การแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

น.ส.อมรัตน์ พิมโคตร

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Transforming Timing Diagram into Timed Automata For Preemptive Scheduling

Miss Amarat Pimkote

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Software Engineering
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ
โดย	น.ส.อมรัตน์ พิมโคตร
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

..... กรรมการ
(ดร.ดวงดาว วิชาดากุล)

..... กรรมการภายนอกมหาวิทยาลัย
(ดร.เดชานุชิต กตัญญูทวีทิพย์)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

อมรัตน์ พิมโคตร : การแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ. (Transforming Timing Diagram into Timed Automata For Preemptive Scheduling) อ.ที่ปรึกษาหลัก : รศ. ดร.วิวัฒน์ วัฒนาวุฒิ

งานที่เกิดขึ้นพร้อมกันในระบบเรียลไทม์อาจต้องการทรัพยากรที่ใช้ร่วมกันอย่างจำกัด เช่นการใช้งานร่วมกันของ CPU เพียงตัวเดียวที่มีงานเป็นจำนวนมาก เมื่อใช้การจัดกระบวนการเชิงพรีเอ็มทีฟ ซึ่งงานที่กำลังทำงานอยู่ที่มีค่าลำดับความสำคัญต่ำกว่ามักจะถูกจัดลำดับให้อยู่ในสถานะพักการทำงานหรือสถานะโดเมน โดยงานใหม่ที่มีค่าลำดับความสำคัญสูงกว่าจะเข้ามาแทนที่ สุดท้ายจึงกลายเป็นว่างานใหม่เข้ามาทำงานแทนลำดับงานที่โดนแทรกหรือถูกพรีเอ็มทีฟไว้ก่อนหน้านี้ที่มีค่าลำดับความสำคัญต่ำกว่า งานที่มีค่าลำดับความสำคัญต่ำกว่าดังกล่าวจะเริ่มต้นทำงานอีกครั้งเพื่อดำเนินการต่อในสถานะที่ทำงานทันทีหลังจากงานที่มีค่าลำดับความสำคัญสูงกว่าได้ทำงานเสร็จสิ้น แผนภาพเวลาเป็นแผนภาพที่มีลักษณะงานเป็นอิสระต่อกัน และงานจะถูกเริ่มต้นพร้อมกัน ผลกระทบของการจัดกระบวนการเชิงพรีเอ็มทีฟเอาไว้จะมีความสัมพันธ์กันและทำให้การดำเนินของเส้นเวลาหรือไทม์ไลน์ของงานที่เกิดขึ้นพร้อมกันเหล่านี้ถูกเปลี่ยนแปลงไป

กฎการจับคู่สำหรับการแปลงแผนภาพเวลาที่เป็นอิสระต่อกันเป็นไทมด์ออโตมาตาที่ได้รับการออกแบบในวิทยานิพนธ์นี้ และยังมีเครื่องมือซอฟต์แวร์ที่ได้รับการพัฒนาเพื่อแปลงไฟล์ต้นทางนำเข้าสกุลไฟล์ XML ของแผนภาพเวลาเป็นไฟล์ไทมด์ออโตมาตา สามารถจำลองแผนภาพไทมด์ออโตมาตาด้วยเครื่องมือ UPPAAL ซึ่งผลลัพธ์ของไทมด์ออโตมาตาจะแสดงกรอบเวลาโดยรวมของงานที่เกิดขึ้นพร้อมกันอันเป็นผลกระทบของการจัดกำหนดการเชิงฟรีเอ็มทีฟ การจำลองไทมด์ออโตมาตาจะจัดเตรียมตัวแปรนาฬิกาและสถานะโดเมนพิเศษเพิ่มเติม จากนั้นจึงนำไทมด์ออโตมาตาที่แปลงมาทวนสอบคุณสมบัติ TCTL ว่าการทำงานนั้นถูกต้อง เครื่องมือซอฟต์แวร์ของเราจะดำเนินการแปลงไดอะแกรมสำหรับการจัดกระบวนการเชิงพรีเอ็มทีฟ และใช้กรณีศึกษาสามกรณีเพื่อแสดงกระบวนการแปลงและการจำลองขั้นตอนกระบวนการทำงาน

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

6170327521 : MAJOR SOFTWARE ENGINEERING

KEYWORD: Simulation, Preemptive Scheduling, Timing Diagram, Timed Automata, Dormant, UPPAAL, TCTL

Amarat Pimkote : Transforming Timing Diagram into Timed Automata For Preemptive Scheduling. Advisor: Assoc. Prof. WIWAT VATANAWOOD, Ph.D.

In the real-time system, the concurrent tasks occasionally require the limit shared resource, such as a single CPU would be shared among multiple tasks. When the preemptive scheduling is typically selected, the running task with lower priority value would be preempted into the dormant state by the new coming task with the higher priority value which would finally become the new running task instead. Then the previous preempted task with lower priority value would finally be activated back to resume the running state immediately after the end of the task with higher priority value. Given several timing diagrams of the independent tasks which would be concurrently initiated, the effects of preemptive scheduling would be concerned and cause the changing of the overall timelines of these concurrent tasks.

In this thesis, the mapping rules of transforming the independent timing diagrams into timed automata are designed and a software tool is developed to transform the input XML files of the timing diagrams into the timed automata file, to be simulated with the UPPAAL timed automata simulation tool. The resulting timed automata shows the overall timelines of the input concurrent tasks under the effects of the preemptive scheduling. The clock variables and the extra dormant states are included to enable the observations of the timed automata

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study: Software Engineering Student's Signature .....

Academic Year: 2021 Advisor's Signature .....

