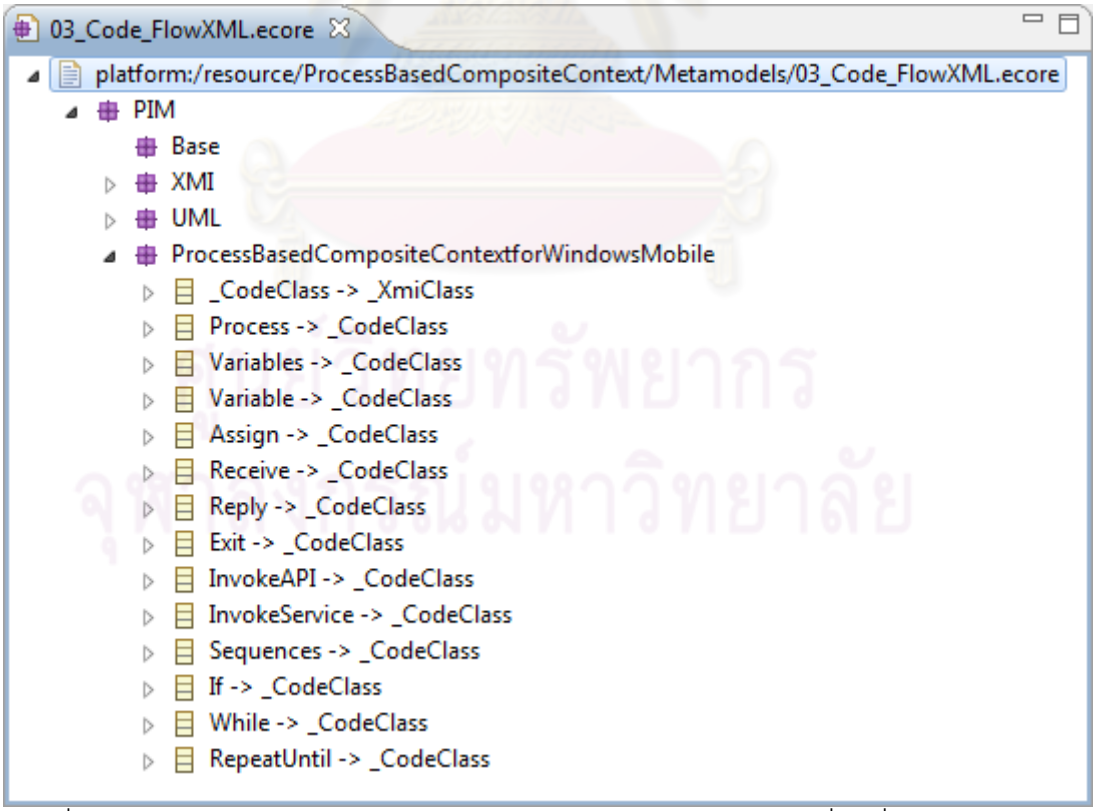
ไฟล์แบบจำลองเมตาข้างต้นจะถูกนำไปใช้ในการพัฒนาแอปพลิเคชันแบบล่วงรู้บริบทด้วยกระบวนการแบบเอ็มดีเอในหัวข้อที่ 4.4 ต่อไป



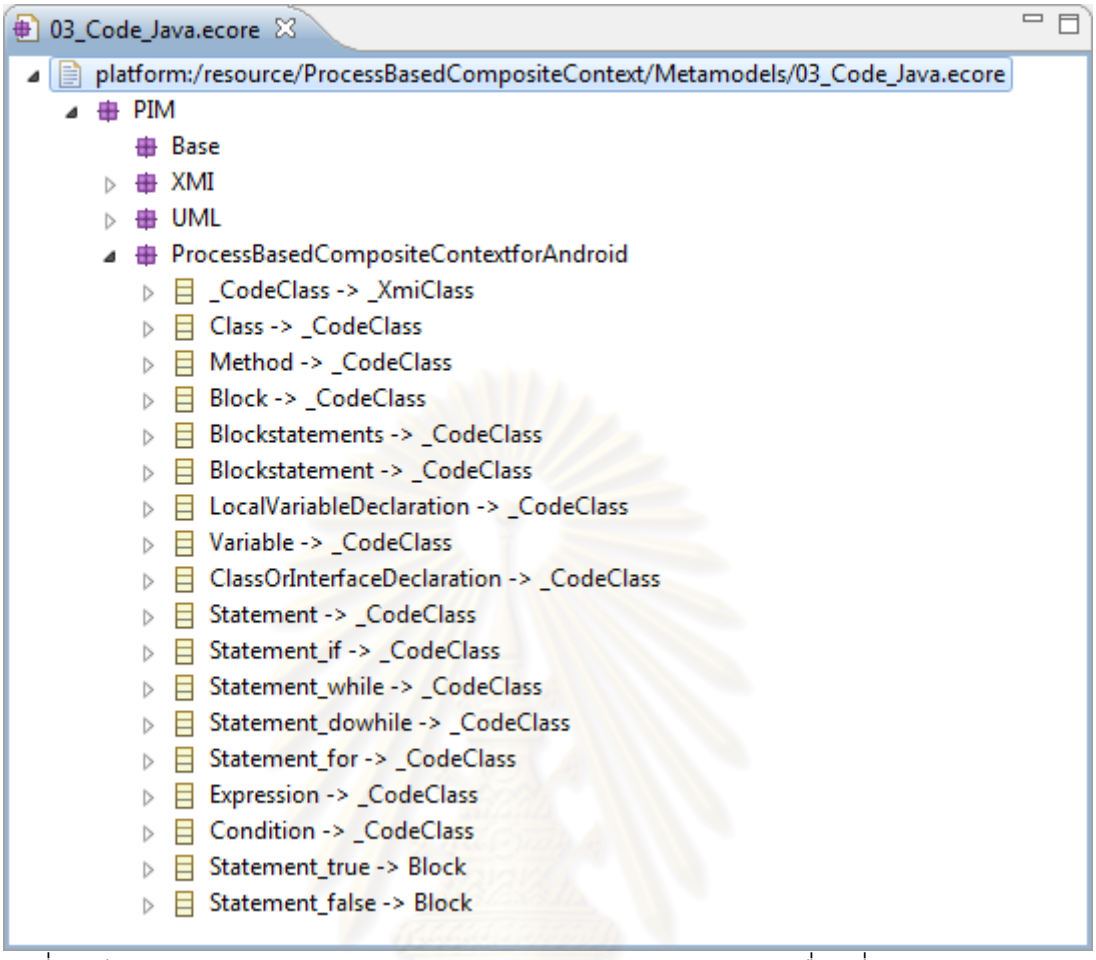
รูปที่ 4.4 ไฟล์ ecore ของแบบจำลองเมตาระดับพีไอเอ็ม



รูปที่ 4.5 ไฟล์ ecore ของแบบจำลองเมตาระดับพีเอสเอ็ม



รูปที่ 4.6 ไฟล์ ecore ของแบบจำลองเมตาระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์



รูปที่ 4.7 ไฟล์ ecore ของแบบจำลองเมตาระดับโค้ดสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

4.3 การพัฒนากฎการแปลง

การพัฒนากฎการแปลง (Transformation Rule Development) ในงานวิจัยนี้อยู่ในรูปแบบของภาษา ATL (ATLAS Transformation Language) [19] ซึ่งเป็นภาษาสำหรับการแปลงรูปแบบหนึ่ง โดยภาษานี้ได้ยึดถือเอา Query/View/Tranformation (QVT) ที่กำหนดโดย OMG เป็นหลัก

กฎการแปลงที่พัฒนาขึ้นนี้จัดเก็บในรูปแบบของไฟล์ ATL โดยกำหนดให้ไฟล์ ATL แต่ละไฟล์ แทนกฎการแปลงระหว่างแบบจำลองเมตาหนึ่งคู่ เช่น ไฟล์ ATL สำหรับ “กฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็มสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์” รายละเอียดการเขียนกฎอ้างอิงตามที่ได้ออกแบบไว้ในหัวข้อที่ 3.2

เครื่องมือหลักที่ใช้พัฒนากฎการแปลงเป็นชุดเดียวกับที่ใช้พัฒนาแบบจำลองเมตาซึ่งก็คือ Eclipse เวอร์ชัน 3.6.x (Helios) รุ่นแพ็คเกจ Eclipse Modeling Tools (Includes Incubating Components) [17] โดยลงส่วนเสริมที่เกี่ยวกับ ATL [19] เพิ่มเติม เพื่อใช้คุณสมบัติของการรองรับการเขียนและประมวลภาษา ATL

ผลลัพธ์ของการพัฒนากฎการแปลงในงานวิจัยนี้ได้ไฟล์ ATL ทั้งหมด 4 ไฟล์ ได้แก่

1) ไฟล์ ATL ของกฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม สำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์
2) ไฟล์ ATL ของกฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม สำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์
3) ไฟล์ ATL ของกฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับโค้ด สำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์
4) ไฟล์ ATL ของกฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับโค้ด สำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

รูปที่ 4.8 เป็นตัวอย่างส่วนหนึ่งของไฟล์ ATL ของกฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีไอเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม สำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ โดยในบรรทัดที่ 11 – 52 เป็นกฎที่ชื่อว่า tInvokeAPI ใช้สำหรับแปลงแอคทิวิตีระดับพีไอเอ็มที่มี Stereotype ระบุว่า RetrieveContext ไปเป็นแอคทิวิตีระดับพีเอสเอ็มที่มี Stereotype ระบุว่า InvokeAPI

หลักการทำงานของกฎ tInvokeAPI คือ ค้นหาแอคทิวิตีระดับพีไอเอ็มที่มี Stereotype ระบุว่า RetrieveContext ซึ่งมีส่วนของข้อความในแท็ก detail ระบุถึงเอพีไอที่ระบบแอนดรอยด์ในงานวิจัยนี้รู้จัก (บรรทัดที่ 12 - 19) แล้วทำการแปลงไปยังแอคทิวิตีระดับพีเอสเอ็มที่มี Stereotype ระบุว่า InvokeAPI (บรรทัดที่ 20 - 51) การตีความข้อความในแท็ก detail ของ Stereotype ในระดับพีไอเอ็ม ทำตามไวยากรณ์ที่ออกแบบไว้ในตารางที่ 3.1 โดยในส่วนของ operation และ package ตัวกฎจะมีข้อมูลอยู่ว่าควรระบุเป็นอย่างไร (บรรทัดที่ 35-50)

```
1  -- @path PIM=/ProcessBasedCompositeContext/Metamodels/01_PIM.ecore
2  -- @path PSM=/ProcessBasedCompositeContext/Metamodels/02_PSM.ecore
3
4  module PIM2PSM;
5  create OUT : PSM from IN : PIM;
…  …
11 rule tInvokeAPI {
12     from
13         input : PIM!RetrieveContext(
14             input.detail.split(' ').at(4).toString() =
                   'GetMoviePreference_CinemaDistance' or
15             input.detail.split(' ').at(4).toString() =
                   'GetMoviePreference_PreferredGenre' or
16             input.detail.split(' ').at(4).toString() = 'GetUserLocation' or
17             input.detail.split(' ').at(4).toString() = 'AddToList' or
18             input.detail.split(' ').at(4).toString() = 'CalculateDistanceInMeters'
19         )
20     to
21         output : PSM!InvokeAPI (
22             __xmiID__ <- input.__xmiID__, _id <- input.__xmiID__, _id <- input._id,
23             base_CallOperationAction <- input.base_CallOperationAction,
24             name <- input.name
25         )
26     do{
27         output.outputVariable <- input.detail.split(' ').at(2).trim();
28         output.operation <- input.detail.split(' ').at(4).trim();
29         output.package <- '';
30
31         if(input.detail.split(' ').size() > 5){
32             output.inputVariable <- input.detail.split(' Where ').at(2).trim();
33         }
34
35         if(output.operation = 'GetMoviePreference_CinemaDistance'){
36             output.operation <- 'MoviePreferenceContext.getCinemaDistance';
37             output.package <- 'org.me.android.context.preference';
38         } else if(output.operation = 'GetMoviePreference_PreferredGenre'){
39             output.operation <- 'MoviePreferenceContext.getPreferredGenre';
40             output.package <- 'org.me.android.context.preference';
41         } else if(output.operation = 'GetUserLocation'){
42             output.operation <- 'LocationContext.getCurrentLocation';
43             output.package <- 'org.me.android.context';
44         } else if(output.operation = 'AddToList'){
45             output.operation <- 'List.addToList';
46             output.package <- 'org.me.android.common';
47         } else if(output.operation = 'CalculateDistanceInMeters'){
48             output.operation <- 'LocationContext.calculateDistanceInMeters';
49             output.package <- 'org.me.android.context';
50         }
51     }
52 }
…  …
```

รูปที่ 4.8 ตัวอย่างส่วนหนึ่งของไฟล์ ATL ของกฎการแปลงสำหรับแปลงแบบจำลองระดับพีไอเอ็มไปยังแบบจำลองระดับพีเอสเอ็มสำหรับแอนดรอยด์

รูปที่ 4.9 เป็นตัวอย่างส่วนหนึ่งของไฟล์ ATL ของกฎการแปลงสำหรับแปลง แบบจำลองบริบทประกอบเชิงกระบวนการระดับพีเอสเอ็ม ไปยัง แบบจำลองบริบทประกอบเชิงกระบวนการระดับโค้ด สำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ โดยในบรรทัดที่ 11 – 31 เป็นกฎที่ชื่อ tExpression_InvokeAPI ใช้สำหรับแปลงแอคทิวิตีระดับพีเอสเอ็มที่มี Stereotype ระบุว่า InvokeAPI ไปเป็นสกีมาระดับโค้ดที่มี Stereotype ระบุว่า Expression

หลักการทำงานของกฎ tExpression_InvokeAPI คือ ค้นหาแอคทิวิตีระดับพีเอสเอ็มที่มี Stereotype ระบุว่า InvokeAPI ทุกแอคทิวิตี (บรรทัดที่ 12 - 13) แล้วทำการแปลงไปยังสกีมาระดับโค้ดที่มี Stereotype ระบุว่า Expression (บรรทัดที่ 14 - 30) โดยการระบุรายละเอียดในแต่ละ tag ของสกีมาระดับโค้ดทำตามตารางที่ 3.7

```
1  -- @path PSM=/ProcessBasedCompositeContext/Metamodels/02_PSM.ecore
2  -- @path Code=/ProcessBasedCompositeContext/Metamodels/03_Code_Java.ecore
3
4  module PSM2Code_FlowXML;
5  create OUT : Code from IN : PSM;
…  …
11 unique lazy rule tExpression_InvokeAPI {
12     from
13         input : PSM!InvokeAPI
14     to
15         output : Code!Expression(
16             __xmiID__ <- input.__xmiID__, _id <- input.__xmiID__, _id <- input._id,
17             base <- input.base_CallOperationAction,
18             type <- 'method-invocation',
19             detail <- (input.outputVariable)
20                         .concat(' = ')
21                         .concat(input.package)
22                         .concat('.')
23                         .concat(input.operation)
24                         .concat('(')
25                         .concat((if(input.inputVariable.oclIsUndefined())
26                                         then ''
27                                         else input.inputVariable.replace(' ', ', ')
28                                         endif))
29                         .concat(')')
30         )
31 }
…  …
```

รูปที่ 4.9 ตัวอย่างส่วนหนึ่งของไฟล์ ATL ของกฎการแปลงสำหรับแปลงแบบจำลองระดับพีเอสเอ็มไปยังแบบจำลองระดับโค้ดสำหรับแอนดรอยด์

โค้ดด้านล่างนี้เป็นส่วนหนึ่งของแบบจำลองบริบทระดับพีไอเอ็มที่แสดงในรูปที่ 4.3 โดยเขียนในรูปแบบของไฟล์ XMI และนำมาเฉพาะตัวอย่างของแอคทิวิตีที่มี Stereotype ระบุว่า RetrieveContext เพื่อแสดงผลของการใช้งานกฎการแปลงในรูปที่ 4.8 และ รูปที่ 4.9

```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.1" xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore"
    xmlns:uml="http://www.eclipse.org/uml2/2.0.0/UML"
    xmlns:PIM="platform:/resource/ProcessBasedCompositeContext/Metamodels/01_PIM.ecore">
    …
    <PIM:RetrieveContext base_CallOperationAction="_16_6_1_2430132_1277959482285_387640_1270"
        detail="Retrieve $moviePreference_CinemaDistance From
        GetMoviePreference_CinemaDistance" xmi:id="_Y3h_oGKXEeCGgJeNNV-Q1g"/>
    …
</xmi:XMI>
```

เมื่อนำแบบจำลองระดับพีไอเอ็มข้างต้นไปแปลงด้วยกฎในรูปที่ 4.8 จะได้แบบจำลองระดับพีเอสเอ็ม ซึ่งมีแอคทิวิตีตัวอย่างเขียนในรูปแบบของไฟล์ XMI ได้ดังนี้