กิตติกรรมประกาศ

ในการทำวิทยานิพนธ์เรื่องการแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟสามารถทำสำเร็จลุล่วงไปด้วยดี ดิฉันขอขอบพระคุณ รศ. ดร. วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์รวมถึงคณะกรรมการสอบวิทยานิพนธ์ ได้แก่ ดร. ดวงดาว วิชาดากุล, ดร. เดชานุชิต กตัญญูทวีทิพย์, รศ. ดร. พรศิริ หมื่นไชยศรี และอาจารย์สาขาวิศวกรรมซอฟต์แวร์ทุกท่าน ที่คอยให้คำปรึกษาและชี้แนะแนวทางในการแก้ปัญหาต่าง ๆ รวมถึงตรวจสอบความถูกต้อง ให้ความรู้ต่าง ๆ และแก้ไขเกี่ยวกับวิทยานิพนธ์ฉบับนี้เป็นอย่างดี ขอขอบพระคุณ คุณพ่อ คุณแม่ และญาติพี่น้องทุก ๆ ท่านที่เป็นกำลังใจสำคัญอย่างยิ่งที่ทำให้วิทยานิพนธ์นี้สำเร็จไปด้วยดี และขอบคุณเพื่อน ๆ ทุกคนทั้งในสาขาวิชาวิศวกรรมซอฟต์แวร์รวมไปถึงเพื่อน ๆ ของดิฉันที่มีส่วนช่วยเหลือในด้านต่าง ๆ เกี่ยวกับวิทยานิพนธ์ อีกทั้งสอบถามความคืบหน้าของวิทยานิพนธ์อยู่เสมอ

อมรัตน์ พิมโคตร

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

..... ค

บทคัดย่อภาษาไทย ..... ค

..... ง

บทคัดย่อภาษาอังกฤษ ..... ง

กิตติกรรมประกาศ ..... จ

สารบัญ ..... ฉ

สารบัญตาราง ..... ญ

สารบัญรูปภาพ ..... ฏ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญ 1

1.2 วัตถุประสงค์ของโครงงาน 3

1.3 ขอบเขตของโครงงาน 3

1.4 ขั้นตอนการดำเนินงานของวิทยานิพนธ์ 4

1.5 ประโยชน์ที่คาดว่าจะได้รับ 4

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 4

บทที่ 2 ทฤษฎี งานวิจัย และเครื่องมือที่เกี่ยวข้อง 5

2.1 ทฤษฎีที่เกี่ยวข้อง 5

2.1.1 ระบบเวลาจริง 5

2.1.2 ประเภทอัลกอริทึมสำหรับการทำกำหนดการเวลา [8], [9] 6

2.1.3 นโยบายการทำกำหนดการเวลา 11

2.1.4 แผนภาพเวลา 13

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.5 ออโตมาตา 16
2.1.5.1 ไทมด์ออโตมาตา 16
2.2 งานวิจัยที่เกี่ยวข้อง 20
2.2.1 งานวิจัย “Unifying Modeling and Simulation Based on UML Timing Diagram and UPPAAL โดย Kangle Cui, Zongyuan Yang, Jinkui Xie, Kaiyu Wan ปี ค.ศ. 2010 [3] 20
2.2.3 งานวิจัย “Formalizing Real-Time Embedded System into Promela” โดย Punwess Sukvanich, Arthit Thongtak, Wiwat Vatanawood ปี ค.ศ. 2015 [4]22
2.2.4 งานวิจัย “Transforming WS-BPEL into Timed Automata” โดย Pramate Chandratarat และ Wiwat Vatanawood ปี ค.ศ. 2019 [5] 23
2.2.5 เครื่องมือ UPPAAL 25
บทที่ 3 วิธีการวิจัยของการสร้างแบบจำลองแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ 31
3.1 การออกแบบและสร้างกฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟ 32
3.2 อธิบายการแปลงแผนภาพเวลาและไทมด์ออโตมาตาในรูปแบบของโครงสร้างเอกซ์เอ็มแอล 38
3.2.1 การแปลงแผนภาพเวลาให้อยู่ในรูปแบบของไฟล์เอกซ์เอ็มแอล 38
3.2.2 การแปลงไทมด์ออโตมาตาในรูปแบบของไฟล์เอกซ์เอ็มแอล 41
บทที่ 4 การออกแบบและพัฒนาเครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ 52
4.1 ภาพรวมของการพัฒนาเครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ 52
4.1.1 การสร้างตัวแบบซอฟต์แวร์ (Software Modeling) 52
4.1.2 ภาพรวมการทำงานของเครื่องมือ 73
4.1.3 ข้อกำหนดความต้องการเชิงฟังก์ชันของเครื่องมือ (Functional Requirements) 74
4.1.4 ข้อจำกัดเครื่องมือ 74

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.1.5 โครงสร้างของส่วนต่อประสานกับผู้ใช้ระบบ 74
บทที่ 5 การทดสอบและประเมินผล 78
5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ 78
5.1.1 ฮาร์ดแวร์ (Hardware) 78
5.1.2 ซอฟต์แวร์ (Software) 78
5.2 กรณีการทดสอบเครื่องมือการแปลง 78
5.3 กรณีทดสอบ 79
5.4 การทดสอบเครื่องมือการแปลงโดยใช้กรณีศึกษา 80
5.4.1 กรณีทดสอบที่ 1 เมื่อมีรถฉุกเฉินมา (CASE01) 80
5.4.1.1 แผนภาพเวลากรณีทดสอบที่ 1 80
5.4.1.2 ตารางแสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ 82
5.4.2 กรณีทดสอบที่ 2 ณ โรงพยาบาลแห่งหนึ่งที่มีการจัดลำดับความสำคัญ (CASE02) 88
5.4.2.1 แผนภาพเวลากรณีทดสอบที่ 2 89
5.4.2.2 ตารางแสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ 89
5.4.3 กรณีทดสอบที่ 3 ลำดับความสำคัญของงานบำรุงรักษา (CASE03) 92
5.4.3.1 แผนภาพเวลากรณีทดสอบที่ 3 93
5.4.3.2 ตารางแสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ 94
5.5 กรณีทดสอบหน้าเว็บเครื่องมือการแปลงอัตโนมัติที่ผู้วิจัยสร้างขึ้น “Timing Diagram into Timed Automata Converter” 99
บทที่ 6 บทสรุปและข้อเสนอแนะวิทยานิพนธ์ 97
6.1 สรุปผลวิทยานิพนธ์ 97
บรรณานุกรม 98
ประวัติผู้เขียน 100

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 2-1 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ [10]..... 6

ตารางที่ 2-2 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟและผลลัพธ์การคำนวณจากสมการ (1), (2) และเวลาที่กำหนดเสร็จ [10] 8

ตารางที่ 2-3 แสดงลำดับของกระบวนการทำงานของซีพียูแบบไม่มีพรีเอ็มทีฟ [10] 9

ตารางที่ 2-4 แสดงลำดับของกระบวนการทำงานของซีพียูแบบไม่มีพรีเอ็มทีฟและผลลัพธ์จากการคำนวณจากสมการ (1), (2) และเวลาที่กำหนดเสร็จ [10]..... 10

ตารางที่ 2-5 สรุปการจัดกำหนดรายการจัดกระบวนการแบบพรีเอ็มทีฟและกระบวนการแบบไม่มีพรีเอ็มทีฟ 12

ตารางที่ 2-6 ตัวอย่างสัญลักษณ์ที่มักพบในแผนภาพเวลา [7], [12]..... 14

ตารางที่ 2-7 แผนภาพสัญลักษณ์ไทมด์ออโตมาตา [13] 17

ตารางที่ 2-8 กฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตา 21

ตารางที่ 2-9 แสดงสัญลักษณ์ไทมด์ออโตมาตาในเครื่องมือ UPPAAL [15]..... 26

ตารางที่ 2-10 การอธิบายคุณสมบัติการทวนสอบของ TCTL [12] 28

ตารางที่ 3-1 แท็กเอกซ์เอ็มแอลของแผนภาพเวลาไฟล์สกุลต้นทาง 39

ตารางที่ 3-2 แท็กเอกซ์เอ็มแอลไทมด์ออโตมาตาไฟล์สกุลปลายทาง 42

ตารางที่ 4-1แสดงรายละเอียดของยูสเคสไอดีและยูสเคสทั้งหมดของเครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” 53

ตารางที่ 4-2 คำอธิบายยูสเคสของการนำเข้าแผนภาพเวลาในรูปแบบไฟล์เอ็กซ์เอ็มแอล..... 54

ตารางที่ 4-3 คำอธิบายยูสเคสของการแปลงไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลาไปเป็นไฟล์เอ็กซ์เอ็มแอลของไทมด์ออโตมาตา 55

ตารางที่ 4-4 คำอธิบายยูสเคสของการอ่านไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลา 56

ตารางที่ 4-5 คำอธิบายยูสเคสของการสร้างข้อมูลแบบจำลองไปเป็นโครงสร้างไฟล์เอ็กซ์เอ็มแอลของเครื่องมือ UPPAAL 57

ตารางที่ 4-6 คำอธิบายยูสเคสของการบันทึกไฟล์เอ็กซ์เอ็มแอลของไทมด์ออโตมาตา..... 58

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4-7 คำอธิบายยูสเคสของการเปิดเครื่องมือ UPPAAL 59
ตารางที่ 4-8 คำอธิบายยูสเคสของการนำเข้าไฟล์เอ็กซ์เอ็มแอลของไทมด์ออโตมาตาไปยังเครื่องมือ UPPAAL 60
ตารางที่ 4-9 คำอธิบายยูสเคสของการทวนสอบการออกแบบไทมด์ออโตมาตาในเครื่องมือ UPPAAL 61
ตารางที่ 5-1 รายการกรณีทดสอบและการใช้งานกฎของกรณีทดสอบ CASE01-CASE03 79
ตารางที่ 5-2 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ [13] 82
ตารางที่ 5-3 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ 90
ตารางที่ 5-4 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ 95
ตารางที่ 5-4 ทดสอบกรณีไม่ใส่ไฟล์ 100
ตารางที่ 5-5 ทดสอบกรณีไม่ใส่ไฟล์ที่มีนามสกุลเอกซ์เอ็มแอลต้นทาง 100
ตารางที่ 5-6 ทดสอบกรณีใส่ไฟล์ที่มีนามสกุลต้นทางเอกซ์เอ็มแอลแต่ไม่มีโครงสร้างที่ผู้วิจัยได้ออกแบบไว้ข้างต้น 101

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญรูปภาพ

หน้า

รูปที่ 2-1 แผนภูมิแกนต์แสดงการจัดกระบวนการแบบไม่มีพรีเอ็มทีฟ 7

รูปที่ 2-2 แผนภูมิแกนต์แสดงการจัดกระบวนการแบบไม่มีพรีเอ็มทีฟ 9

รูปที่ 2-3 ตัวอย่างแผนภาพเวลา [7] 15

รูปที่ 2-4 สวิตช์ไฟ [14] 19

รูปที่ 2-5 ออโตมาตอนที่มีนาฬิกา 1 ตัว [6] 19

รูปที่ 2-6 การทรานซิชั่นที่มีนาฬิกา 2 ตัว [3] 20

รูปที่ 2-7 แผนภาพเวลา [3] 21

รูปที่ 2-8 ไทมด์ออโตมาตอนทั้งหมด 4 เทมเพลต 22

รูปที่ 2-9 วิธีงานวิจัย [5] 24

รูปที่ 2-10 แสดงหน้าโปรแกรม UPPAAL [15] 25

รูปที่ 2-11 แบบจำลองไทมด์ออโตมาตา [15] 26

รูปที่ 2-12 ตัวอย่างการทำงานประสานกันของแบบจำลองโคมไฟ 2 ออโตมาตอน [15] 28

รูปที่ 3-1 ภาพรวมของการทำวิทยานิพนธ์ 31

รูปที่ 3-2 แผนภาพเวลา Task A และ Task B จะจับคู่ออโตมาตอนได้ 3 แบบได้แก่ Task A, Task B และ Main 33

รูปที่ 3-3 การจับคู่สถานะเริ่มต้นของแผนภาพเวลา TDTP 33

รูปที่ 3-4 การจับคู่ช่วงเวลาของสถานะเริ่มต้นของแผนภาพเวลา TDTP 34

รูปที่ 3-5 การจับคู่สถานะถัดไปของสถานะเริ่มต้นแผนภาพเวลา TDTP 35

รูปที่ 3-6 จับคู่ช่วงเวลาของสถานะ p1run1 และ p2run1 แผนภาพเวลา TDTP 36

รูปที่ 3-7 เพิ่มสถานะโดเมนในแผนภาพเวลา TDTP เมื่อมีลำดับความสำคัญสูงเข้าแทรก 37

รูปที่ 3-8 เพิ่มสถานะโดเมนในแผนภาพเวลา TDTP กลับมาสถานะ “run” 37

รูปที่ 3-9 ผลลัพธ์ของไทมด์ออโตมาตา Task A และ Task B จากข้อมูลนำเข้าของแผนภาพเวลา ... 38

รูปที่ 3-10 ตัวอย่างแท็กเอกซ์เอ็มแอลของแผนภาพเวลา 40

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 3-11 ตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของแผนภาพเวลา 41