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:uml="http://www.eclipse.org/uml2/2.0.0/UML"
    xmlns:PSM="platform:/resource/ProcessBasedCompositeContext/Metamodels/02_PSM.ecore">
    …
    <PSM:InvokeAPI xmi:id="_Y3h_oGKXEeCGgJeNNV-Q1g" _id="_Y3h_oGKXEeCGgJeNNV-Q1g"
        base_CallOperationAction="_16_6_1_2430132_1277959482285_387640_1270"
        operation="MoviePreferenceContext.getCinemaDistance"
        outputVariable="$moviePreference_CinemaDistance"
        package="org.me.android.context.preference"/>
    …
</xmi:XMI>
```

และเมื่อนำแบบจำลองระดับพีเอสเอ็มที่แปลงได้มานี้ไปแปลงด้วยกฎในรูปที่ 4.9 จะได้แบบจำลองระดับโค้ด ซึ่งมีสกีมาตัวอย่างเขียนในรูปแบบของไฟล์ XMI ดังนี้

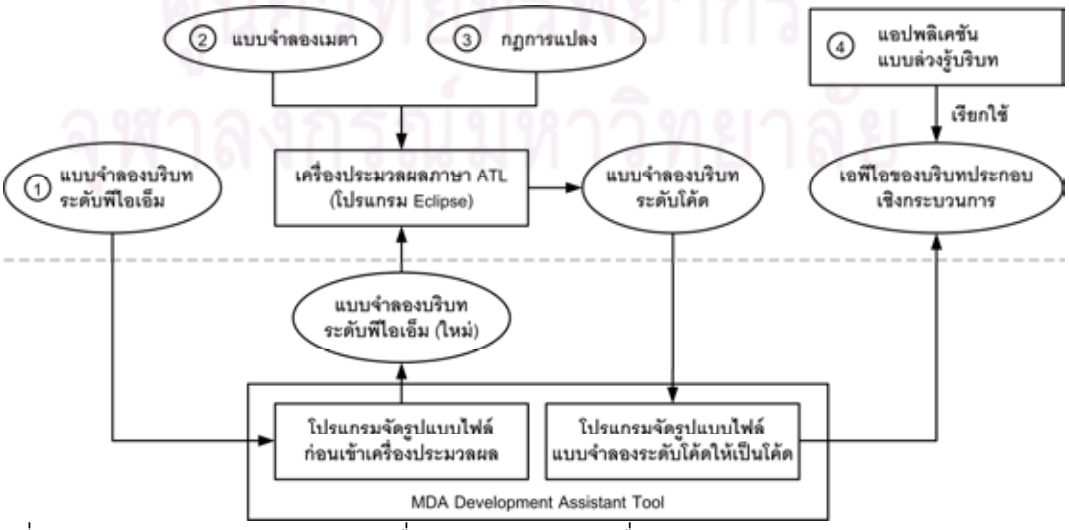
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:uml="http://www.eclipse.org/uml2/2.0.0/UML"
    xmlns:Code="platform:/resource/ProcessBasedCompositeContext/Metamodels/03_Code_Java.ecore">
    …
    <Code:Class …>
        …
        <Expression _id="_Y3h_oGKXEeCGgJeNNV-Q1g"
            base="_16_6_1_2430132_1277959482285_387640_1270" type="method-invocation"
            detail="$moviePreference_CinemaDistance =
            org.me.android.context.preference.MoviePreferenceContext.getCinemaDistance()"/>
        …
    </Code:Class>
    …
</xmi:XMI>
```

ไฟล์กฎการแปลงทั้ง 4 ไฟล์ข้างต้นที่พัฒนาขึ้นในขั้นตอนนี้จะถูกนำไปใช้ในการพัฒนาแอปพลิเคชันแบบล่วงรู้บริบทด้วยกระบวนการแบบเอ็มดีเอในหัวข้อที่ 4.4 ต่อไป

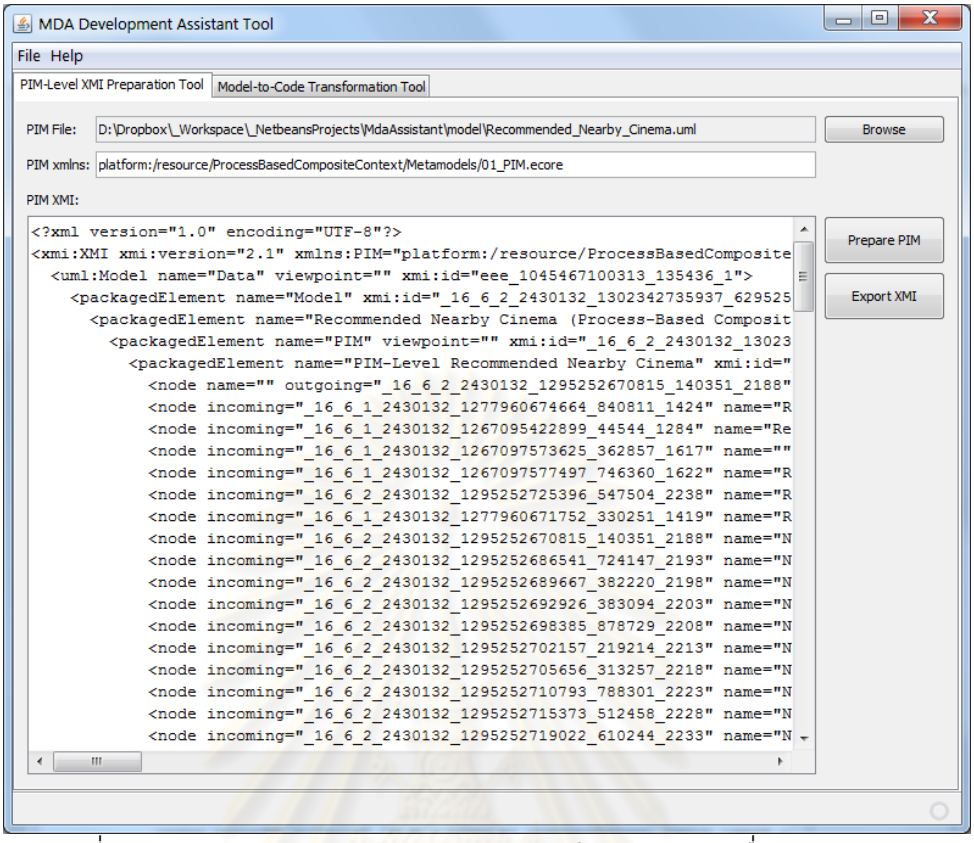
4.4 การพัฒนาแอปพลิเคชันต้นแบบ

การพัฒนาแอปพลิเคชันต้นแบบ (Application Development) ได้ทำการพัฒนาแอปพลิเคชันแบบล่วงรู้บริบทโดยการเรียกใช้เอพีไอของบริบทประกอบที่ได้สร้างขึ้น โดยเอพีไอของบริบทประกอบเป็นผลลัพธ์ของการประมวลผลแบบจำลองบริบทที่ได้พัฒนาขึ้นในหัวข้อที่ 4.1 ด้วยกฎการแปลงที่พัฒนาขึ้นในหัวข้อที่ 4.3 โดยใช้แบบจำลองเมตาที่พัฒนาขึ้นในหัวข้อที่ 4.2 ด้วยเครื่องประมวลผลภาษา ATL ตามภาพรวมของการทดสอบที่แสดงไว้ในรูปที่ 4.1

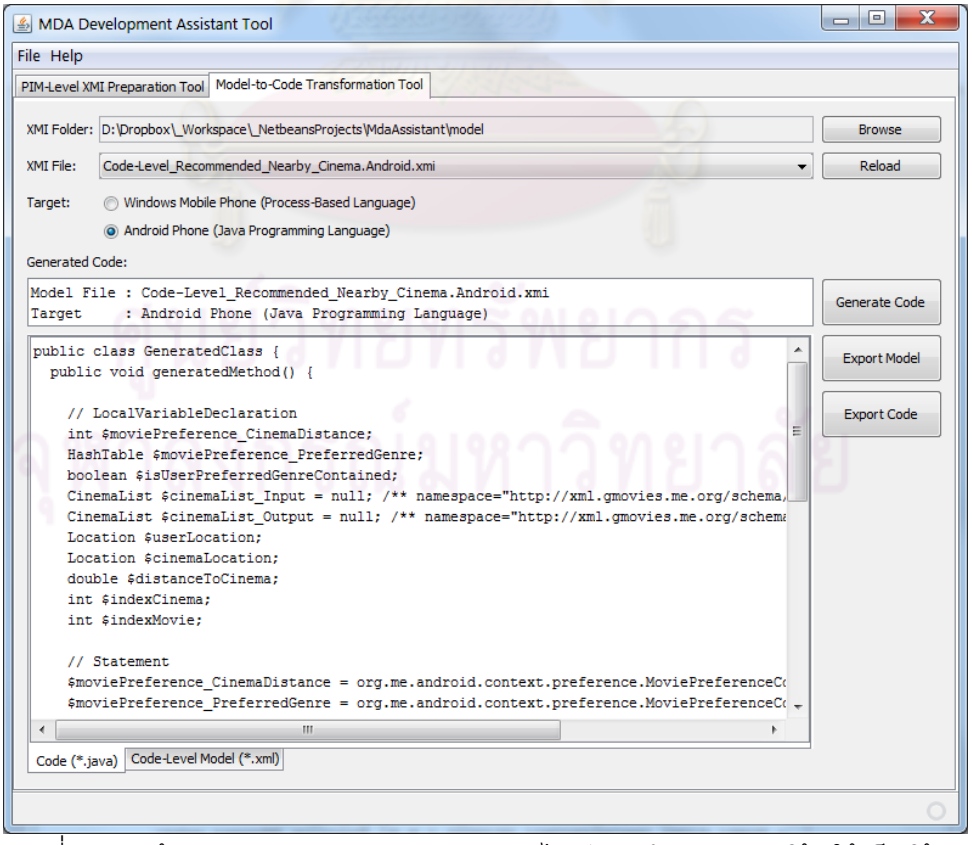
ในงานวิจัยนี้การใช้งานเครื่องประมวลผล ATL จำเป็นต้องมีการเตรียมไฟล์แบบจำลองระดับพีไอเอ็มให้อยู่ในรูปแบบที่เหมาะสม โดยลบข้อมูลส่วนที่ไม่สำคัญที่ไม่ได้ระบุในแบบจำลองเมตาออก และแก้ไขค่าเนมสเปซของยูเอ็มโปรไฟล์สำหรับแบบจำลองระดับพีไอเอ็มให้ตรงกับที่อยู่ของไฟล์แบบจำลองเมตาที่เครื่องประมวลมองเห็น นอกจากนั้นหลังจากทำการแปลงด้วยเครื่องประมวลผล ATL แล้วผลลัพธ์ที่ได้ยังอยู่ในรูปแบบของไฟล์ XMI ซึ่งยังไม่สามารถใช้งานได้โดยตรง ต้องจัดรูปแบบใหม่เสียก่อน ในงานวิจัยนี้จึงได้พัฒนาแอปพลิเคชัน “MDA Development Assistant Tool” ขึ้น เพื่ออำนวยความสะดวกสำหรับ 2 ขั้นตอนดังกล่าว ทั้งสำหรับจัดรูปแบบไฟล์ก่อนเข้าเครื่องประมวลผล และจัดรูปแบบของไฟล์แบบจำลองระดับโค้ดที่ได้จากเครื่องประมวลผลให้เป็นโค้ดที่ใช้งานได้อีกที โค้ดที่ได้นี้อาจยังไม่สมบูรณ์นัก อาจต้องมีการแก้ไขเพิ่มเติมอีกเล็กน้อย เช่น ในส่วนของการ import เป็นต้น ภาพรวมของการทดสอบเมื่อใช้แอปพลิเคชันดังกล่าวช่วยในการจัดการไฟล์แสดงได้ดังรูปที่ 4.10 แอปพลิเคชันที่พัฒนาขึ้นมีหน้าตาดังรูปที่ 4.11 - รูปที่ 4.12



รูปที่ 4.10 ภาพรวมของการทดสอบเมื่อใช้แอปพลิเคชันเพื่ออำนวยความสะดวกช่วยในการจัดการไฟล์



รูปที่ 4.11 หน้าจอแอปพลิเคชันส่วนจัดรูปแบบไฟล์ก่อนเข้าเครื่องประมวลผล



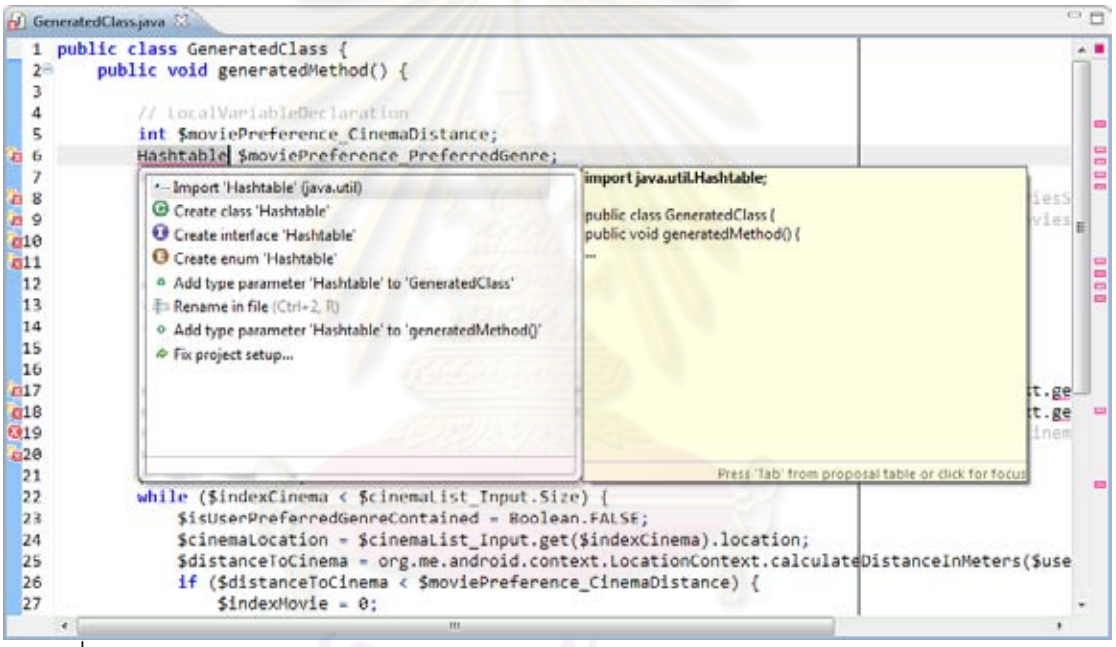
รูปที่ 4.12 หน้าจอแอปพลิเคชันส่วนจัดรูปแบบไฟล์แบบจำลองระดับโค้ดให้เป็นโค้ด

การทำงานของแอปพลิเคชันส่วนจัดรูปแบบไฟล์ก่อนเข้าเครื่องประมวลผล ทำโดยค้นหาแท็กของเอ็กซ์เอ็มแอลที่เก็บข้อมูลแบบจำลองไว้ทั้งในส่วนของโครงสร้าง (บรรทัดที่ 103 – 297 ในรูปที่ 4.13) และส่วนของโปรไฟล์ (บรรทัดที่ 301 – 319 ในรูปที่ 4.13) จากนั้นทำการลบข้อมูลส่วนอื่น ๆ ออก และแก้ไขค่าเนมสเปซของยูเอ็มโปรไฟล์ตามที่ได้ระบุไว้ในแอปพลิเคชันดังกล่าว (บรรทัดที่ 2 ในรูปที่ 4.13) ตัวอย่างของโครงสร้างไฟล์ที่ถูกจัดรูปแล้วแสดงได้ดังรูปที่ 4.13 โดยส่วนของโค้ดที่ถูกขีดเส้นทับคือโค้ดส่วนที่ถูกลบออกจากไฟล์

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <xmi:XMI xmi:version="2.1"
        xmlns:PIM="platform:/resource/ProcessBasedCompositeContext/Metamodels/01_PIM.ecore"
        ...>
3       <uml:Model ...>
…           <ownedComment .../>
…           <packageImport ...>  ...  </packageImport>
…           <packagedElement ...>  ...  </packagedElement>
…
101         <packagedElement xmi:type="uml:Package" ...>
102             <packagedElement xmi:type="uml:Package" ...>
103                 <packagedElement xmi:type="uml:Model" ...>
104                     <packagedElement xmi:type="uml:Activity" ...>
…                           <node ...>
…                           <edge ...>
…                           <group ...>
…                           ...
281                     </packagedElement>
…                       <profileApplication> ... </profileApplication>
297                 </packagedElement>
298             </packagedElement>
299         </packagedElement>
…
…           <profileApplication ...>  ...  </profileApplication>
…       </uml:Model>
…
…       <ModelTransformationProfile:map .../>
…       <MagicDrawProfile:auxiliaryResource .../>
…
301     <PIM:...>
302     <PIM:...>
…       ...
319     <PIM:...>
…
…   </xmi:XMI>
```

รูปที่ 4.13 โครงสร้างไฟล์แบบจำลองบริบทระดับพีไอเอ็มที่ถูกจัดรูปแบบก่อนเข้าเครื่องประมวลผล

เมื่อนำไฟล์แบบจำลองบริบทระดับพีไอเอ็มที่ถูกจัดรูปแบบแล้ว ไปเข้าเครื่องประมวลผลเพื่อทำการแปลงแบบจำลองจากระดับพีไอเอ็ม ไปเป็นแบบจำลองระดับพีเอสเอ็ม และแบบจำลองระดับโค้ดตามลำดับ แบบจำลองระดับโค้ดที่ได้เป็นผลลัพธ์ออกมายังคงอยู่ในรูปของไฟล์ XMI ที่ยังใช้งานโดยตรงไม่ได้ ต้องนำมาจัดรูปแบบใหม่ก่อนโดยในงานวิจัยนี้ใช้แอปพลิเคชันส่วนจัดรูปแบบไฟล์แบบจำลองระดับโค้ดให้เป็นโค้ดที่ได้พัฒนาขึ้นเป็นเครื่องมือช่วยในการจัดรูปแบบโค้ด โดยโค้ดในภาษาสำหรับบริบทเชิงกระบวนการสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ที่ได้จากการจัดรูปแบบด้วยแอปพลิเคชันนี้สามารถนำไปใช้งานได้โดยตรง แต่โค้ดในภาษาจาวาสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ที่ได้จากการจัดรูปแบบด้วยแอปพลิเคชันนี้อาจต้องได้รับการปรับแก้เพิ่มเติมอีกเล็กน้อย



รูปที่ 4.14 ตัวอย่างการปรับแก้โค้ดส่วน import ของโค้ดในภาษาจาวาด้วยโปรแกรม Eclipse

ส่วนที่ต้องปรับแก้เพิ่มเติมสำหรับโค้ดในภาษาจาวาที่ได้จากการจัดรูปแบบด้วยแอปพลิเคชันดังกล่าวมี 2 ส่วนหลัก ส่วนแรกคือส่วนของการประกาศ import สำหรับตัวแปรที่ไม่ใช่ตัวแปรพื้นฐานของภาษาจาวา ซึ่งได้แก่ตัวแปรประเภท Object ต่าง ๆ ซึ่งการปรับแก้ในส่วนนี้สามารถใช้โปรแกรม Eclipse ช่วยได้ โดยโปรแกรม Eclipse สามารถแสดงคำแนะนำส่วนของโค้ดที่จำเป็นต้องเพิ่มได้ดังตัวอย่างในรูปที่ 4.14 ส่วนที่สองคือโค้ดส่วนที่เรียกใช้เว็บเซอร์วิซ ซึ่งแอปพลิเคชันนี้แปลงออกมาเป็นคำสั่งง่าย ๆ ที่ยังใช้งานไม่ได้จริง ต้องเขียนโค้ดเพิ่มเติมโดยใส่รายละเอียดของเว็บเซอร์วิซ ซึ่งดูได้จาก WSDL ของเว็บเซอร์วิซนั้น ๆ

โค้ดด้านล่างนี้เป็นส่วนของโค้ดในภาษาจาวาที่ได้จากการจัดรูปแบบด้วยแอปพลิเคชันช่วยจัดรูปแบบไฟล์ที่ได้กล่าวไปข้างต้น โดยแสดงเฉพาะคำสั่งที่มีการเรียกเว็บเซอร์วิซ