รูปที่ 3-12 ตัวอย่างไทมด์ออโตมาตาในรูปแบบของเอกสารเอกซ์เอ็มแอล 43

รูปที่ 3-13 ตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ 46

รูปที่ 3-14 ผลลัพธ์การแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ 51

รูปที่ 4-1แผนภาพยูสเคสของ “Transforming Timing Diagram into Timed Automata Converter” 52

รูปที่ 4-2 แผนภาพกิจกรรมของเครื่องมือแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ 62

รูปที่ 4-3 ภาพรวมของแผนภาพคลาสเครื่องมือ “Timing Diagram into Timed Automata Converter” 63

รูปที่ 4-4 แผนภาพแพ็กเกจของเครื่องมือ “Timing Diagram into Timed Automata Converter” ที่พัฒนา 64

รูปที่ 4-5 แผนภาพคลาสย่อยในโมเดลชื่อ Coordinates 65

รูปที่ 4-6 แผนภาพคลาสย่อยในโมเดลชื่อ Input 66

รูปที่ 4-7 แผนภาพคลาสย่อยในโมเดลชื่อ Output 68

รูปที่ 4-8 แผนภาพคลาสย่อยในเซอร์วิส 69

รูปที่ 4-9 แผนภาพคลาสย่อยในเว็บ 70

รูปที่ 4-10 แผนภาพคลาสย่อยใน validator 70

รูปที่ 4-11 แผนภาพลำดับงานของกรณีผู้ใช้งาน Import Timing Diagram File 71

รูปที่ 4-12 แผนภาพลำดับงานของกรณีกรณีใช้งาน Transform Timing Diagram File to Timed Automata File 72

รูปที่ 4-13 แผนภาพดีพอยเมนท์การพัฒนาเครื่องมือการแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ 72

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 4-14 แผนภาพการทำงานเครื่องมือแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ 73
รูปที่ 4-15 หน้าจอหลักของเครื่องมือ “Timing Diagram into Timed Automata Converter”การแปลง 75
รูปที่ 4-16 หน้าจอการเลือกนำเข้าไฟล์แผนภาพเวลา TDTP สกุลเอกซ์เอ็มแอล 75
รูปที่ 4-17 หน้าชื่อจอเอกสารนำเข้า 76
รูปที่ 4-18 หน้าจอการแปลงข้อมูลนำเข้า 76
รูปที่ 4-19 หน้าจอนำไฟล์ที่แปลงเข้าเครื่องมือทวนสอบ UPPAAL 77
รูปที่ 4-20 หน้าจอการแบบจำลองที่เข้าเครื่องมือ UPPAAL 77
รูปที่ 5-1 แผนภาพเวลาของกรณีทดสอบที่ 1 80
รูปที่ 5-2 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 1 83
รูปที่ 5-3 ผลการทวนสอบจากเครื่องมือทวนสอบ UPPAAL ของกรณีศึกษาที่ 1 88
รูปที่ 5-4 แผนภาพเวลาของกรณีทดสอบที่ 2 89
รูปที่ 5-5 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 2 90
รูปที่ 5-6 ผลการทวนสอบจากเครื่องมือทวนสอบ UPPAAL ของกรณีที่ 2 92
รูปที่ 5-7 แผนภาพเวลาของกรณีทดสอบที่ 3 93
รูปที่ 5-8 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 3 96
รูปที่ 5-9 ผลการทวนสอบจากเครื่องมือทวนสอบ UPPAAL ของกรณีที่ 3 99

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ระบบเวลาจริง (Realtime System) [1] มีความสำคัญกับผู้ใช้งานซอฟต์แวร์และการพัฒนาระบบซอฟต์แวร์ที่ให้ความสำคัญลักษณะงานที่ต้องใช้เวลา เนื่องจากความต้องการของผู้ใช้ระบบซอฟต์แวร์บางประเภทมีความต้องการในการใช้งานจะขึ้นอยู่กับเวลาเป็นหลักและพัฒนาภายใต้ข้อจำกัดที่เข้มงวดในเวลาเส้นตาย (Deadline) หรือถ้าหากผู้ใช้ระบบซอฟต์แวร์บางประเภทต้องการระบบที่สามารถยืดหยุ่นเวลาได้และไม่ก่อให้เกิดความเสียหายแก่ระบบมากนัก ในอุตสาหกรรมซอฟต์แวร์ปัจจุบัน การพัฒนาซอฟต์แวร์มีหลากหลายรูปแบบมีทั้งการพัฒนาซอฟต์แวร์ขนาดใหญ่และการพัฒนาซอฟต์แวร์ขนาดเล็ก การพัฒนาซอฟต์แวร์ขนาดใหญ่อาจจะมีความซับซ้อนสูงหรืออาจจะมีโอกาสเกิดข้อผิดพลาดสูง ข้อผิดพลาดดังกล่าวอาจจะเป็นเวลา ค่าใช้จ่าย ทรัพยากร ที่สามารถเกิดการสูญเสียได้

ดังนั้นอุตาสหกรรมซอฟต์แวร์จึงต้องตระหนักในขั้นตอนการออกแบบซอฟต์แวร์ของระบบเวลาจริง นั่นย่อมเป็นสิ่งสำคัญในการส่งมอบซอฟต์แวร์ให้กับผู้ใช้เพื่อให้ตรงกับความต้องการและลดข้อผิดพลาดดังกล่าวข้างต้น จึงต้องทวนสอบในขั้นตอนการออกแบบซอฟต์แวร์ การทวนสอบเชิงรูปนัย (Formal Verification) [2] จึงเป็นทางเลือกหนึ่งที่สามารถทวนสอบพฤติกรรมที่มีเวลาเข้ามาเกี่ยวข้องและเป็นที่นิยม รวมไปถึงในขณะที่การทดสอบซอฟต์แวร์แบบเรียลไทม์ดำเนินการเพียงขั้นตอนการออกแบบซอฟต์แวร์เท่านั้น การทวนสอบเชิงรูปนัยเป็นการพิสูจน์เพื่อให้แน่ใจว่าระบบซอฟต์แวร์ที่เป็นเป้าหมายมีคุณสมบัติปลอดภัยโดยใช้วิธีทางคณิตศาสตร์ การทวนสอบส่วนใหญ่เริ่มต้นด้วยการสร้างตัวแทนนามธรรมเพื่อจำลองระบบโลกแห่งความจริงเรียกว่าวิธีเชิงรูปนัย (Formal method) และมักจะทำได้โดยการใช้เครื่องมือทางคณิตศาสตร์และตรรกศาสตร์มาช่วยพิสูจน์แบบจำลองที่เป็นไปได้ มีหลากหลายรูปแบบที่ได้พิจารณากระบวนการทางซอฟต์แวร์โดยบูรณการวิธีเชิงรูปนัยเข้าด้วยกัน นอกจากนั้นระบบเวลาจริงจะทำงานโดยมีการประมวลผลที่ส่งผลต่อประสิทธิภาพของระบบ ถ้ามีการทำงานของกระบวนการ (Process) เข้ามาเกี่ยวข้อง โดยการทำงานของกระบวนการจะต้องมีการใช้ทรัพยากรของระบบอย่างจำกัด ทำให้ต้องมีการจัดลำดับเข้าไปใช้งาน ดังนั้นซีพียูจึงเป็นทรัพยากรที่จำเป็นต่อการจัดสรรกระบวนการเข้าไปใช้งาน และเมื่อมีกระบวนการทำงานหลายงานพร้อมกันโดยที่ซีพียูทำงานตัวเดียว จะเป็นผลทำให้เกิดข้อจำกัดว่ากระบวนการไม่สามารถทำงานพร้อม ๆ กันหลายงานได้ เนื่องจากซีพียูสามารถทำงานได้แค่ครั้งละหนึ่งงาน ณ ช่วงเวลาหนึ่งเท่านั้น ถ้าหากมีกระบวนการที่เร่งด่วนหรือจำเป็นต้องทำก่อน เราจะต้องหาวิธีการจัดการอย่างไรให้มีประสิทธิภาพ และสามารถทำงานได้ในปริมาณงานที่มากขึ้นกว่าการให้ซีพียูทำงาน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ให้เสร็จทีละกระบวนการ ทำให้ต้องมีการจัดลำดับความสำคัญ (Priority) ของกระบวนการในเวลาที่กำหนดสำหรับเข้าไปใช้งานซีพียูและการจัดกระบวนการเชิงพรีเอ็มทีฟ (Preemptive Scheduling) ในเวลาที่กระบวนการโดนแทรกกลางคัน

มีหลายงานวิจัยที่นำเสนอแนวทางการแก้ไขปัญหาข้างต้น โดยมีงานวิจัยได้นำเสนอวิธีการแปลงแผนภาพเวลามาเป็นไทมด์ออโตมาตา [3], [4], [5] โดยไม่ได้คำนึงลำดับความสำคัญของกระบวนการ เมื่อมีการทำงานร่วมกันในระบบเวลาจริงภายใต้ลำดับความสำคัญที่ต่างกัน และมีการทำพรีเอ็มทีฟเมื่อโดนแทรกกลางคัน เนื่องจากคุณสมบัติข้อจำกัดของระบบเวลาจริงที่มีซีพียูเพียงตัวเดียวนั้น ควรต้องคำนึงถึงลำดับความสำคัญของกระบวนการเพื่อไม่ให้เกิดความเสียหายแก่ระบบ ให้งานดำเนินการภายในเส้นตายและสามารถเข้าแทรกเวลา เมื่อเกิดภาวะฉุกเฉินหรือความปลอดภัยอย่างมีประสิทธิภาพ และผู้วิจัยได้ศึกษางานวิจัยที่เกี่ยวข้องเพื่อนำมาศึกษาเกี่ยวกับการใช้เครื่องมือการแปลงและสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟที่ผู้วิจัยสนใจ ผู้วิจัยจึงเห็นความสำคัญของปัญหาดังกล่าว

การทวนสอบจะต้องมีสิ่งที่จำเป็น 2 สิ่งนั่นคือแบบจำลองเชิงรูปนัยของระบบ และคุณลักษณะเงื่อนไขที่ใช้ทวนสอบที่เขียนด้วยสูตรเชิงเวลา โดยส่วนมากที่แพร่หลายนั่นก็คือการอธิบายแบบจำลองเชิงรูปนัยโดยใช้แผนภาพที่สามารถแสดงสถานะและการเปลี่ยนแปลงไปมาของสถานะได้ และ ไทมด์ออโตมาตา (Timed Automata) [3] ที่นิยมนำมาใช้อธิบายแบบจำลองเชิงรูปนัยที่เกี่ยวข้องกับระบบเวลาจริงอย่างแพร่หลาย และสามารถทวนสอบพฤติกรรมของระบบเวลาจริงได้ เนื่องจากแผนภาพเวลาไม่สามารถนำมาทวนสอบได้จึงต้องใช้การทวนสอบเชิงรูปนัย (Formal Verification) เข้ามาช่วยในการทวนสอบระบบเวลาจริง รวมไปถึงผู้วิจัยได้ใช้ไทมด์ออโตมาตา (Timed Automata) [6] ในการอธิบายแบบจำลองเชิงรูปนัยที่เกี่ยวข้องกับระบบเวลาจริงอย่างแพร่หลายและสามารถทวนสอบพฤติกรรมของระบบเวลาจริงได้

โดยวิทยานิพนธ์นี้แผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟและการทดลอง ผู้วิจัยจะสร้างแบบจำลองและการทวนสอบการออกแบบซอฟต์แวร์ของการเข้าไปใช้งานซีพียูในระบบเวลาจริงว่าเป็นไปตามข้อกำหนดคุณสมบัติความต้องการของระบบหรือไม่ ตัวอย่างเช่น คุณสมบัติภาวะติดตาย (Deadlock Properties), คุณสมบัติการเข้าถึง (Reachability Properties), คุณสมบัติความปลอดภัย (Safety Properties) เป็นต้น นอกจากนี้ผู้วิจัยใช้แผนภาพยูเอ็มแอล (UML) [7] ในการอธิบายแบบจำลองต่าง ๆ หรือเป็นภาษาสัญลักษณ์รูปภาพมาตรฐานสำหรับใช้ในการสร้างแบบจำลองเชิงวัตถุที่ให้ผู้ใช้กับนักพัฒนาระบบสามารถติดต่อสื่อสารเข้าใจได้ง่ายขึ้น แผนภาพยูเอ็มแอลในงานวิทยานิพนธ์ในเล่ม ผู้วิจัยเลือกเป็นแผนภาพเวลา (Timing Diagram) [7] เป็นแผนภาพที่ถูกใช้อธิบายการแสดงสถานะที่เราสนใจ ณ ช่วงเวลาใด ๆ แผนภาพเวลานำมาประยุกต์ได้หลากหลายอย่าง เช่น ซอฟต์แวร์ธุรกิจ (Business Software) ใช้แผนภาพเวลา