```
…
$cinemaList_Input = call(http://Por-ISEL.myvnc.com:80/GMoviesService/GMoviesService,
    GetCinemaListByLocationName, {});
…
```

คำสั่งข้างต้นนั้นยังไม่สามารถใช้งานได้จริง ต้องทำการเขียนโค้ดเพิ่มเติมโดยใส่รายละเอียดต่าง ๆ ของการเรียกใช้งานเว็บเซอร์วิซ ซึ่งสามารถเขียนโค้ดได้ดังโค้ดด้านล่างนี้

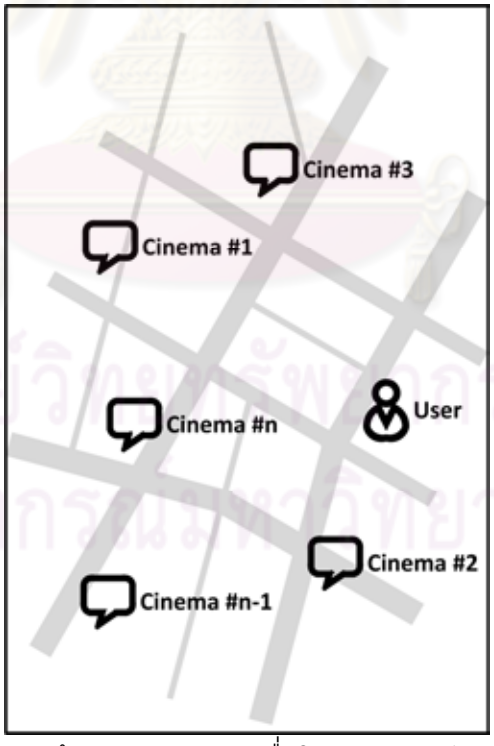
```
…

String wsUrl = "http://Por-ISEL.myvnc.com:80/GMoviesService/GMoviesService";
String wsNamespace = "http://xml.gmovies.me.org/schema/GMoviesService/";
String wsObjectRequest "GetCinemaListByLocationNameRequest";,
String wsObjectResponse "GetCinemaListByLocationNameResponse";
String wsSoapAction = "GetCinemaListByLocationName";

SoapObject request = null;
SoapObject response = null;
SoapSerializationEnvelope envelope = null;
AndroidHttpTransport androidHttpTransport = null;

try {
    request = new SoapObject(wsNamespace, wsObjectRequest);
    request.addProperty("LocationName", "Thailand");
    request.addProperty("Language", "English");

    envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
    envelope.setOutputSoapObject(request);

    androidHttpTransport = new AndroidHttpTransport(wsUrl);
    androidHttpTransport.call(wsSoapAction, envelope);

    response = new SoapObject(wsNamespace, wsObjectResponse);
    response = (SoapObject) envelope.getResponse();

    // do soap object extraction to get data
    $cinemaList_Input = MovieContext.extractCinemaListFromSoapObject(response);
}
catch (final SoapFault e) { throw new SoapFault(); }
catch (final IOException e) { throw new IOException(); }
catch (final XmlPullParserException e) { throw new XmlPullParserException(""); }

…
…
```

อย่างไรก็ตามส่วนที่ต้องปรับแก้ทั้งสองส่วนที่ได้กล่าวไปแล้วนี้เป็นเพียงข้อจำกัดของแอปพลิเคชันเพื่อช่วยจัดรูปแบบที่ได้พัฒนาขึ้นในงานวิจัยนี้เท่านั้น

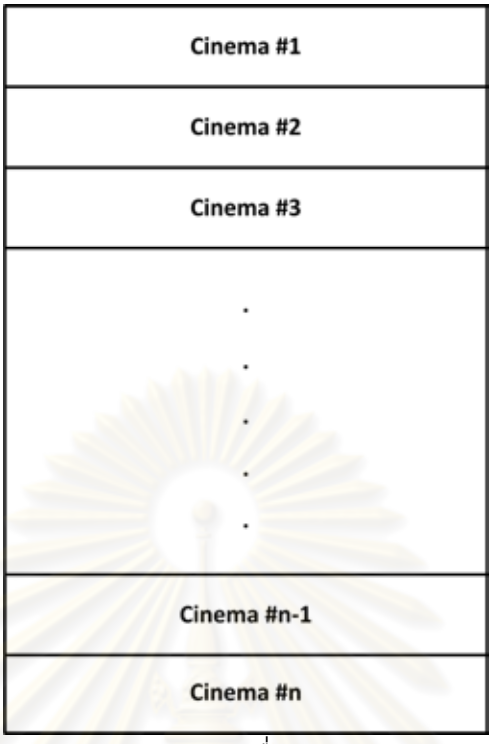
การออกแบบแอปพลิเคชันแบบล่วงรู้บริบทต้นแบบเพื่อทดสอบการเรียกใช้งานเอพีไอต้นแบบสำหรับให้ค่าบริบทประกอบเชิงกระบวนการที่ได้พัฒนาขึ้นโดยกระบวนการแบบเอ็มดีเอ กำหนดให้แอปพลิเคชันมีความต้องการเชิงหน้าที่ดังนี้

1) มีการเรียกใช้งานเอพีไอของบริบทประกอบเชิงกระบวนการ Recommended Nearby Cinema
2) มีส่วนแสดงผลค่าบริบทประกอบเชิงกระบวนการที่ได้จากการเรียกใช้งานเอพีไอต้นแบบ ซึ่งคือ รายชื่อโรงภาพยนตร์ ให้ผู้ใช้งาน
3) มีส่วนแสดงผลข้อมูลของแต่ละโรงภาพยนตร์ พร้อมทั้งรายการภาพยนตร์ที่ฉายอยู่
4) มีส่วนสำหรับระบุความพึงใจของผู้ใช้ ประกอบด้วย ประเภทของภาพยนตร์ที่ชื่นชอบ และ ระยะทางที่สามารถยอมรับได้

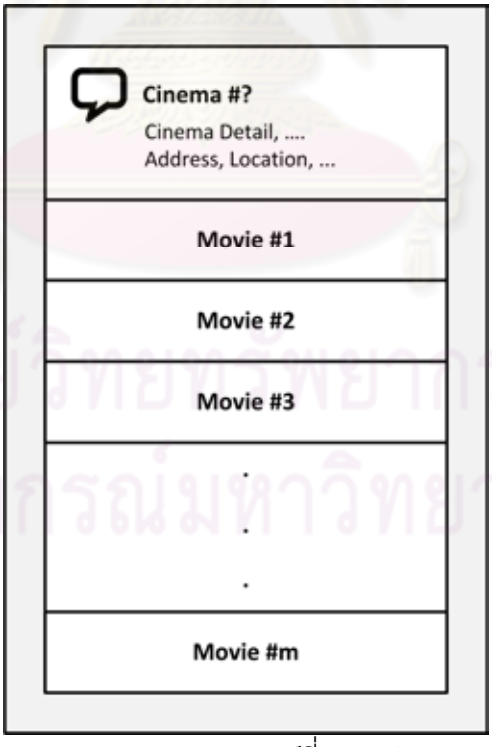
จากความต้องการเชิงหน้าที่ที่ได้กำหนดไว้ ผู้วิจัยจึงได้ออกแบบหน้าตาของแอปพลิเคชันไว้ดังรูปที่ 4.15 - รูปที่ 4.18



รูปที่ 4.15 หน้าจอแสดงผลรายชื่อโรงภาพยนตร์แบบแผนที่

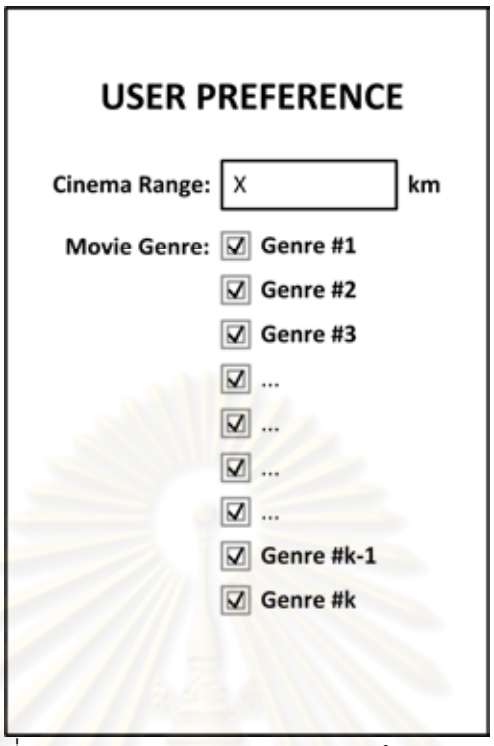


รูปที่ 4.16 หน้าจอแสดงผลรายชื่อโรงภาพยนตร์แบบรายการ



รูปที่ 4.17 หน้าจอแสดงรายการภาพยนตร์ที่ฉายอยู่ของแต่ละโรงภาพยนตร์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



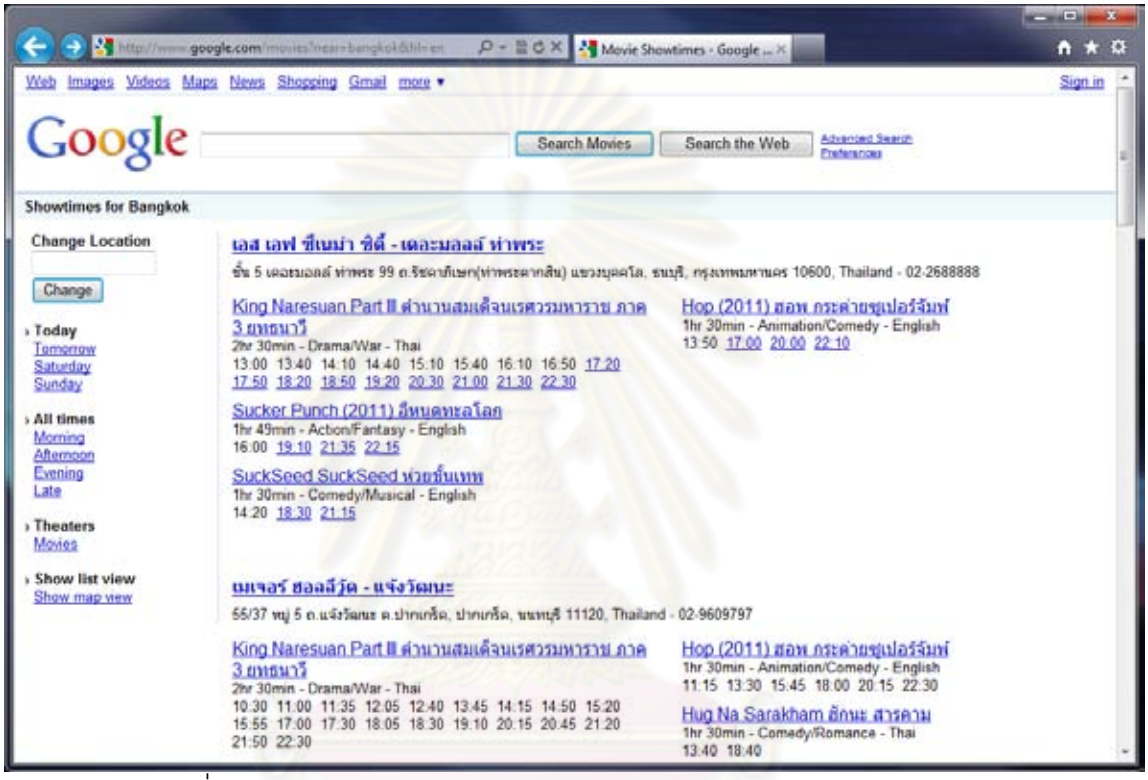
รูปที่ 4.18 หน้าจอสำหรับระบุความพึงใจของผู้ใช้

รายละเอียดการพัฒนาส่วนเอพีไอและแอปพลิเคชันสำหรับโทรศัพท์เคลื่อนที่แต่ละประเภทอยู่ในหัวข้อที่ 4.4.1 - 4.4.2 โดยที่งานวิจัยนี้ได้พัฒนาแอปพลิเคชันเพื่อทดสอบบนโทรศัพท์เคลื่อนที่ 2 ประเภท รายละเอียดของอุปกรณ์ที่ใช้ทดสอบและชุดเครื่องมือที่ใช้ในการพัฒนามีดังนี้คือ

1) โทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์
   - อุปกรณ์ที่ใช้ทดสอบคือ Asus P525 ซึ่งติดตั้งระบบปฏิบัติการวินโดวส์โมบายล์ เวอร์ชัน 5 พร้อมด้วยอุปกรณ์เสริม RoyalTek RBT-2110 Bluetooth GPS
2) โทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์
   - อุปกรณ์ที่ใช้ทดสอบคือ Google Nexus One ซึ่งติดตั้งระบบปฏิบัติการแอนดรอยด์ เวอร์ชัน 2.3.3 (Ginger Bread)

ในส่วนของเซอร์วิซให้บริการข้อมูลรายชื่อโรงภาพยนตร์พร้อมทั้งรายละเอียด ผู้วิจัยได้พัฒนาเว็บเซอร์วิซสำหรับให้บริการข้อมูลดังกล่าวขึ้น เนื่องจากไม่สามารถหาตัวอย่างของเซอร์วิซที่ให้บริการข้อมูลดังกล่าวสำหรับประเทศไทยได้ เว็บเซอร์วิซที่ได้พัฒนาขึ้นนี้ใช้ข้อมูลที่ได้จากการดึงข้อมูลจากหน้าเว็บ Google Movies (http://www.google.com/movies) โดยมีตัวอย่างของ

หน้าเว็บดังรูปที่ 4.19 โดยการใช้งานหน้าเว็บ Google Movies การระบุคำค้นสถานที่ไม่สามารถระบุเป็นชื่อประเทศ เช่น “Thailand” ได้ ต้องระบุเป็นเมืองหรือเขตย่อย ๆ ลงมา เช่น “Bangkok” เป็นต้น งานวิจัยนี้จึงใช้คำค้นข้อมูลเพื่อเป็นตัวแทนสำหรับสถานที่กระจายทั่วทุกภูมิภาคในประเทศไทย 7 แห่งได้แก่ “Bangkok”, “Chon Buri”, “Chiang Mai”, “Hua Hin”, “Nakhon Ratchasima”, “Phuket”, “Ubon Ratchathani”



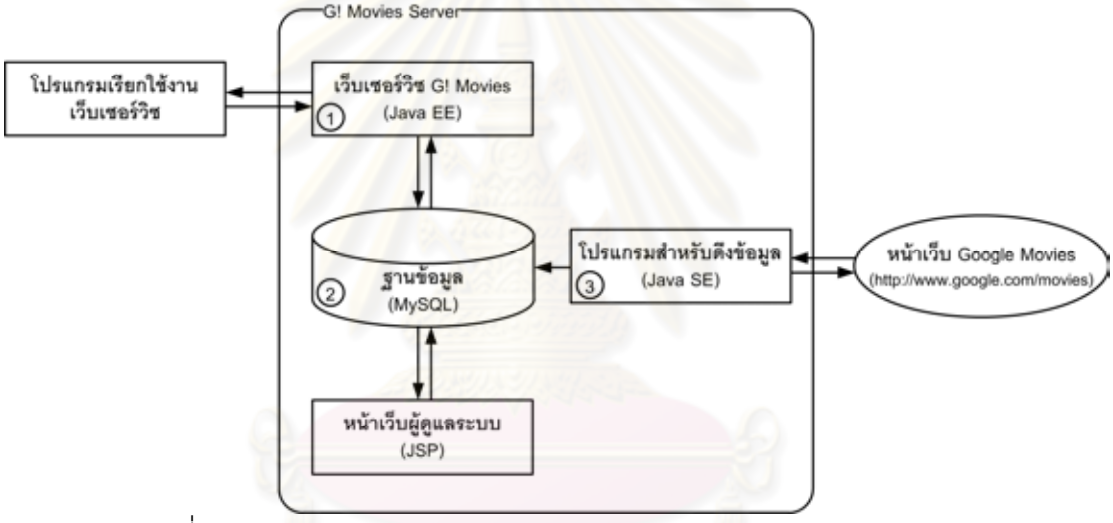
รูปที่ 4.19 ตัวอย่างหน้าเว็บ Google Movies โดยใช้คำค้น “Bangkok”

ระบบเว็บเซอร์วิซที่พัฒนาขึ้นให้ชื่อว่า “G! Movies Webservice” ประกอบด้วยส่วนต่าง ๆ ซึ่งแสดงแผนภาพสรุปส่วนประกอบของระบบเว็บเซอร์วิซโดยสังเขปได้ดังรูปที่ 4.20 โดยมีรายละเอียดดังนี้

1) เว็บเซอร์วิซถูกพัฒนาขึ้นโดยใช้ภาษา Java EE ทำงานอยู่บน Glassfish Server 2.2.1 มีชุดเครื่องมือที่ใช้ในการพัฒนาคือ NetBeans IDE 6.7.1 และ Java EE SDK ให้บริการในรูปแบบของเว็บเซอร์วิซแบบ SOAP โดยมีตัวอย่างของ Request Object และ Response Object อยู่ในรูปที่ 4.21 และ รูปที่ 4.22 ตามลำดับ ซึ่งข้อมูลที่ต้องระบุในส่วนของ Request Object ประกอบด้วยชื่อสถานที่ และภาษาที่ต้องการ (บรรทัดที่ 7 - 8 ในรูปที่ 4.21) โดยสถานที่ในที่นี้หมายถึงรายละเอียดที่มีปรากฏในส่วนของที่อยู่ เช่น ในรูปตัวอย่างใส่ว่า “Thailand” หมายความว่าให้เว็บเซอร์วิซคืนค่า

รายการโรงภาพยนตร์ที่มีข้อมูลที่อยู่ระบุว่า “Thailand” ส่วนข้อมูลใน Response Object ที่ได้รับจากเว็บเซอร์วิซ ประกอบด้วยรายชื่อโรงภาพยนตร์ต่าง ๆ ซึ่งในแต่ละรายการของโรงภาพยนตร์ประกอบด้วยรายละเอียดของโรงภาพยนตร์ทั้งชื่อ ที่อยู่ ละติจูด ลองจิจูด (บรรทัดที่ 6 - 9 ในรูปที่ 4.22) และรายการภาพยนตร์ที่ฉายอยู่ในวันนั้น ๆ (บรรทัดที่ 10 - 39 ในรูปที่ 4.22) โดยมีรายละเอียดสำหรับการเรียกใช้งานดังนี้

- URL: *http://Por-ISEL.myvnc.com/GMoviesService/GMoviesService*
- WSDL: *http://Por-ISEL.myvnc.com/GMoviesService/GMoviesService?wsdl*
- SOAP Action: *GetCinemaListByLocationName*



รูปที่ 4.20 ส่วนประกอบของระบบเว็บเซอร์วิซ “G! Movies Webservice”

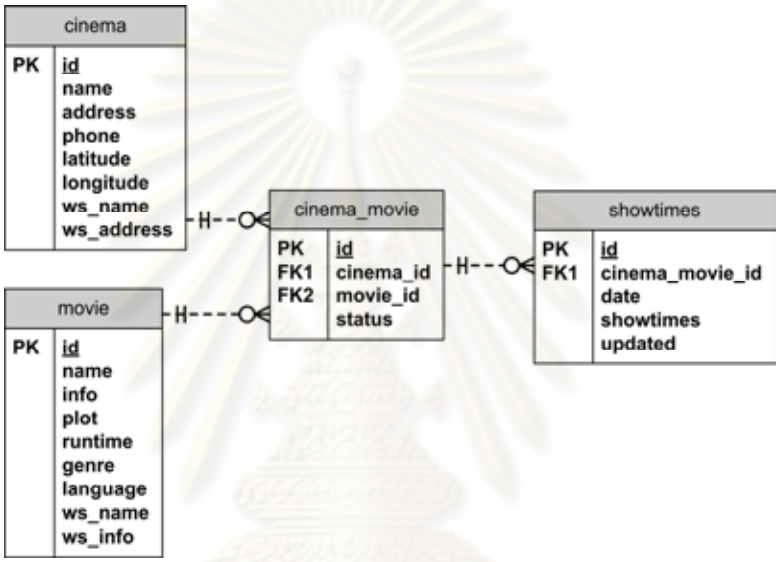
```
1  <soapenv:Envelope
2    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
3    xmlns:gmov="http://xml.gmovies.me.org/schema/GMoviesService/">
4     <soapenv:Header/>
5     <soapenv:Body>
6        <gmov:GetCinemaListByLocationNameRequest>
7           <gmov:LocationName>Thailand</gmov:LocationName>
8           <gmov:Language>English</gmov:Language>
9        </gmov:GetCinemaListByLocationNameRequest>
10    </soapenv:Body>
11 </soapenv:Envelope>
```

รูปที่ 4.21 ตัวอย่างของ Request Object สำหรับใช้เรียกเว็บเซอร์วิซที่พัฒนาขึ้น

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <GetCinemaListByLocationNameResponse
            xmlns="http://xml.gmovies.me.org/schema/GMoviesService/">
         <CinemaList>
            <Cinema>
               <Name>SF Cinema City - The Mall Thaphra</Name>
               <Address>The Mall Thaphra (5th Floor), 99 Ratchadapisek Road (Thaphra
                  Taksin), Bukkhalo, Thonburi, Bangkok, 10600, Thailand</Address>
               <Latitude>13.713619</Latitude>
               <Longitude>100.480273</Longitude>
               <MovieList>
                  <Movie>
                     <Name>SuckSeed</Name>
                     <Runtime>1hr 30min</Runtime>
                     <Genre>Comedy/Musical</Genre>
                     <Language>English</Language>
                     <Showtimes>11:10 - 14:20 - 18:30 - 21:15</Showtimes>
                  </Movie>
                  <Movie>
                     <Name>Sucker Punch (2011)</Name>
                     <Runtime>1hr 30min</Runtime>
                     <Genre>Action/Fantasy</Genre>
                     <Language>English</Language>
                     <Showtimes>11:50 - 16:00 - 19:10 - 21:35 - 22:15</Showtimes>
                  </Movie>
                  <Movie>
                     <Name>King Naresuan Part III</Name>
                     <Runtime>2hr 30min</Runtime>
                     <Genre>Drama/War</Genre>
                     <Language>Thai</Language>
                     <Showtimes>11:15 - 14:10 - 17:20 - 20:30</Showtimes>
                  </Movie>
                  <Movie>
                     <Name>Hop (2011)</Name>
                     <Runtime>1hr 30min</Runtime>
                     <Genre>Animation/Comedy</Genre>
                     <Language>English</Language>
                     <Showtimes>11:30 - 13:50 - 17:00 - 20:00 - 22:10</Showtimes>
                  </Movie>
               </MovieList>
            </Cinema>
            ...
            ...
            ...
         </CinemaList>
      </GetCinemaListByLocationNameResponse>
   </S:Body>
</S:Envelope>
```