ในการดูสถานะของการควบคุมระบบหัวฉีดอัตโนมัติในโรงงาน สถานะในซอฟต์แวร์แบบฝังตัว (Embedded Software) สถานะข้อกำหนด (Specifications) ในระบบเวลาจริง และงานที่มีการประมวลผลต่าง ๆ ที่มีการตอบสนอง เป็นต้น เนื่องจากแผนภาพเวลาไม่สามารถนำมาทวนสอบได้จึงต้องใช้การทวนสอบเชิงรูปนัย เข้ามาช่วยในการทวนสอบระบบเวลาจริง รวมไปถึงผู้วิจัยได้ใช้ไทมด์ออโตมาตา ในการอธิบายแบบจำลองเชิงรูปนัยที่เกี่ยวข้องกับระบบเวลาจริงและสามารถทวนสอบพฤติกรรมของระบบเวลาจริงได้ ดังนั้นในงานวิทยานิพนธ์นี้ จึงนำเสนอแผนภาพเวลาหลายกระบวนการมาแปลงเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟโดยใช้เครื่องมือที่ชื่อว่า UPPAAL เพื่อทวนสอบคุณสมบัติดังกล่าวและพัฒนาเครื่องมือเพื่อรองรับการแปลงแผนภาพเวลาหลายกระบวนการมาแปลงเป็นไทมด์ออโตมาตาแบบอัตโนมัติ

1.2 วัตถุประสงค์ของโครงงาน

1) ออกแบบกฎการแปลงแผนภาพเวลาให้เป็นไทมด์ออโตมาตาที่สนับสนุนการทำพรีเอ็มทีฟ

2) พัฒนาเครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่สนับสนุนการทำพรีเอ็มทีฟ

1.3 ขอบเขตของโครงงาน

1) ออกแบบกฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่สนับสนุนการทำพรีเอ็มทีฟ

2) พัฒนาเครื่องมือในการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่สนับสนุนการทำพรีเอ็มทีฟ

2.1 ข้อมูลนำเข้า (input) ของงานวิทยานิพนธ์จะเป็นแผนภาพเวลาของกระบวนการ ในรูปแบบไฟล์สกุลเอกซ์เอ็มแอลและมีหมายเลขลำดับความสำคัญของกระบวนการสามารถนำเข้าแผนภาพเวลาได้อย่างน้อย 2 กระบวนการ

2.2 แผนภาพเวลาที่นำเข้าแยกกันเป็นอิสระต่อกัน

2.3 สัญลักษณ์ที่ใช้แผนภาพเวลาจะใช้สัญลักษณ์ สถานะ, ข้อความ, เส้นชีวิต, ช่วงเวลาจำกัดเป็นอย่างน้อย

2.4 ข้อมูลส่งออก (output) เป็นไทมด์ออโตมาตาในรูปแบบแฟ้มไฟล์สกุลเอกซ์เอ็มแอล และเมื่อมีกระบวนการมาทำงานร่วมกันจะมีการสนับสนุนการทำพรีเอ็มทีฟ

3) ใช้กรณีศึกษา 3 กรณีของแผนภาพเวลาของกระบวนการเพื่อที่นำมาแปลงเป็นไทมด์ออโตมาตา ได้แก่ การแยกลำดับความสำคัญในกรณีผู้ป่วยเข้าใช้ห้องฉุกเฉิน (Emergency Room (ER)) เป็นต้น

4) นำผลลัพธ์ที่ได้จากการแปลงเป็นไทมด์ออโตมาตามาทวนสอบในเครื่องมือ UPPAAL โดยการทวนสอบจะทวนสอบแบบสืบค้นคุณสมบัติได้แก่ คุณสมบัติการเข้าถึง, คุณสมบัติการติดตาย, คุณสมบัติความปลอดภัย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.4 ขั้นตอนการดำเนินงานของวิทยานิพนธ์

1) ศึกษาค้นคว้าทฤษฎีที่เกี่ยวข้องกับงานวิทยานิพนธ์

1.1 ศึกษาระบบเวลาจริง

1.2 ศึกษาการจัดลำดับซีพียูและการจัดกำหนดรายการเวลาบนระบบเวลาจริง

1.3 ศึกษาทฤษฎีไทมด์ออโตมาตา

1.4 ศึกษาทฤษฎีแผนภาพเวลา

2) ศึกษาเครื่องมือที่เกี่ยวข้องในงานวิจัย

3) กำหนดแนวคิด วิธีวิจัย และขอบเขตงานวิจัย

4) ทดสอบและปรับปรุงเครื่องมือที่ใช้ในงานวิทยานิพนธ์

5) ประเมินผลลัพธ์การทดสอบ

6) สรุปผลการวิจัยและข้อเสนอแนะ

7) จัดทำรูปเล่มวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) กฎการแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาที่สนับสนุนการทำพรีเอ็มทีฟ

2) เครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่สนับสนุนการทำพรีเอ็มทีฟ

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์เล่มนี้ ได้รับการตีพิมพ์เป็นบทความวิชาการเรื่อง“Simulation of Preemptive Scheduling of the Independent Tasks Using Timed Automata” โดย อมรัตน์ พิมโคตร และรองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ ในงานประชุมวิชาการ In 2021 10th International Conference on Software and Computer Applications (ICSCA 2021) ระหว่างวันที่ 23 ถึง 26 กุมภาพันธ์ พ.ศ. 2564 ณ เมืองกัวลาลัมเปอร์, ประเทศมาเลเซีย

บทที่ 2
ทฤษฎี งานวิจัย และเครื่องมือที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ในหัวข้อนี้ผู้วิจัยได้ศึกษาและค้นคว้าทฤษฎีที่เกี่ยวข้องกับงานวิทยานิพนธ์ ผู้วิจัยได้คำนึงถึงเรื่องระบบเวลาจริง, แผนภาพเวลาและเมื่อมีหลาย ๆ กระบวนการทำงานพร้อมกัน แต่ซีพียูทำงานได้ครั้งละหนึ่งกระบวนการ จึงทำให้ต้องคำนึงถึงการจัดลำดับความสำคัญของกระบวนการนั้นเข้าไปใช้ซีพียู ผู้วิจัยจึงเสนอการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่สนับสนุนการทำพรีเอ็มทีฟ เพื่อทวนสอบคุณสมบัติของระบบโดยใช้เครื่องมือ UPPAAL ผู้วิจัยจะแสดงการอธิบายทฤษฎีที่เกี่ยวข้องไว้ด้านล่างดังต่อไปนี้