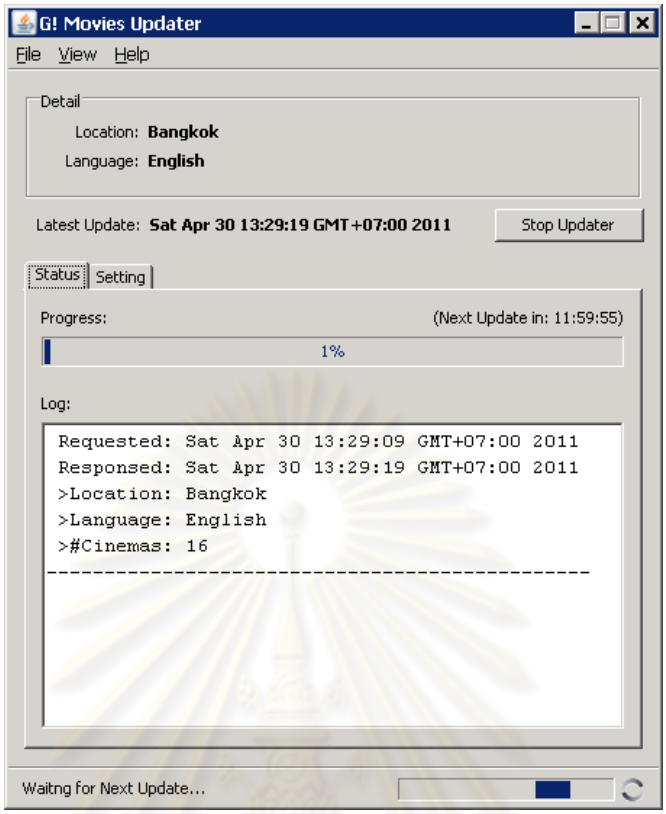
รูปที่ 4.22 ตัวอย่างของ Response Object ที่ได้จากเว็บเซอร์วิซที่พัฒนาขึ้น

2) ฐานข้อมูลของระบบเว็บเซอร์วิซ “G! Movies Webservice” พัฒนาโดยใช้ MySQL 5.1 โดยออกแบบให้มี 4 ตาราง มีแผนภาพอีอาร์ตามรูปที่ 4.23 ประกอบด้วย
   - cinema ใช้สำหรับเก็บข้อมูลเกี่ยวกับโรงภาพยนตร์
   - movie ใช้สำหรับเก็บข้อมูลภาพยนตร์
   - cinema_movie ใช้สำหรับเก็บข้อมูลความสัมพันธ์ระหว่างโรงภาพยนตร์และภาพยนตร์
   - showtimes ใช้สำหรับเก็บข้อมูลตารางการฉายภาพยนตร์

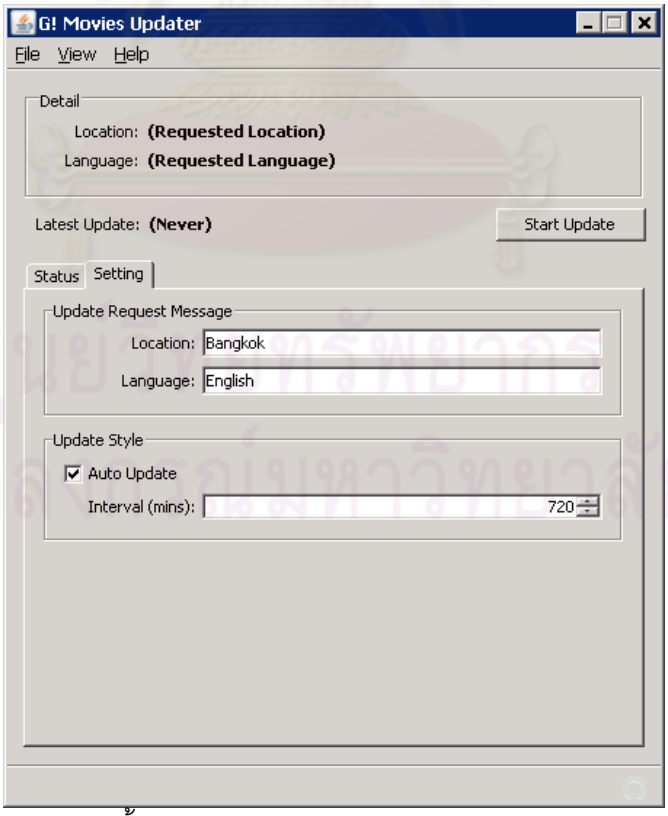


รูปที่ 4.23 แผนภาพอีอาร์ของฐานข้อมูลของระบบเว็บเซอร์วิซ “G! Movies Webservice”

3) แอปพลิเคชันสำหรับดึงข้อมูลมาเก็บไว้ในฐานข้อมูล หรือ “G! Movies Updater” พัฒนาโดยใช้ภาษา Java SE มีชุดเครื่องมือที่ใช้ในการพัฒนาคือ NetBeans IDE 6.7.1 และ Java SE SDK โดยมีหน้าที่ดึงข้อมูลจากหน้าเว็บ Google Movies มาเก็บไว้ในฐานข้อมูลเพื่อให้เว็บเซอร์วิซ “G! Movies Webservice” เรียกไปใช้งานอีกที แอปพลิเคชันมีหน้าตาดังรูปที่ 4.24 - รูปที่ 4.25 การทำงานภายในของแอปพลิเคชันนี้เริ่มจากส่งคำร้องขอค้นหาข้อมูลไปยังเว็บ Google Movies โดยใช้คำค้นสถานที่ตามที่ได้ระบุไว้ในแอปพลิเคชันนี้ ข้อมูลที่ได้รับตอบกลับมานั้นอยู่ในรูปแบบของไฟล์ HTML จากนั้นทำการสกัดข้อมูลที่ต้องการออกมา และนำข้อมูลที่ได้ไปบันทึกใส่ไว้ในฐานข้อมูล ถ้าผลจากการค้นหามีมากกว่าหนึ่งหน้าจะทำการวนซ้ำจนครบจำนวนหน้าทั้งหมด



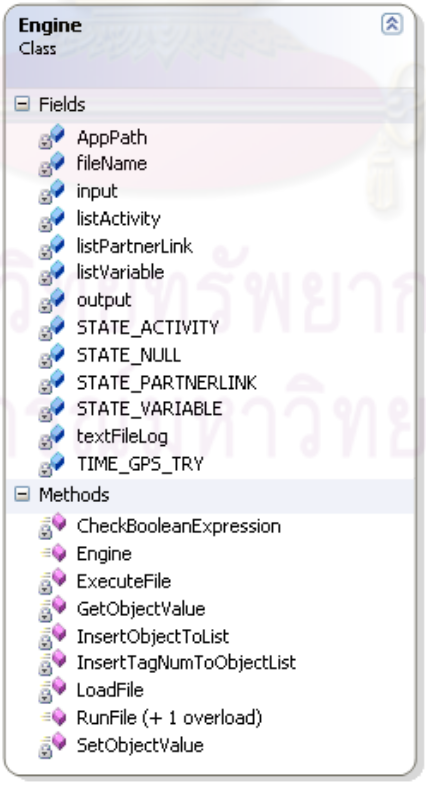
รูปที่ 4.24 หน้าจอระหว่างการทำงานของแอปพลิเคชันสำหรับดึงข้อมูลมาเก็บไว้ในฐานข้อมูล



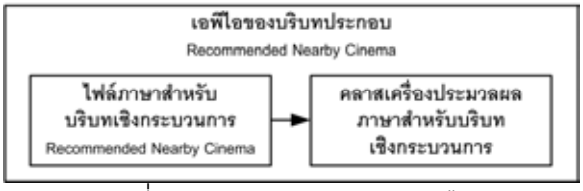
รูปที่ 4.25 หน้าจอการตั้งค่าของแอปพลิเคชันสำหรับดึงข้อมูลมาเก็บไว้ในฐานข้อมูล

4.4.1 การพัฒนาแอปพลิเคชันต้นแบบสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์

การพัฒนาแอปพลิเคชันสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ในงานวิจัยนี้ใช้ภาษา C# บน .NET Compact Framework 3.5 เป็นหลัก เครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันประกอบด้วย โปรแกรม Visual Studio 2008 และ Windows Mobile 5.0 SDK พร้อมทั้งชุดไลบรารีเสริม Smart Device Framework [27] สำหรับช่วยในการแสดงผลแบบรายการ ชุดไลบรารีเสริม GMap.NET [28] สำหรับช่วยในการแสดงผลแบบแผนที่ และชุด API สำหรับเรียกข้อมูลบริบทบนโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์จากงานวิจัยของ Hongpisuttikul และ Sukapirom [29] โดยที่การพัฒนาในส่วนของเอพีไอต้นแบบใช้ภาษาสำหรับบริบทเชิงกระบวนการ (Process-Based Context Language) ที่ผู้วิจัยได้ออกแบบไว้ในงานวิจัยก่อนหน้า [10] ภาษาสำหรับบริบทเชิงกระบวนการเป็นภาษาที่อยู่ในรูปของข้อกำหนดเอ็กซ์เอ็มแอล โดยมีสกีมาดังรูปที่ 3.5 โดยผู้วิจัยได้พัฒนาเครื่องประมวลผลภาษาสำหรับบริบทเชิงกระบวนการ (Process-Based Context Execution Engine) เพื่อใช้สำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ไว้ในงานวิจัยดังกล่าวด้วย โดยมีรายละเอียดเครื่องประมวลผลภาษาสำหรับบริบทเชิงกระบวนการดังรูปที่ 4.26 มีแผนภาพภายในเอพีไอดังรูปที่ 4.27 และตัวอย่างการเรียกใช้งานดังรูปที่ 4.28



รูปที่ 4.26 คลาสเครื่องประมวลผลภาษาสำหรับบริบทเชิงกระบวนการ



รูปที่ 4.27 แผนภาพภายในเอพีไอ

```
…
    Engine engine = new Engine();
    return engine.Runfile(“RecommendedNearbyCinema.xml”);
…
```

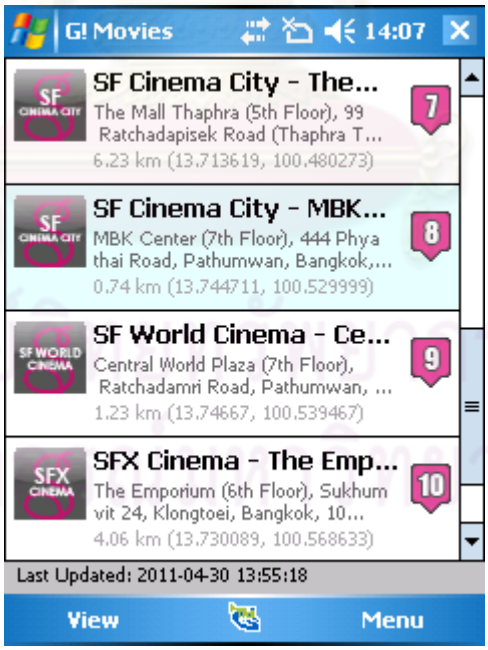
รูปที่ 4.28 ตัวอย่างโค้ดการเรียกใช้งานคลาสเครื่องประมวลผล

การพัฒนาเอพีไอทำตามกระบวนการที่แสดงไว้ในรูปที่ 4.1 ผลลัพธ์จากการพัฒนาเอพีไอโดยกระบวนการแบบเอ็มดีเออยู่ในรูปของแบบจำลองระดับโค้ด จากนั้นนำแบบจำลองระดับโค้ดดังกล่าวไปจัดรูปแบบใหม่ให้อยู่ในรูปของไฟล์เอ็กซ์เอ็มแอลที่เขียนด้วยภาษากระแสบริบท ด้วยแอปพลิเคชัน “MDA Development Assistant Tool” ที่กล่าวไว้ในส่วนต้นของหัวข้อ 4.4 เพื่อให้ได้โค้ดสำหรับนำไปใช้ร่วมกับคลาสเครื่องประมวลผลภาษาสำหรับบริบทเชิงกระบวนการ เพื่อสร้างเป็นเอพีไอเพื่อให้ค่าบริบทประกอบเชิงกระบวนการ Recommended Nearby Cinema สำหรับให้แอปพลิเคชันแบบล่วงรู้บริบทบนโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์มาเรียกใช้งาน