2.1.1 ระบบเวลาจริง

ระบบเวลาจริง คือ [1] ระบบจะต้องมีข้อจำกัดที่ชัดเจนของเวลาตอบสนองหรือความเสี่ยงที่อาจส่งผลกระทบต่อความรุนแรงรวมถึงความล้มเหลว (Failure) ในความล้มเหลวนี้หมายความว่าระบบไม่สามารถตอบสนองความต้องการที่ระบุไว้ในข้อกำหนดระบบได้ พฤติกรรมการทำงานในระบบเวลาจริงขึ้นอยู่กับความต้องการ 2 ประการนั่นก็คือ ความถูกต้องเชิงตรรกะและความถูกต้องเชิงเวลา

ระบบเวลาจริงมักจะมีปฏิกิริยา (Reactive) หรือเป็นระบบฝังตัว (Embedded Systems) ระบบที่มีปฏิกิริยาจะเป็นระบบที่มีการกำหนดเวลาว่า จะแสดงปฏิสัมพันธ์อย่างต่อเนื่องกับสภาพแวดล้อม ตัวอย่างเช่น ระบบควบคุมอัคคีภัยมีการตอบสนองเมื่อนักบินกดปุ่ม ส่วนระบบฝังตัวจะเป็นระบบคอมพิวเตอร์ขนาดเล็กที่ฝังไว้ในอุปกรณ์เครื่องอิเล็กทรอนิกส์หรือเครื่องใช้ไฟฟ้าต่าง ๆ เพื่อเพิ่มความสามารถให้กับอุปกรณ์ ตัวอย่างเช่น รถยนต์สมัยใหม่ประกอบด้วยคอมพิวเตอร์ฝังตัวจำนวนมากที่ไว้ควบคุมการฉีดน้ำมันเชื้อเพลิง, การปรับใช้ถุงลมนิรภัย (Airbag), การเบรก เป็นต้น ในงานวิจัยส่วนมากจะกล่าวถึงระบบเวลาจริงซึ่งสามารถจำแนกได้ 2 ประเภทดังต่อไปนี้

1. ระบบเวลาจริงแบบอ่อน (Soft Real Time System) เป็นระบบที่สามารถยืดหยุ่นเวลาได้โดยไม่ส่งผลกระทบต่อชีวิตมนุษย์ ตัวอย่างเช่น แชท, เอทีเอ็ม (ATM), การสตรีมมัลติมีเดีย เป็นต้น

2. ระบบเวลาจริงแบบแข็ง (Hard Real Time System) เป็นระบบที่ต้องทำตามเวลาที่กำหนด ไม่สามารถยืดหยุ่นเวลาได้ ถ้าไม่ทำตามเวลาที่กำหนดจะเกิดความเสียหายแก่ระบบหรือการสูญเสียถึงชีวิตได้ ตัวอย่างเช่น เครื่องช่วยหายใจ ถ้าเกิดเครื่องพังจะกระทบกับชีวิตมนุษย์ เป็นต้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.2 ประเภทอัลกอริทึมสำหรับการทำกำหนดการเวลา [8], [9]

ประเภทอัลกอริทึมสำหรับการทำกำหนดการเวลา (Scheduling Algorithms Taxonomy) เป็นการจัดกระบวนการเพื่อเข้าไปใช้งานซีพียู ซึ่งแบ่งออกเป็น 2 ประเภท ได้แก่ การจัดกระบวนการแบบพรีเอ็มทีฟ (preemptive scheduling) และการจัดกระบวนการแบบไม่มีพรีเอ็มทีฟ (Non-Preemptive Scheduling)

1. การจัดกระบวนการแบบพรีเอ็มทีฟ เป็นการจัดกระบวนการที่สามารถถูกกระบวนการอื่นมาแทรกได้ โดยกระบวนการมาถึงในคิว (Queue) เมื่อพร้อมใช้งาน ลำดับความสำคัญของกระบวนการจะถูกเปรียบเทียบกับลำดับความสำคัญของกระบวนการอื่น ๆ ที่มีอยู่ในคิวเช่นเดียวกับกระบวนการที่ถูกประมวลผลโดยซีพียู ณ เวลานั้น กระบวนการที่กำลังดำเนินการสามารถหยุดลงเมื่อมีกระบวนการที่มีลำดับความสำคัญสูงกว่ามาแทรกกลางคัน ถ้ากระบวนการที่ถูกแทรกยังดำเนินงานไม่เสร็จ กระบวนการที่ถูกแทรกต้องรอจนกว่ากระบวนการที่แทรกดำเนินการเสร็จสิ้นจึงสามารถดำเนินงานต่อได้

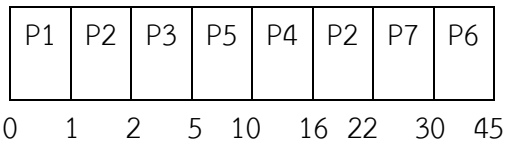
ตัวอย่างที่ 2.1 การจัดกระบวนการแบบพรีเอ็มทีฟ

ตารางที่ 2-1 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ [10]

Process Id	Arrival Time	Burst Time	Priority
P1	0	1	2
P2	1	7	6
P3	2	3	3
P4	3	6	5
P5	4	5	4
P6	5	15	10
P7	15	18	9

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในตารางที่ 2-1 ประกอบด้วย 7 กระบวนการ ได้แก่ P1, P2, P3, P4, P5, P6 และ P7, ลำดับความสำคัญ (Priority) โดยระบบจะกำหนดค่าที่น้อยที่สุดแสดงถึงลำดับความสำคัญมากที่สุดและให้ตัวเลขที่มีค่ามากที่สุดแสดงถึงลำดับความสำคัญน้อยที่สุด ตัวอย่างในตารางที่ 2-1 ลำดับความสำคัญมากที่สุดคือ ลำดับความสำคัญที่ 2 ใน P1 เพราะว่าเป็นตัวเลขที่น้อยที่สุดทั้ง 7 กระบวนการ เวลาที่แต่ละกระบวนการเข้ามาในระบบ (Arrival Time), เวลาที่กระบวนการใช้งาน (Burst Time) เป็นเวลาในการดำเนินงานของแต่ละกระบวนการ จากตารางที่ 2-1 แสดงการรอของแต่ละกระบวนการสามารถเขียนแผนภูมิแกนต์ (Gantt Chart) ได้ดังรูปที่ 2-1 ข้างล่าง