แอปพลิเคชันแบบล่วงรู้บริบทต้นแบบที่พัฒนาได้สำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์มีความสามารถตามที่ออกแบบไว้ในความต้องการเชิงหน้าที่ โดยเรียกใช้เอพีไอที่สร้างขึ้นโดยกระบวนการแบบเอ็มดีเอ และนำข้อมูลบริบทประกอบที่ได้มาแสดงผลได้อย่างถูกต้อง แอปพลิเคชันต้นแบบสำหรับวินโดวส์โมบายล์ประกอบด้วยหน้าจอ 4 หน้า รายละเอียดและตัวอย่างหน้าจอดังนี้

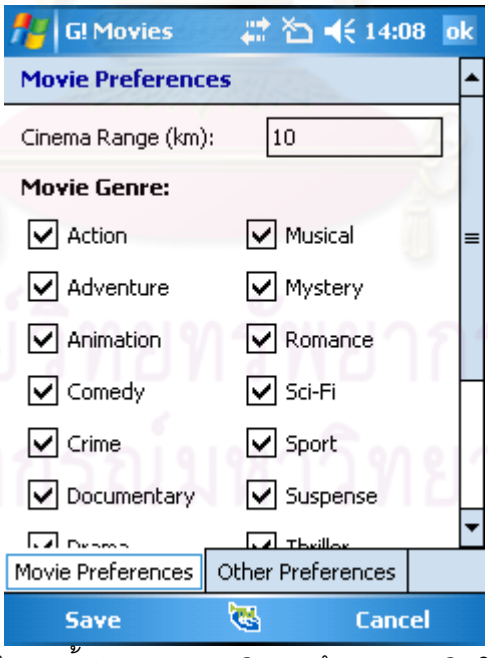
- Map View หน้าจอสำหรับแสดงผลรายชื่อโรงภาพยนตร์แบบแผนที่ (รูปที่ 4.29)
- List View หน้าจอสำหรับแสดงผลรายชื่อโรงภาพยนตร์แบบรายการ (รูปที่ 4.30)
- Cinema Detail View หน้าจอสำหรับแสดงรายการภาพยนตร์ที่ฉายอยู่ของแต่ละโรงภาพยนตร์ (รูปที่ 4.31)
- Preference Setting หน้าจอสำหรับปรับเปลี่ยนความพึงใจของผู้ใช้ (User Preference) (รูปที่ 4.32)

รูปที่ 4.29 หน้าจอแบบแผนที่ของแอปพลิเคชันต้นแบบบนวินโดวส์โมบายล์



รูปที่ 4.30 หน้าจอแบบรายการของแอปพลิเคชันต้นแบบบนวินโดวส์โมบายล์

รูปที่ 4.31 หน้าจอแสดงรายการภาพยนตร์ที่ฉายอยู่ของแต่ละโรงภาพยนตร์ของแอปพลิเคชันต้นแบบบนวินโดวส์โมบายล์



รูปที่ 4.32 หน้าจอตั้งค่าของแอปพลิเคชันต้นแบบบนวินโดวส์โมบายล์

4.4.2 การพัฒนาแอปพลิเคชันต้นแบบสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

การพัฒนาแอปพลิเคชันสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ในงานวิจัยนี้ใช้ภาษา Java เป็นหลัก เครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันประกอบด้วย โปรแกรม Eclipse 3.6 (Helios) และ Android SDK พร้อมทั้งไลบรารีเสริม ksoap2-android [30] สำหรับช่วยในการเรียกใช้งานเว็บเซอร์วิซแบบ SOAP บนโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ ซึ่งเป็นไลบรารีที่ได้รับการปรับปรุงมาจาก kSOAP2 [31] อีกทีหนึ่ง

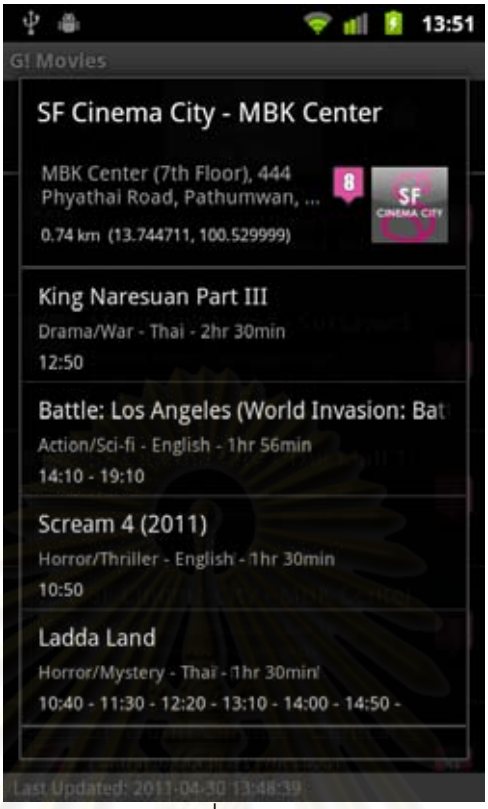
การพัฒนาเอพีไอทำตามกระบวนการที่แสดงไว้ในรูปที่ 4.1 ผลลัพธ์จากการพัฒนาเอพีไอโดยกระบวนการแบบเอ็มดีเออยู่ในรูปของแบบจำลองระดับโค้ด จากนั้นนำแบบจำลองดังกล่าวไปจัดรูปแบบใหม่ให้อยู่ในรูปของเมท็อดของคลาสด้วยแอปพลิเคชัน “MDA Development Assistant Tool” ที่กล่าวไว้ในส่วนต้นของหัวข้อ 4.4 เพื่อให้ได้โค้ด เพื่อสร้างเป็นเอพีไอเพื่อให้ค่าบริบทประกอบเชิงกระบวนการ Recommended Nearby Cinema สำหรับให้แอปพลิเคชันแบบล่วงรู้บริบทบนโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์มาเรียกใช้งาน

แอปพลิเคชันแบบล่วงรู้บริบทต้นแบบที่พัฒนาได้สำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์มีความสามารถตามที่ออกแบบไว้ในความต้องการเชิงหน้าที่ โดยเรียกใช้เอพีไอที่สร้างขึ้นโดยกระบวนการแบบเอ็มดีเอ และนำข้อมูลบริบทประกอบที่ได้มาแสดงผลได้อย่างถูกต้อง แอปพลิเคชันต้นแบบสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ประกอบด้วยหน้าจอ 4 หน้าตามที่ออกแบบไว้เช่นกัน รายละเอียดและตัวอย่างหน้าจอดังนี้

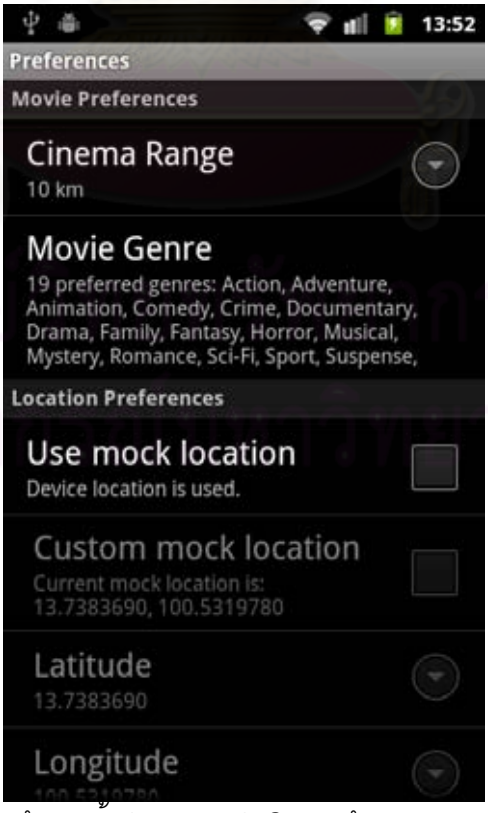
- Map View หน้าจอสำหรับแสดงผลรายชื่อโรงภาพยนตร์แบบแผนที่ (รูปที่ 4.33)
- List View หน้าจอสำหรับแสดงผลรายชื่อโรงภาพยนตร์แบบรายการ (รูปที่ 4.34)
- Cinema Detail View หน้าจอสำหรับแสดงรายการภาพยนตร์ที่ฉายอยู่ของแต่ละโรงภาพยนตร์ (รูปที่ 4.35)
- Preference Setting หน้าจอสำหรับปรับเปลี่ยนความพึงใจของผู้ใช้ (User Preference) (รูปที่ 4.36)

รูปที่ 4.33 หน้าจอแบบแผนที่ของแอปพลิเคชันต้นแบบบนแอนดรอยด์

รูปที่ 4.34 หน้าจอแบบรายการของแอปพลิเคชันต้นแบบบนแอนดรอยด์



รูปที่ 4.35 หน้าจอแสดงรายการภาพยนตร์ที่ฉายอยู่ของแต่ละโรงภาพยนตร์ของแอปพลิเคชันต้นแบบบนแอนดรอยด์



รูปที่ 4.36 หน้าจอตั้งค่าของแอปพลิเคชันต้นแบบบนแอนดรอยด์

ในบทนี้ได้ทดสอบการนำสิ่งที่ได้ออกแบบไว้ในบทที่ 3 ทั้งแบบจำลองเมตา และกฎการแปลง มาใช้ในการพัฒนาบริบทประกอบเชิงกระบวนการโดยอิงสถาปัตยกรรมอิงแบบจำลอง โดยผลของการพัฒนานำเสนอในรูปแบบของเอพีไอที่ให้ค่าบริบท ซึ่งในการทดสอบนี้ใช้บริบทต้นแบบ "Recommended Nearby Cinema" เป็นตัวอย่างของบริบทประกอบเชิงกระบวนการ โดยได้พัฒนาเอพีไอของบริบทดังกล่าวสำหรับโทรศัพท์เคลื่อนที่ที่ใช้ในการทดสอบ 2 แพลตฟอร์ม ได้แก่ โทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ และโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ นอกจากนั้นยังได้พัฒนาแอปพลิเคชันต้นแบบสำหรับแต่ละแพลตฟอร์มเพื่อทดสอบการเรียกใช้งานเอพีไอที่ได้พัฒนาขึ้นด้วย ซึ่งจากการทดสอบพบว่าสิ่งที่ได้ออกแบบไว้ในบทที่ 3 ทั้งแบบจำลองเมตา และกฎการแปลง สามารถนำมาใช้ในการพัฒนาบริบทประกอบเชิงกระบวนการโดยอิงสถาปัตยกรรมอิงแบบจำลองได้เป็นอย่างดี และการพัฒนาบริบทด้วยวิธีการดังกล่าวช่วยให้การพัฒนาบริบทสำหรับแพลตฟอร์มที่หลากหลายทำได้ง่ายขึ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5
บทสรุป

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอการสร้างบริบทประกอบรูปแบบใหม่โดยให้ชื่อว่า *บริบทประกอบเชิงกระบวนการ (Process-Based Composite Context)* โดยอาศัยแนวคิดจากการพัฒนาเชิงกระบวนการมาประยุกต์กับการคำนวณแบบล่วงรู้บริบท ซึ่งทำให้การพัฒนาบริบทประกอบมีประสิทธิภาพมากขึ้นโดยบริบทประกอบสามารถถูกสร้างขึ้นโดยการเชื่อมโยงการประมวลผลค่าบริบทอื่น ๆ ในลักษณะเชิงกระบวนการ ทำให้เห็นการทำงานของบริบทประกอบได้อย่างชัดเจนและสามารถเปลี่ยนแปลงบริบทที่เป็นส่วนประกอบได้โดยง่าย

นอกจากนั้นยังได้นำเสนอการพัฒนาบริบทประกอบเชิงกระบวนการโดยประยุกต์ใช้กับแนวคิดของเอ็มดีเอหรือสถาปัตยกรรมอิงแบบจำลอง ซึ่งสามารถทำให้การสร้างบริบทประกอบเชิงกระบวนการสำหรับแพลตฟอร์มที่หลากหลายทำได้โดยง่าย เมื่อจำเป็นต้องมีการเปลี่ยนแปลงบริบทที่เป็นส่วนประกอบก็สามารถทำได้โดยง่าย เพียงแก้ไขแบบจำลองในระดับพีไอเอ็มเท่านั้น โดยในงานวิจัยนี้ได้แสดงการออกแบบองค์ประกอบที่จำเป็นสำหรับการพัฒนาบริบทประกอบเชิงกระบวนการโดยอิงสถาปัตยกรรมอิงแบบจำลอง ทั้งแบบจำลองเมตา และกฎการแปลง รวมถึงแสดงตัวอย่างของแบบจำลองบริบทประกอบเชิงกระบวนการ การใช้งานแบบจำลองเมตา และกฎการแปลง และพัฒนาแอปพลิเคชันแบบล่วงรู้บริบทเพื่อทดสอบการใช้งานจริง

จากการทดสอบพบว่าสิ่งที่ออกแบบไว้สามารถนำไปประยุกต์ใช้งานได้จริง โดยแอปพลิเคชันที่ใช้เอพีไอของบริบทประกอบเชิงกระบวนการที่ได้จากการพัฒนาแบบเอ็มดีเอสามารถทำงานได้อย่างถูกต้อง อีกทั้งการพัฒนาแบบเอ็มดีเอ ยังช่วยลดเวลาการพัฒนาแอปพลิเคชันได้อีกด้วย โดยเฉพาะในกรณีที่ต้องการพัฒนาแอปพลิเคชันสำหรับหลายแพลตฟอร์ม เมื่อต้องการเปลี่ยนแปลงกระบวนการการทำงานของบริบทประกอบเชิงกระบวนการสามารถทำได้โดยเปลี่ยนในแบบจำลองระดับพีไอเอ็มเท่านั้น จากนั้นใช้เครื่องมือที่ได้พัฒนาขึ้นช่วยในการแปลงเป็นแบบจำลองระดับพีเอสเอ็มและโค้ดตามลำดับ ซึ่งทำให้ได้เอพีไอระดับโค้ดสำหรับหลายแพลตฟอร์มได้ทันที

5.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย

5.2.1 การพัฒนาโดยอิงสถาปัตยกรรมอิงแบบจำลองโดยทั่วไปนั้นนิยมเริ่มต้นแบบจำลองเมตาระดับพีไอเอ็มโดยใช้แผนภาพคลาสของยูเอ็มแอล ซึ่งมีการระบุกฎการเขียนแท็กอย่างชัดเจนทำให้ง่ายต่อการแปลงสู่แบบจำลองระดับรอง ๆ ลงไป ในขณะที่ในงานวิจัยนี้เริ่มต้นแบบจำลองเมตาระดับพีไอเอ็มโดยใช้แผนภาพกิจกรรมของยูเอ็มแอล ซึ่งมีข้อดีคือทำให้เห็นกระบวนการทำงานภายในได้อย่างชัดเจน อย่างไรก็ตามเนื่องจากแผนภาพกิจกรรมไม่ได้กำหนดรายละเอียดของแท็กไว้ กล่าวคือ โดยทั่วไปนิยมเขียนแค่ชื่อกิจกรรมเป็นภาษาพูดเพื่ออธิบายการทำงานนั้น ๆ เท่านั้นโดยไม่ได้ใส่ใจรายละเอียด ทำให้การนำแบบจำลองที่เขียนได้โดยรูปแบบปกติมาเข้าสู่กระบวนการแปลงแบบจำลองตามลำดับลงมาทำได้ยาก ในงานวิจัยนี้แก้ปัญหาโดยการสร้างข้อบังคับด้านไวยากรณ์สำหรับการเขียนรายละเอียดของแท็กของ Stereotype ระดับพีไอเอ็ม เพิ่มเติมเพื่อให้ง่ายต่อการแปลงแบบจำลอง

5.2.2 โปรแกรมเขียนแบบจำลองที่รองรับการเขียนโปรไฟล์ของแผนภาพกิจกรรมของยูเอ็มแอลมีไม่มากนัก โดยทั่วไปไม่สามารถกำหนด Stereotype ของแผนภาพกิจกรรมของยูเอ็มแอลได้ อย่างไรก็ตามโปรแกรม MagicDraw UML ที่ได้เลือกใช้ในงานวิจัยนี้เพื่อเขียนแบบจำลองระดับพีไอเอ็มนั้นสามารถรองรับการกำหนด Stereotype ของแผนภาพกิจกรรมได้

5.2.3 การเรียกค่าบริบทสำหรับโทรศัพท์เคลื่อนที่ประเภทต่าง ๆ นั้นสามารถเรียกได้ไม่เท่ากัน เนื่องจากการรองรับของตัวระบบปฏิบัติการนั้น ๆ บางกรณีถึงแม้ตัวเครื่องจะมีฮาร์ดแวร์ที่ให้ค่าบริบทได้ แต่ตัวระบบปฏิบัติการยังไม่รองรับก็สามารถเรียกใช้ได้ยาก ในงานวิจัยนี้จึงเลือกใช้โทรศัพท์เคลื่อนที่ที่สามารถให้ค่าบริบทที่จำเป็นสำหรับการทดสอบได้เหมือนกัน เพื่อให้ง่ายต่อการทดสอบ

5.2.4 การพัฒนาแอปพลิเคชันบนโทรศัพท์เคลื่อนที่บางประเภทจำกัดแต่บนระบบปฏิบัติการหนึ่ง ๆ เท่านั้น ในงานวิจัยนี้จึงไม่สามารถทดลองได้เนื่องจากขาดอุปกรณ์สำหรับใช้ในการพัฒนา ตัวอย่างของโทรศัพท์เคลื่อนที่ดังกล่าวได้แก่ โทรศัพท์เคลื่อนที่ประเภท iPhone ซึ่งใช้ระบบปฏิบัติการ iOS เนื่องจาก Xcode ซึ่งเป็นเครื่องมือที่จำเป็นในการพัฒนาโปรแกรมสำหรับระบบปฏิบัติการ iOS จำเป็นต้องติดตั้งบนเครื่องคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการ Mac OS เท่านั้น

5.2.5 ข้อมูลที่ต้องการที่เกี่ยวกับรายการโรงภาพยนตร์ไม่มีบริการในรูปแบบเว็บเซอร์วิซ มีแต่อยู่ในรูปแบบของหน้าเว็บปกติเท่านั้น ผู้วิจัยจึงแก้ปัญหาโดยการพัฒนาเว็บเซอร์วิซที่ให้บริการข้อมูลดังกล่าวขึ้นมาโดยเฉพาะ

5.3 ข้อเสนอแนะ

งานวิจัยนี้สามารถพัฒนาเพิ่มเติมได้โดยการพัฒนาแบบจำลองเมตาสำหรับแพลตฟอร์มที่หลากหลายขึ้น พัฒนากฎการแปลงให้มีประสิทธิภาพมากขึ้น เช่น ให้ครอบคลุมเอพีไอที่มีทั้งหมดในระบบ เนื่องจากในงานวิจัยนี้กฎการแปลงครอบคลุมเอพีไอเฉพาะตัวอย่างเอพีไอที่ถูกเรียกใช้ในแบบจำลองต้นแบบเท่านั้น พัฒนาแอปพลิเคชันสำหรับช่วยจัดรูปแบบไฟล์แบบจำลองระดับโค้ดให้เป็นโค้ดให้มีประสิทธิภาพมากขึ้น โดยลดขั้นตอนที่จำเป็นต้องปรับแก้ด้วยมือทั้งส่วนของ import และส่วนของการเรียกใช้งานเว็บเซอร์วิซสำหรับโค้ดในภาษาจาวา นอกจากนั้นยังสามารถพัฒนาเพิ่มเติมเพื่อให้รองรับการพัฒนาแบบจำลองแบบ Round-Trip Engineering ได้ กล่าวคือสามารถแปลงแบบจำลองจากระดับพีไอเอ็มไปเป็นแบบจำลองระดับโค้ด และแบบจำลองระดับโค้ดกลับไปยังแบบจำลองระดับพีไอเอ็ม กลับไปกลับมาได้ ซึ่งจะทำให้การพัฒนาบริบทประกอบเชิงกระบวนการสามารถทำได้ง่ายยิ่งขึ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] Mostéfaoui, G. K., Pasquier-Rocha, J., and Brézillon, P. Context-Aware Computing: A Guide for the Pervasive Computing Community. Proceedings of the IEEE/ACS International Conference on Pervasive Services 2004 (ICPS 2004), pp. 39-48. Beirut, Lebanon, July 19-23, 2004. New Jersey: IEEE, 2004.

[2] Dey, A. K. and Abowd, G. D. Towards a Better Understanding of Context and Context-Awareness. Proceedings of Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing System (CHI 2000), The Hague, The Netherlands, April 3, 2000. New York: ACM, 2000.

[3] Sheng, Q. Z. and Benatallah, B. ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services. Proceedings of the International Conference on Mobile Business 2005 (ICMB 2005), pp. 206-212. Sydney, Australia, July 11-13, 2005. New Jersey: IEEE, 2005.

[4] Han, J., Cho, Y., and Choi, J. A Workflow Language Based on Structural Context Model for Ubiquitous Computing. Proceedings of the International Conference on Embedded and Ubiquitous Computing 2005 (EUC 2005), pp. 879-889. Nagasaki, Japan, December 6-9, 2005. Berlin: Springer Berlin / Heidelberg, 2005.

[5] Abowd, G. D., Iftode, L., and Mitchell, H. Guest Editors' Introduction: The Smart Phone: A First Platform for Pervasive Computing. IEEE Pervasive Computing 4 (April-June 2005): 18-19.

[6] Ballagas, R., Borchers, J., Rohs, M., and Sheridan, J. G. The Smart Phone: A Ubiquitous Input Device. IEEE Pervasive Computing 5 (January-March 2006): 70-77.

[7] Raento, M., Oulasvirta, A., Petit, R., and Toivonen, H. ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. IEEE Pervasive Computing 4 (April-June 2005): 51-59.

[8] Object Management Group, Inc. Business Process Modeling Notation (BPMN) Version 1.1 [Online]. 2008. Available from: http://www.bpmn.org/Documents/BPMN_1-1_Specification.pdf [2011, March 31].

[9] OASIS. Web Services Business Process Execution Language (WSBPEL) Version 2.0 [Online]. 2007. Available from: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf [2011, March 31].

[10] Gultawatvichai, E. and Senivongse, T. A Framework for Context-Flow Composite Contexts for Mobile Phone Applications. Proceedings of The 12th National Computer Science and Engineering Conference (NCSEC 2008), pp. 478-482. Pattaya, Thailand, November 19-21, 2008.

[11] Gultawatvichai, E. and Senivongse, T. A Development of Process-Based Composite Contexts for Mobile Device Platforms Based on Model Driven Architecture. Proceedings of 2011 Eighth International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhon Pathom, Thailand, May 11-13, 2011.

[12] Dey, A. K., Salber, D., and Abowd, G. D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human Computer Interaction (HCI) 16 (2001): 97-166.

[13] Object Management Group, Inc. UML Version 2.2, Infrastructure Specification [Online]. 2009. Available from: http://www.omg.org/spec/UML/2.2/Infrastructure/PDF [2011, March 31].

[14] Object Management Group, Inc. UML Version 2.2, Superstructure Specification [Online]. 2009. Available from: http://www.omg.org/spec/UML/2.2/Superstructure/PDF [2011, March 31].

[15] Object Management Group, Inc. Requirements for UML Profiles (Green Paper) [Online]. 1999. Available from: http://www.omg.org/cgi-bin/doc?ad/99-12-32 [2011, March 31].

[16] Object Management Group, Inc. MDA Guide Version 1.0.1 [Online]. 2003. Available from: http://www.omg.org/cgi-bin/doc?omg/03-06-01 [2011, March 31].

[17] Modeling Amalgamation Project. Eclipse Modeling Tools (includes Incubating components) - Helios SR2 Packages [Online]. 2010. Available from: http://www.eclipse.org/downloads/packages/eclipse-modeling-tools-includes-incubating-components/heliossr2 [2011, March 31].

[18] Eclipse Modeling Project. Eclipse Modeling Project [Online]. 2010. Available from: http://www.eclipse.org/modeling/ [2011, March 31].

[19] ATL Project. ATL - a model transformation technology [Online]. 2010. Available from: http://www.eclipse.org/atl/ [2011, March 31].

[20] Henricksen, K., Indulska, J., and Rakotonirainy, A. Modeling Context Information in Pervasive Computing Systems. Proceedings of the First International Conference on Pervasive Computing (Pervasive 2002), pp. 167-180. Zürich, Switzerland, August 26-28, 2002. Berlin: Springer Berlin / Heidelberg, 2002.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

[21] Ayed, D. and Berbers, Y. UML Profile for the Design of a Platform-Independent Context-Aware Applications. Proceedings of the First Workshop on MOdel Driven Development for Middleware 2006 (MODDM 2006), pp. 1-5. Melbourne, Australia, November 27 - December 1, 2006. New York: ACM, 2006.

[22] Zimmer, T. and Beigl, M. AwareOffice: Integrating Modular Context-Aware Applications. Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops 2006 (ICDCSW 2006), pp. 59-64. Lisboa, Portugal, July 4-7, 2006. New Jersey: IEEE, 2006.

[23] Fahy, P. and Clarke, S. CASS - Middleware for Mobile Context-Aware Applications. Proceedings of the Second International Conference on Mobile Systems, Applications, and Service (MobiSys 2004) Workshop on Context Awareness, Boston, Massachusetts, USA, June 6-9, 2004. New York: ACM, 2004.

[24] Costa, P. D., Pires, L. F., Sinderen, M. v., and Filho, J. G. P. Towards a Services Platform for Mobile Context-Aware Applications. Proceedings of the 1st International Workshop on Ubiquitous Computing (IWUC 2004) in conjunction with the Sixth International Conference on Enterprise Information Systems (ICEIS 2004), pp. 48-61. Porto, Portugal, April 13-14, 2004. Portugal: INSTICC Press, 2004.

[25] Gosling, J., Joy, B., Steele, G., and Bracha, G. The Java™ Language Specification. Third ed. Boston: Addison-Wesley, 2005.

[26] No Magic, Inc. Introducing MagicDraw [Online]. 2010. Available from: http://www.magicdraw.com/what_is [2011, March 31].

[27] OpenNETCF Consulting, LLC. Smart Device Framework [Online]. 2009. Available from: http://www.opennetcf.com/Products/SmartDeviceFramework/tabid/65/Default.aspx [2011, March 31].

[28] radioman. GMap.NET - Great Maps for Windows Forms & Presentation [Online]. 2011. Available from: http://greatmaps.codeplex.com/ [2011, March 31].

[29] Hongpisuttikul, T. and Sukapirom, R., A Development of Context APIs for Pocket PC Phones. B.Eng. (Computer) Senior Project, Department of Computer Engineering, Chulalongkorn University, 2006.

[30] mosa...@gmail.com. ksoap2-android Project [Online]. 2011. Available from: http://code.google.com/p/ksoap2-android/ [2011, March 31].

[31] Haustein, S. and Seigel, J. kSOAP 2 [Online]. 2006. Available from: http://ksoap2.sourceforge.net/ [2011, March 31].

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

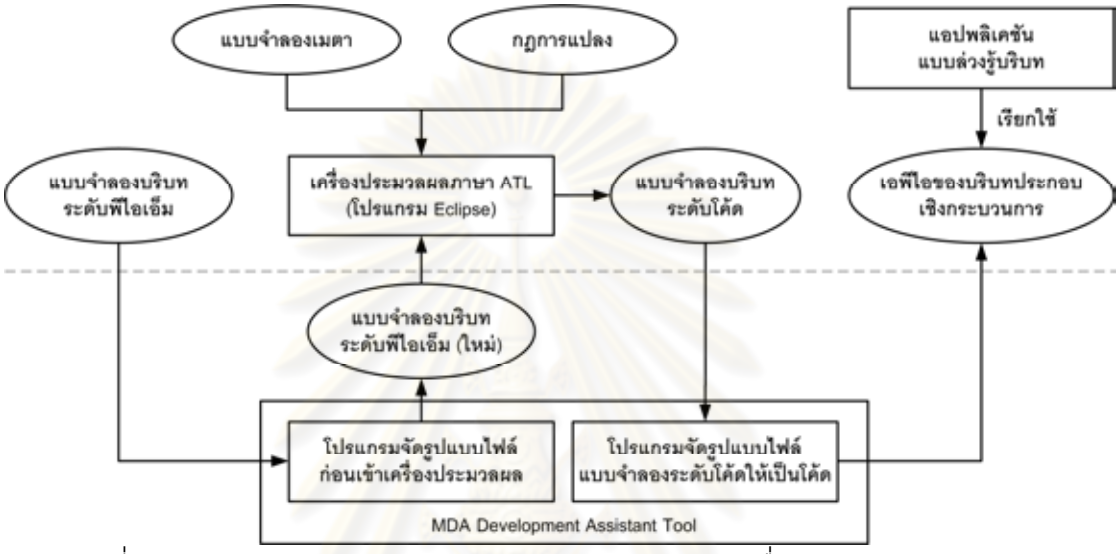
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

คู่มือการใช้งานแอปพลิเคชัน MDA Development Assistant Tool

แอปพลิเคชัน “MDA Development Assistant Tool” ถูกสร้างขึ้นเพื่อช่วยอำนวยความสะดวกในกระบวนการพัฒนาแอปพลิเคชันโดยอิงสถาปัตยกรรมอิงแบบจำลองตามกระบวนการที่กล่าวไว้ในงานวิจัยนี้ โดยสามารถแสดงขั้นตอนที่ใช้งานแอปพลิเคชันนี้ได้ดังรูปที่ ก.1

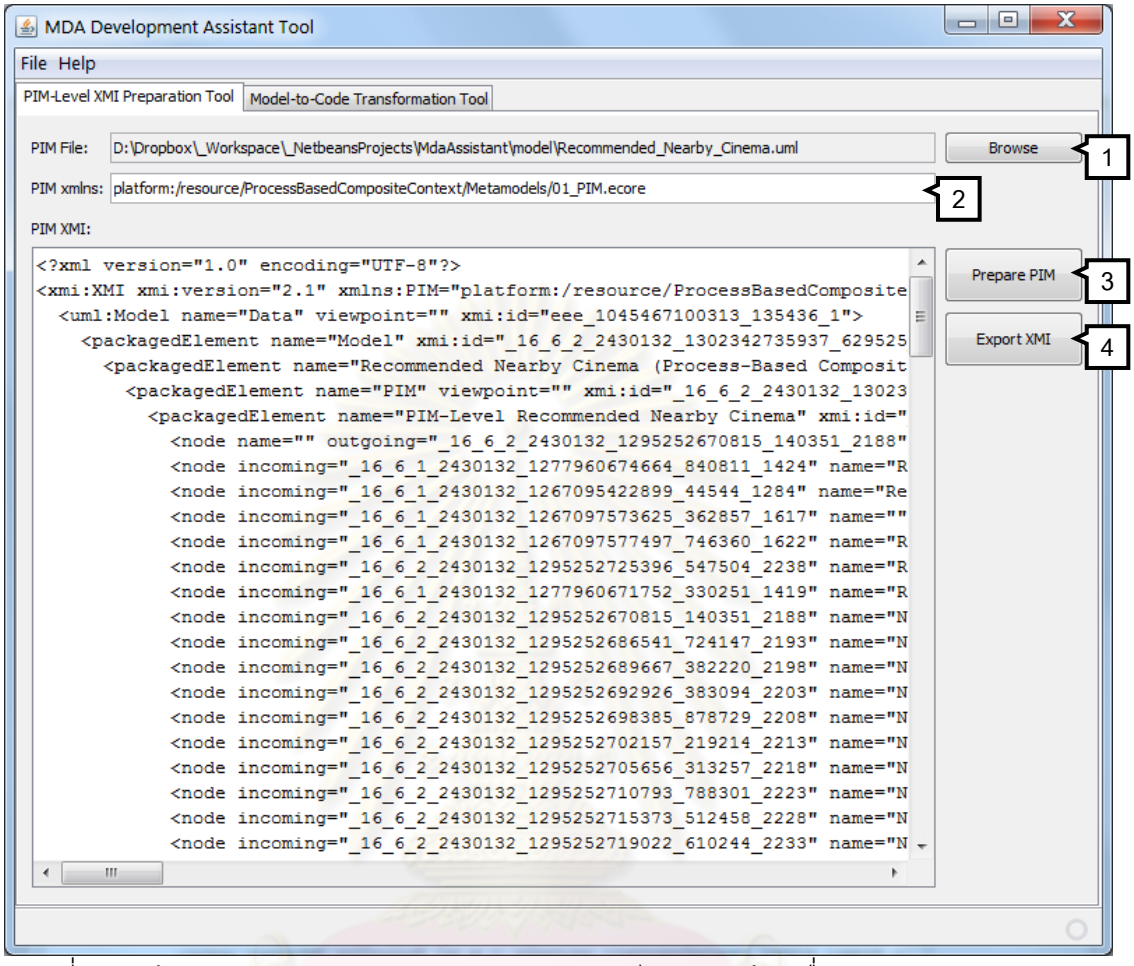


รูปที่ ก.1 ภาพรวมการพัฒนาแอปพลิเคชันแบบล่วงรู้บริบท เมื่อใช้แอปพลิเคชัน MDA Development Assistant Tool ช่วยในการพัฒนา

แอปพลิเคชันนี้แบ่งการทำงานออกเป็น 2 ส่วนคือ

1) ส่วนการจัดรูปแบบไฟล์ก่อนเข้าเครื่องประมวลผล ATL ทำหน้าที่จัดรูปแบบไฟล์ EMF UML2 v2.x XMI ที่ได้จากการเขียนแบบจำลองบริบทด้วยโปรแกรม MagicDraw UML ให้อยู่ในรูปแบบของไฟล์ XMI ที่สามารถนำไปเข้าเครื่องประมวลผลภาษา ATL ได้ทันที โดยลบข้อมูลส่วนที่ไม่สำคัญที่ไม่ได้ระบุในแบบจำลองเมตาที่ใช้กับเครื่องประมวลผลภาษา ATL ออก และแก้ไขค่าเนมสเปซของยูเอ็มโปรไฟล์สำหรับแบบจำลองระดับพีไอเอ็มให้ตรงกับที่อยู่ของไฟล์แบบจำลองเมตาที่เครื่องประมวลภาษา ATL มองเห็น
2) ส่วนการจัดรูปแบบไฟล์แบบจำลองระดับโค้ดที่ได้จากเครื่องประมวลผล ATL ให้เป็นโค้ด ทำหน้าที่จัดรูปแบบไฟล์ที่ได้จากเครื่องประมวลผลภาษา ATL ซึ่งเป็นแบบจำลองระดับโค้ด ซึ่งอยู่ในรูปแบบของไฟล์ XMI ให้เป็นโค้ดตามแพลตฟอร์มเป้าหมาย ซึ่งในที่นี้รองรับการแปลงสำหรับ 2 แพลตฟอร์ม ได้แก่ โทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ และโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์