รูปที่ 2-1 แผนภูมิแกนต์แสดงการจัดกระบวนการแบบไม่มีพรีเอ็มทีฟ

การรอแต่ละกระบวนการรูปที่ 2-1 สามารถอธิบายได้ดังนี้ [11]

1. P1 เข้ามาในระบบและได้รับการจัดสรรให้ใช้ซีพียูทันที จึงไม่ต้องเสียเวลาในการรอซีพียู เวลาในการรอซีพียูจึงเท่ากับ 0 หน่วยเวลา

2. P2 ต้องรอจนกว่า P1 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 1 หน่วยเวลา

3. ในขณะที่ P2 ทำงานอยู่และมีกระบวนการใหม่เข้ามาในคิวพร้อม นั่นก็คือ P3 และกระบวนการ P3 ต้องการใช้ซีพียูเป็นเวลาน้อยกว่าและมีลำดับความสำคัญมากกว่ากระบวนการที่กำลังทำงาน P2 กระบวนการเดิมที่กำลังทำงานจะถูกขัดจังหวะให้หยุดการทำงานและคืนซีพียูให้แก่ระบบเพื่อระบบจะส่งซีพียูให้กระบวนการใหม่ที่ต้องการใช้ซีพียูเป็นเวลาน้อยกว่าและมีลำดับความสำคัญมากกว่าได้เข้าไปทำงานในซีพียูก่อน กรณีนี้เรียกการทำงานแบบพรีเอ็มทีฟ ดังนั้น P3 เวลาในการรอเท่ากับ 2 หน่วยเวลา

4. P5 ต้องรอจนกว่า P3 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 5 หน่วยเวลา

5. P4 ต้องรอจนกว่า P5 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 10 หน่วยเวลา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

6. P2 ต้องรอจนกว่า P4 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอ P4 ถึงระยะเวลาเท่ากับ 16 หน่วยเวลา

7. P7 ต้องรอจนกว่า P2 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 22 หน่วยเวลา

8. P6 ต้องรอจนกว่า P7 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 30 หน่วยเวลา

การคำนวณเวลาครบวงงานของแต่ละกระบวนการและการคำนวณการรอเวลา สูตรการคำนวณเวลาครบวงงานของแต่ละกระบวนการ

จะได้ Turnaround Time = Completion Time - Arrival Time (1)

- เวลาครบวงงานของแต่ละกระบวนการ (Turnaround Time) เป็นเวลาต่อรอบงานคือเวลาที่ผู้ใช้ต้องรอเริ่มจากนำกระบวนการเข้าสู่ระบบจนได้รับข้อมูลนำออกที่ต้องกลับมา ถ้าเวลาครบวงงานมากจะทำให้เกิดความล่าช้าต่อความต้องการของผู้ใช้
- เวลากำหนดเสร็จ (Completion Time) ของแต่ละกระบวนการ เป็นเวลาที่กระบวนการแต่ละกระบวนการเสร็จสิ้นเรียบร้อยดูจากแผนภูมิแกนต์รูปที่ 2.1 ตัวอย่างเช่น กระบวนการ P1 เวลากำหนดเสร็จสุดท้ายเท่ากับ 1 หน่วยเวลา, กระบวนการ P2 เวลากำหนดเสร็จสุดท้ายเท่ากับ 22 หน่วยเวลา

สูตรการคำนวณการรอเวลา

จะได้ Waiting Time = Turn Around Time - Burst Time (2)

- การรอเวลา (waiting time) เป็นเวลาที่กระบวนการที่ถูกรอในคิวพร้อม (Ready queue) ถ้ารอคอยน้อยที่สุดจะมีประสิทธิภาพ

จากสมการ (1), (2) และเวลากำหนดเสร็จที่ได้มาจากแผนภาพแกนต์ จะได้ดังตารางที่ 2-2 ตารางที่ 2-2 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟและผลลัพธ์การคำนวณจากสมการ (1), (2) และเวลาที่กำหนดเสร็จ [10]

Process Id	Arrival Time	Burst Time	Priority	Completion Time	Turnaround Time	Waiting Time
P1	0	1	2	1	1	0
P2	1	7	6	22	21	14
P3	2	3	3	5	3	0
P4	3	6	5	16	13	7
P5	4	5	4	10	6	1
P6	5	15	10	45	40	25
P7	15	18	9	30	24	16

จากตารางที่ 2-2 นำผลลัพธ์เวลาที่ใช้ในการรอคอยมาหาค่าเฉลี่ยเวลาที่ใช้ในการรอคอย

จะได้ Average Waiting Time = (All Waiting Time) / (Number of process) (3)

ค่าในตารางที่ 2-2 จะได้ Average Waiting Time = (0+14+0+7+1+25+16)/7 = 9 หน่วยเวลา

2. การจัดกระบวนการแบบไม่มีพรีเอ็มทีฟ เป็นการจัดกระบวนการที่ไม่สามารถถูกกระบวนการอื่นแทรกหรือขัดจังหวะได้ ต้องทำงานเสร็จสมบูรณ์ก่อนจึงจะสามารถทำงานกระบวนการอื่นต่อไปได้ กระบวนการถูกกำหนดเวลาตามหมายเลขลำดับความสำคัญที่กำหนดไว้ ถ้ามีกระบวนการทั้งหมดอยู่ในคิวพร้อมกัน ซีพียูต้องจัดให้กระบวนการนั้นให้ดำเนินการตามลำดับความสำคัญ หากกระบวนการสองกระบวนการ มีหมายเลขลำดับความสำคัญเท่ากัน กระบวนการที่มีเวลามาถึงค่าน้อยที่สุดจะถูกดำเนินการก่อน

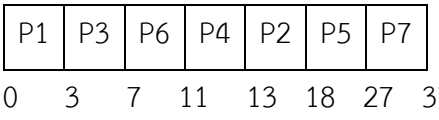
ตัวอย่างที่ 2.2 การจัดกระบวนการแบบไม่มีพรีเอ็มทีฟ

ตารางที่ 2-3 แสดงลำดับของกระบวนการทำงานของซีพียูแบบไม่มีพรีเอ็มทีฟ [10]

Process Id	Arrival Time	Burst Time	Priority
P1	0	3	2