การใช้งานแอปพลิเคชันส่วนการจัดรูปแบบไฟล์ก่อนเข้าเครื่องประมวลผล

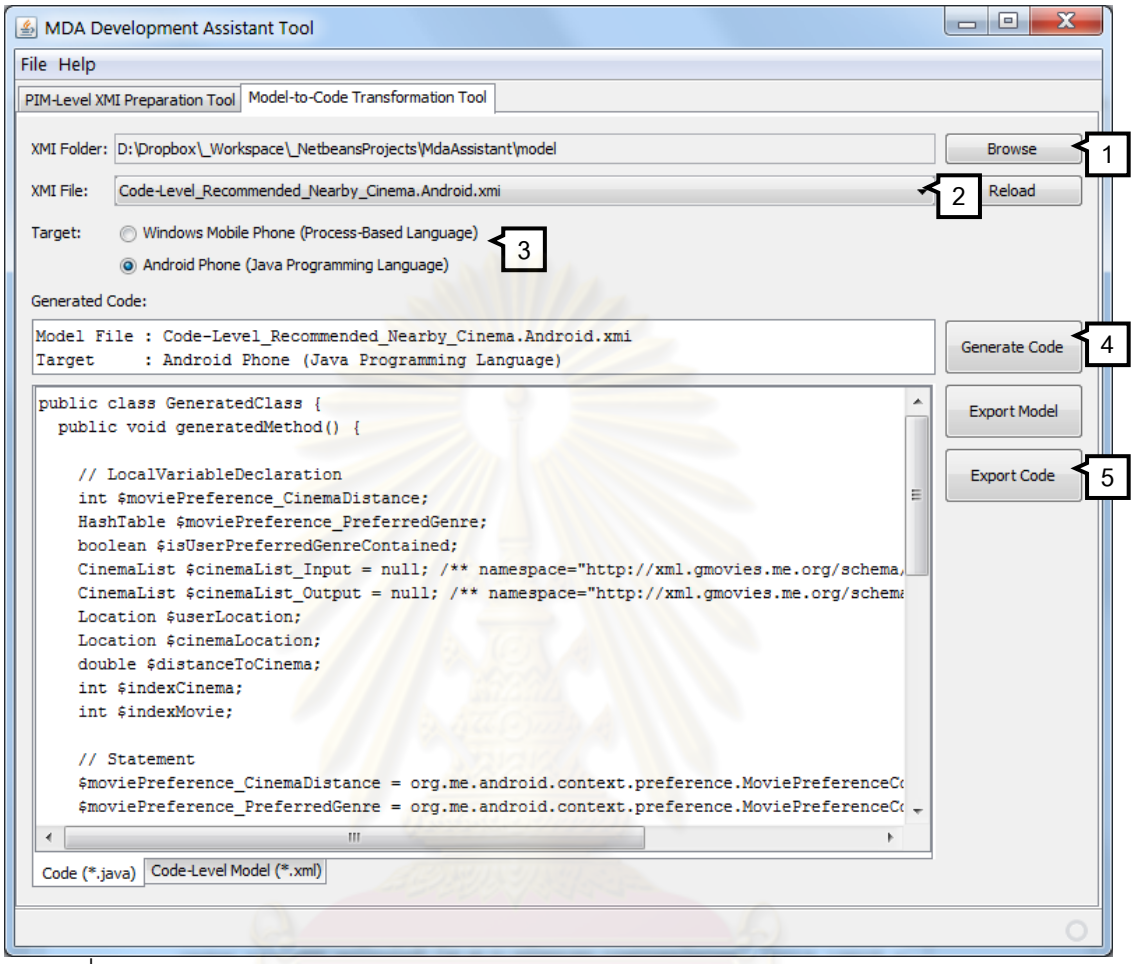


รูปที่ ก.2 หน้าต่างแอปพลิเคชันส่วนการจัดการรูปแบบไฟล์ก่อนเข้าเครื่องประมวลผลภาษา ATL

แอปพลิเคชันส่วนการจัดรูปแบบไฟล์ก่อนเข้าเครื่องประมวลผลภาษา ATL มีหน้าตาดังรูปที่ ก.2 ขั้นตอนการใช้งานแอปพลิเคชันส่วนนี้มีดังนี้

1) กดปุ่ม “Browse” เพื่อเลือกไฟล์ที่มีข้อมูลแบบจำลอง ที่ได้จากโปรแกรม MagicDraw UML โดยปกติจะเป็นไฟล์ *.uml ที่มีชื่อไฟล์ตรงกับชื่อโปรเจคในโปรแกรม
2) ระบุค่าเนมสเปซของยูเอ็มโปรไฟล์สำหรับแบบจำลองระดับพีไอเอ็มให้ตรงกับที่อยู่ของไฟล์แบบจำลองเมตาที่เครื่องประมวลภาษา ATL มองเห็น
3) กดปุ่ม “Prepare PIM” เพื่อทำการจัดรูปแบบไฟล์ โดยที่แอปพลิเคชันจะแสดงผลลัพธ์ในกล่องข้อความ “PIM XMI” ที่อยู่ทางด้านซ้ายของปุ่ม โดยที่สามารถแก้ไขเพิ่มเติมได้ในกล่องข้อความนี้
4) เมื่อได้ไฟล์ที่ต้องการแล้วให้กดปุ่ม “Export XMI” เพื่อทำการบันทึกไฟล์ XMI เพื่อนำไปเข้าเครื่องประมวลผลภาษา ATL ต่อไป

การใช้งานแอปพลิเคชันส่วนการจัดรูปแบบไฟล์แบบจำลองระดับโค้ดให้เป็นโค้ด



รูปที่ ก.3 หน้าต่างแอปพลิเคชันส่วนการจัดการรูปแบบไฟล์แบบจำลองระดับโค้ดให้เป็นโค้ด

แอปพลิเคชันส่วนการจัดรูปแบบไฟล์แบบจำลองระดับโค้ดที่ได้จากเครื่องประมวลผล ATL ให้เป็นโค้ด มีหน้าตาดังรูปที่ ก.3 ขั้นตอนการใช้งานแอปพลิเคชันส่วนนี้มีดังนี้

1) กดปุ่ม “Browse” เพื่อเลือกโฟลเดอร์ที่จัดเก็บไฟล์แบบจำลองระดับโค้ดที่ได้จากการประมวลผลด้วยเครื่องประมวลผลภาษา ATL
2) กดเลือกไฟล์แบบจำลองที่ต้องการ โดยถ้าต้องการให้แอปพลิเคชันค้นหาไฟล์ในโฟลเดอร์นี้ใหม่อีกครั้งให้กดปุ่ม “Reload”
3) กดเลือกแพลตฟอร์มเป้าหมายให้ตรงกับแบบจำลองระดับโค้ดที่ได้เลือกไว้
4) กดปุ่ม “Generate Code” เพื่อทำการจัดรูปแบบไฟล์ โดยที่แอปพลิเคชันจะแสดงผลลัพธ์ในกล่องข้อความ “Generated Code” ที่อยู่ทางด้านซ้ายมือของปุ่ม โดยที่สามารถแก้ไขเพิ่มเติมได้ในกล่องข้อความนี้

5) เมื่อได้ไฟล์ที่ต้องการแล้วให้กดปุ่ม “Export Code” เพื่อทำการบันทึกไฟล์โค้ด เพื่อนำไปใช้สร้างเอพีไอสำหรับแพลตฟอร์มที่ต้องการต่อไป

แถบข้อความที่อยู่ใต้คำว่า “Generated Code” ใช้แสดงชื่อของไฟล์ และแพลตฟอร์มเป้าหมายที่ผู้ใช้งานได้เลือกไว้

ในกรณีที่ต้องการดูว่าไฟล์ XMI ที่ได้จากการประมวลผลด้วยเครื่องประมวลผลภาษา ATL สามารถจัดรูปแบบเป็นแบบจำลองระดับโค้ดที่อยู่ในรูปแบบของไฟล์ XML ที่ตรงตามสกีมาที่ได้ออกแบบไว้ในงานวิจัยนี้ สามารถดูได้โดยคลิกที่แท็บ “Code-Level Model (*.xml)” ที่อยู่ด้านล่างซ้ายของหน้าจอแอปพลิเคชัน และสามารถบันทึกแบบจำลองระดับโค้ดดังกล่าวได้โดยการกดปุ่ม “Export Model”

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

คู่มือการใช้งานแอปพลิเคชัน G! Movies for Windows Mobile

แอปพลิเคชัน G! Movies for Windows Mobile เป็นแอปพลิเคชันต้นแบบสำหรับโทรศัพท์เคลื่อนที่ประเภทวินโดวส์โมบายล์ มีความสามารถในการหาข้อมูลโรงภาพยนตร์ที่อยู่ที่ในบริเวณที่ต้องการ โดยสามารถแสดงรายการโรงภาพยนตร์ได้ทั้งแบบมุมมองแบบแผนที่ และมุมมองแบบรายการ รวมทั้งสามารถรายละเอียดเกี่ยวกับโรงภาพยนตร์ที่สนใจ ทั้งสถานที่ตั้ง ระยะห่างจากตำแหน่งปัจจุบัน และรายการภาพยนตร์ที่ฉายอยู่ ณ โรงภาพยนตร์นั้น ๆ ได้ โดยแอปพลิเคชันต้นแบบนี้มีการเรียกใช้งานเอพีไอของบริบทประกอบเชิงกระบวนการ “Recommended Nearby Cinema” ที่ได้พัฒนาขึ้นในงานวิจัยนี้

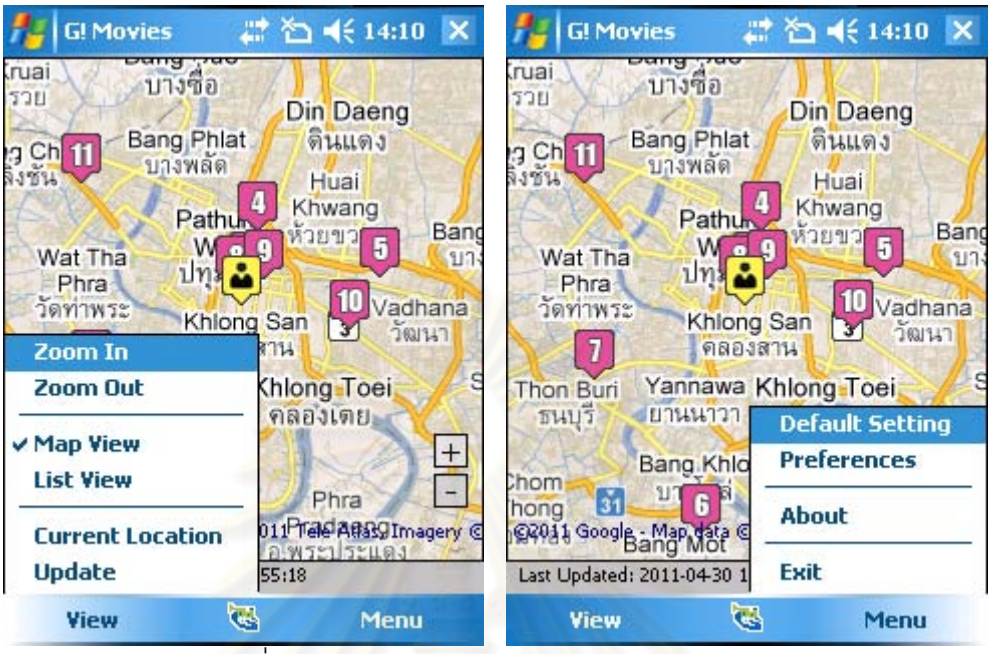
การติดตั้งแอปพลิเคชันสามารถทำได้สองวิธี คือ

1) ติดตั้งจากไฟล์ cab โดยจำเป็นต้องมีการติดตั้ง .NET Compact Framework 3.5 และ Microsoft SQL Server Compact 3.5 ก่อน
2) ติดตั้งโดยตรงจากเครื่องมือพัฒนา (Microsoft Visual Studio 2008) วิธีนี้ไม่จำเป็นต้องติดตั้งโปรแกรมเหล่านั้นเพิ่มเติม เนื่องจากเครื่องมือพัฒนาได้จัดการติดตั้งให้เรียบร้อยแล้ว

รูปที่ ข.1 หน้าต่างหลักของแอปพลิเคชัน G! Movies for Windows Mobile

เมื่อเปิดใช้งานแอปพลิเคชันจะเจอหน้าต่างหลักดังรูปที่ ข.1 โดยแสดงแผนที่พร้อมทั้งข้อมูลรายการโรงภาพยนตร์ที่เคยเรียกใช้งาน และวันที่ที่อัพเดทข้อมูลครั้งล่าสุด

แอปพลิเคชันมีเมนูหลักให้เลือก 2 เมนู คือ View และ Menu ดังรูปที่ ข.2



รูปที่ ข.2 เมนูหลัก และเมนูย่อยของแอปพลิเคชัน

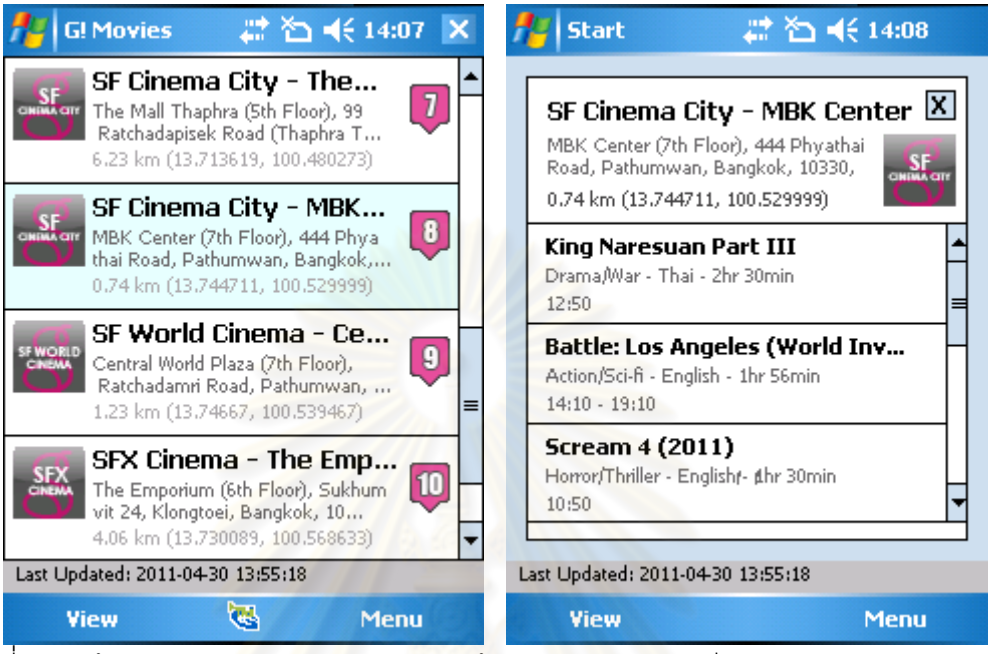
เมนู View ประกอบด้วยเมนูย่อย ๆ ดังนี้

- Zoom In / Zoom Out เมนูนี้ใช้งานได้เฉพาะในมุมมองแบบแผนที่เท่านั้น ใช้สำหรับซูมแผนที่เข้าหรือออก อีกหนึ่งวิธีที่สามารถใช้ในการซูมแผนที่เข้าหรือออกได้คือการกดปุ่ม + หรือ - ที่อยู่ทางมุมขวาล่างของแผนที่
- Map View ใช้เลือกเพื่อแสดงมุมมองแบบแผนที่
- List View ใช้เลือกเพื่อแสดงมุมมองแบบรายการ
- Current Location ใช้เพื่อทำการค้นหา และระบุตำแหน่งของผู้ใช้ใหม่อีกครั้ง
- Update ใช้เพื่ออัพเดทข้อมูลใหม่ โดยการเรียกใช้งานเอพีไอของบริบทประกอบเชิงกระบวนการ “Recommended Nearby Cinema” ที่ได้พัฒนาขึ้นในงานวิจัยนี้

เมนู Menu ประกอบด้วยเมนูย่อย ๆ ดังนี้

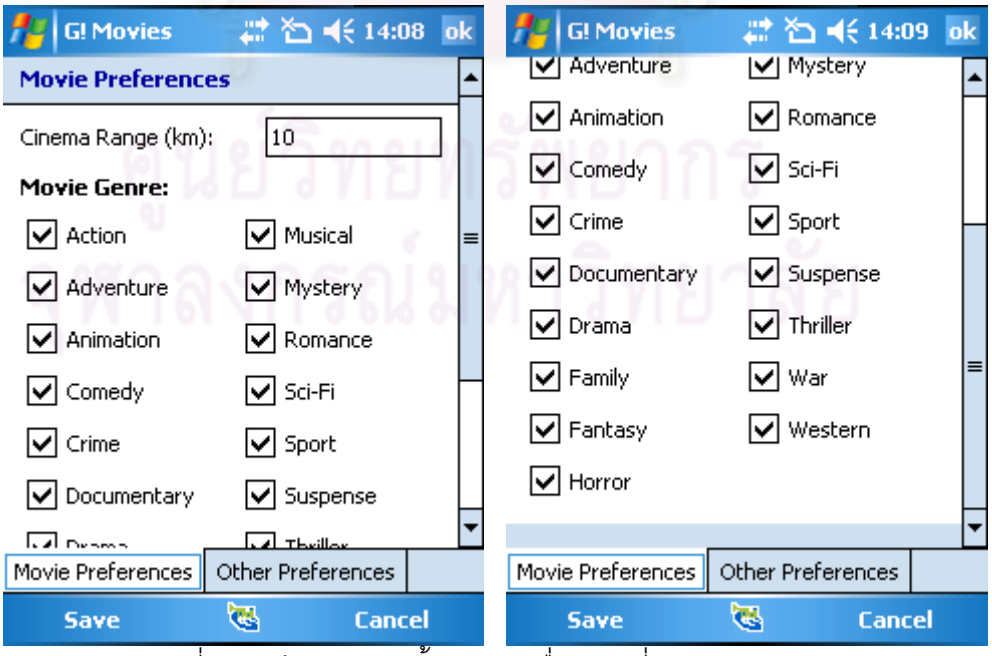
- Default Setting ใช้สำหรับรีเซตการตั้งค่าต่าง ๆ ให้อยู่ในสถานะตั้งต้น
- Preferences ใช้สำหรับปรับตั้งค่าต่าง ๆ สำหรับแอปพลิเคชัน
- About ใช้แสดงรายละเอียดเกี่ยวกับแอปพลิเคชัน เวอร์ชันของแอปพลิเคชัน และรายชื่อผู้พัฒนา
- Exit ใช้สำหรับปิดแอปพลิเคชัน

ในหน้าต่างมุมมองแบบรายการ เมื่อกดที่แต่ละรายการโรงภาพยนตร์จะมีหน้าต่างแสดงรายละเอียดของโรงภาพยนตร์นั้น ๆ พร้อมทั้งรายการภาพยนตร์ที่ฉายอยู่แสดงขึ้นมา ดังรูปที่ ข.3



รูปที่ ข.3 หน้าต่างมุมมองแบบรายการ และหน้าต่างแสดงรายละเอียดของแต่ละโรงภาพยนตร์

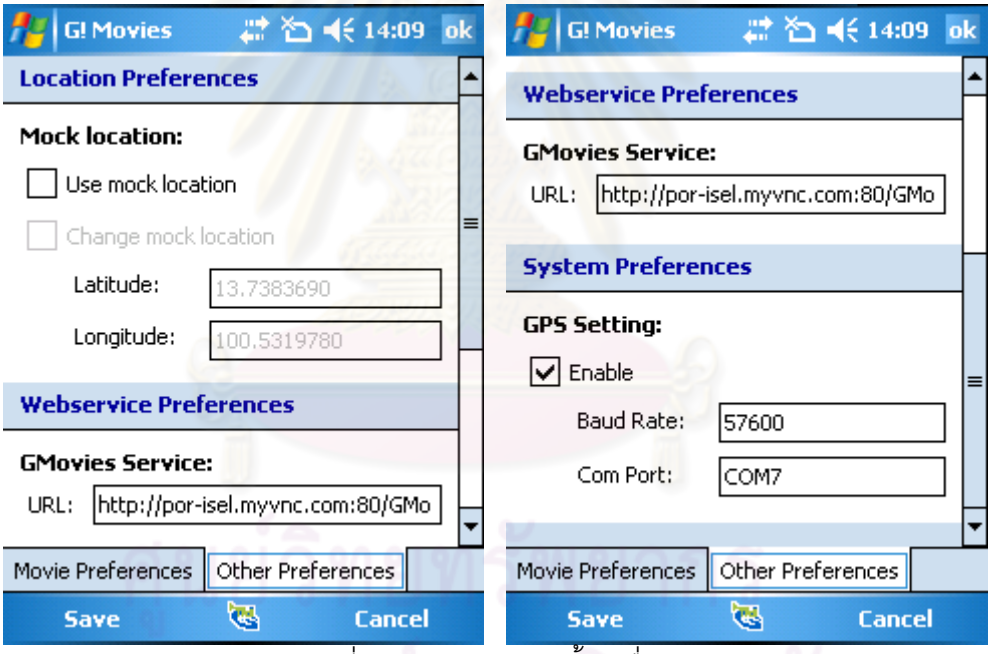
เมื่อต้องการปรับตั้งค่าต่าง ๆ ให้กดที่เมนูย่อย “Preferences” จากหน้าต่างหลัก จะพบหน้าต่างสำหรับปรับตั้งค่าดังรูปที่ ข.4 โดยใช้สำหรับปรับตั้งค่าความชื่นชอบเกี่ยวกับภาพยนตร์ ทั้งระยะห่างของโรงภาพยนตร์จากตำแหน่งปัจจุบัน และชนิดของภาพยนตร์ที่สนใจ



รูปที่ ข.4 หน้าต่างปรับตั้งค่าความชื่นชอบเกี่ยวกับภาพยนตร์

นอกจากหน้าต่างปรับตั้งค่าความชื่นชอบเกี่ยวกับภาพยนตร์แล้ว ยังมีหน้าต่างสำหรับปรับตั้งค่าอื่น ๆ โดยการเข้าใช้งานหน้าต่างนี้ กดที่แท็บ “Other Preferences” ด้านล่างดังรูป ข.5 การปรับตั้งค่าอื่น ๆ ประกอบด้วย

1) Location Preferences ใช้ปรับเลือกได้ว่าจะใช้การระบุตำแหน่งของผู้ใช้แบบไหน ทั้งจากการตรวจสอบตำแหน่งจริง หรือจากการระบุตำแหน่งที่ต้องการเอง
   - หากต้องการระบุตำแหน่งเองให้กดเลือกที่ “Use mock location” จากนั้นกดเลือกที่ “Change mock location” และระบุพิกัดที่ต้องการลงในช่องละติจูด ลองจิจูด โดยที่หากไม่กดเลือกที่ “Change mock location” แอปพลิเคชันจะทำการระบุพิกัดจากค่าตั้งต้นของแอปพลิเคชันดังแสดงในรูปที่ ข.5
2) Webservice Preferences ใช้ระบุตำแหน่งที่อยู่ของเว็บเซอร์วิซ
3) System Preferences ใช้ตั้งค่าการเชื่อมต่อกับอุปกรณ์ GPS แบบ Bluetooth



รูปที่ ข.5 หน้าต่างปรับตั้งค่าอื่น ๆ

ภาคผนวก ค

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

คู่มือการใช้งานแอปพลิเคชัน G! Movies for Android

แอปพลิเคชัน G! Movies for Windows Android เป็นแอปพลิเคชันต้นแบบสำหรับโทรศัพท์เคลื่อนที่ประเภทแอนดรอยด์ มีความสามารถในการหาข้อมูลโรงภาพยนตร์ที่อยู่ที่ในบริเวณที่ต้องการ โดยสามารถแสดงรายการโรงภาพยนตร์ได้ทั้งแบบมุมมองแบบแผนที่ และมุมมองแบบรายการ รวมทั้งสามารถรายละเอียดเกี่ยวกับโรงภาพยนตร์ที่สนใจ ทั้งสถานที่ตั้ง ระยะห่างจากตำแหน่งปัจจุบัน และรายการภาพยนตร์ที่ฉายอยู่ ณ โรงภาพยนตร์นั้น ๆ ได้ โดยแอปพลิเคชันต้นแบบนี้มีการเรียกใช้งานเอพีไอของบริบทประกอบเชิงกระบวนการ “Recommended Nearby Cinema” ที่ได้พัฒนาขึ้นในงานวิจัยนี้

การติดตั้งแอปพลิเคชันสามารถทำได้สองวิธี คือ

1) ติดตั้งจากไฟล์ apk
2) ติดตั้งโดยตรงจากเครื่องมือพัฒนา (Eclipse)

รูปที่ ค.1 หน้าต่างหลักของแอปพลิเคชัน G! Movies for Android

เมื่อเปิดใช้งานแอปพลิเคชันจะเจอหน้าต่างหลักดังรูปที่ ค.1 โดยแสดงแผนที่พร้อมทั้งข้อมูลรายการโรงภาพยนตร์ที่เคยเรียกใช้งาน และวันที่ที่อัพเดทข้อมูลครั้งล่าสุด

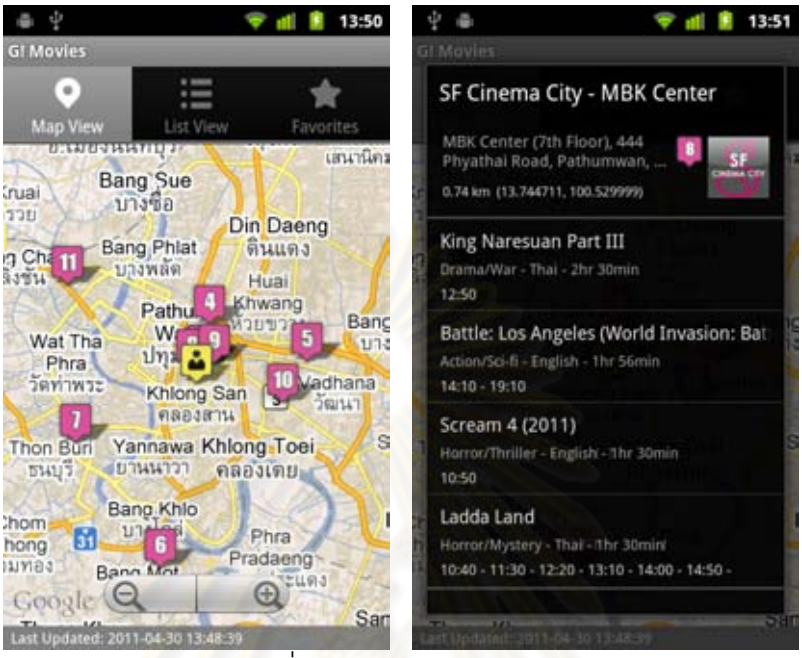
แอปพลิเคชันมีเมนูหลักทั้งหมด 6 เมนู ดังรูปที่ ค.2 โดยการแสดงรายการเมนูให้กดที่ปุ่มเมนูของเครื่องโทรศัพท์ แต่ละเมนูมีหน้าที่ดังนี้

- Update ใช้เพื่ออัพเดทข้อมูลใหม่ โดยการเรียกใช้งานเอพีไอของบริบทประกอบเชิงกระบวนการ “Recommended Nearby Cinema” ที่ได้พัฒนาขึ้นในงานวิจัยนี้
- Current Location ใช้เพื่อทำการค้นหา และระบุตำแหน่งของผู้ใช้ใหม่อีกครั้ง
- Preferences ใช้สำหรับปรับตั้งค่าต่าง ๆ สำหรับแอปพลิเคชัน
- Default Setting ใช้สำหรับรีเซตการตั้งค่าต่าง ๆ ให้อยู่ในสถานะตั้งต้น
- Search ใช้สำหรับค้นหาโรงภาพยนตร์ (ยังไม่สามารถใช้งานได้ในเวอร์ชันนี้)
- About ใช้แสดงรายละเอียดเกี่ยวกับแอปพลิเคชัน เวอร์ชันของแอปพลิเคชัน และรายชื่อผู้พัฒนา

รูปที่ ค.2 เมนูหลักของแอปพลิเคชัน

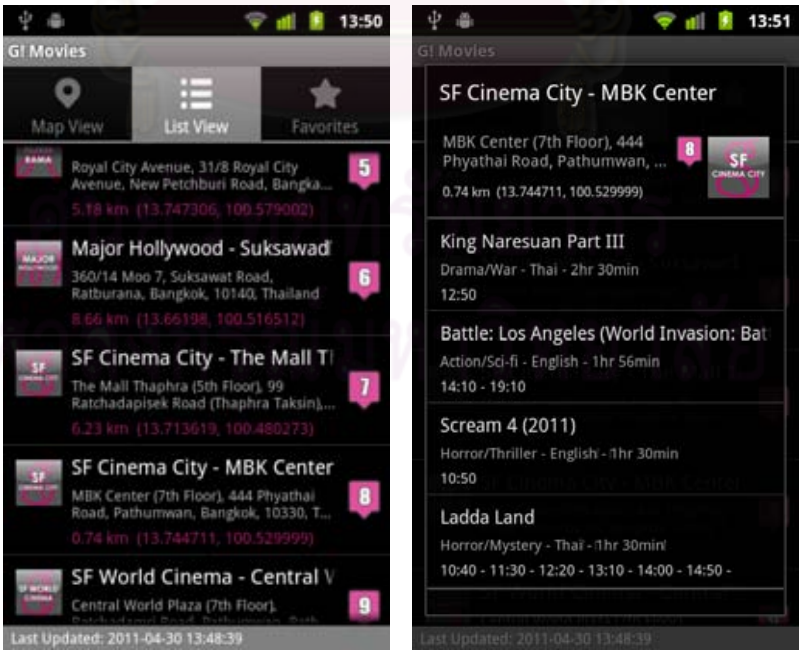
การเลือกมุมมองการแสดงผลสามารถทำได้โดยการกดที่แท็บด้านบน ถ้าต้องการมุมมองแบบแผนที่ให้กดเลือกที่ “Map View” ถ้าต้องการมุมมองแบบรายการให้กดเลือกที่ “List View” โดยที่ในหน้าต่างมุมมองแบบแผนที่ เมื่อกดที่แต่ละไอคอนโรงภาพยนตร์จะมีหน้าต่างแสดงรายละเอียดของโรงภาพยนตร์นั้น ๆ พร้อมทั้งรายการภาพยนตร์ที่ฉายอยู่แสดงขึ้นมา ดังรูปที่ ค.3 การซูมแผนที่ทำได้ 2 วิธี คือ 1) กดที่ปุ่ม + หรือ – ที่ด้านล่างของแผนที่ โดยปุ่มนี้จะแสดงเมื่อมีการ

สัมผัสบนหน้าจอ หรือ 2) ใช้นิ้วสองนิ้วแตะบนหน้าจอ โดยเลื่อนนิ้วออกจากกันเมื่อต้องการซูมเข้า และเลื่อนนิ้วเข้าหากันเมื่อต้องการซูมออก



รูปที่ ค.3 หน้าต่างมุมมองแบบแผนที่ และหน้าต่างแสดงรายละเอียดของแต่ละโรงภาพยนตร์

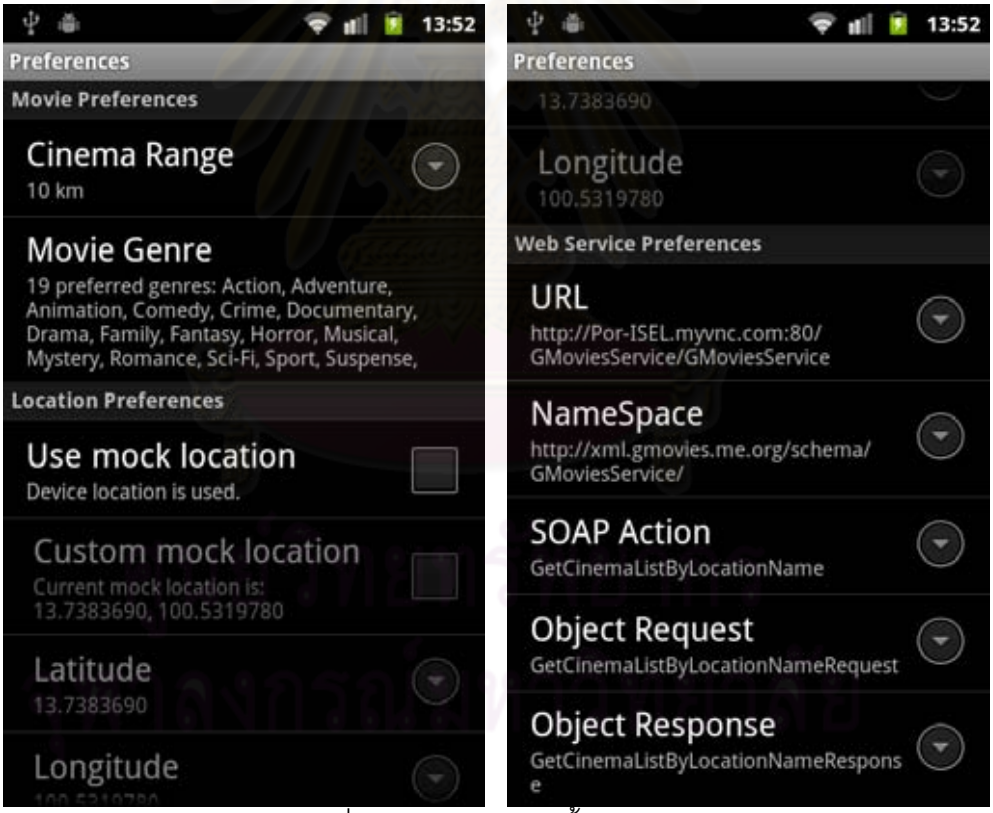
ในหน้าต่างมุมมองแบบรายการ เมื่อกดที่แต่ละรายการโรงภาพยนตร์จะมีหน้าต่างแสดงรายละเอียดของโรงภาพยนตร์นั้น ๆ พร้อมทั้งรายการภาพยนตร์ที่ฉายอยู่แสดงขึ้นมา ดังรูปที่ ค.4



รูปที่ ค.4 หน้าต่างมุมมองแบบรายการ และหน้าต่างแสดงรายละเอียดของแต่ละโรงภาพยนตร์

เมื่อต้องการปรับตั้งค่าต่าง ๆ ให้กดที่เมนู “Preferences” จากหน้าต่างหลัก จะพบหน้าต่างสำหรับปรับตั้งค่าดังรูปที่ ค.5 โดยใช้สำหรับปรับตั้งค่าต่าง ๆ ดังนี้

1) Movie Preferences ใช้สำหรับปรับตั้งค่าความชื่นชอบเกี่ยวกับภาพยนตร์ ทั้งระยะห่างของโรงภาพยนตร์จากตำแหน่งปัจจุบัน และชนิดของภาพยนตร์ที่สนใจ
2) Location Preferences ใช้ปรับเลือกได้ว่าจะใช้การระบุตำแหน่งของผู้ใช้แบบไหน ทั้งจากการตรวจสอบตำแหน่งจริง หรือจากการระบุตำแหน่งที่ต้องการเอง
   - หากต้องการระบุตำแหน่งเองให้กดเลือกที่ “Use mock location” จากนั้นกดเลือกที่ “Custom mock location” และระบุพิกัดที่ต้องการลงในช่องละติจูด ลองจิจูด โดยที่หากไม่กดเลือกที่ “Custom mock location” แอปพลิเคชันจะทำการระบุพิกัดจากค่าตั้งต้นของแอปพลิเคชันดังแสดงในรูปที่ ค.5
3) Webservice Preferences ใช้ข้อมูลเกี่ยวกับเว็บเซอร์วิซ



รูปที่ ค.5 หน้าต่างปรับตั้งค่าต่าง ๆ

ประวัติผู้เขียนวิทยานิพนธ์

นายเอกวิชญ์ กุลธวัชวิชัย เกิดเมื่อวันที่ 3 ตุลาคม พ.ศ. 2528 ที่จังหวัดแพร่ สำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ เกียรตินิยมอันดับสอง จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2550 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ณ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551 งานวิจัยที่สนใจ ได้แก่ การคำนวณแบบล่วงรู้บริบท และเทคโนโลยีเกี่ยวกับโทรศัพท์เคลื่อนที่ประเภทสมาร์ทโฟน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย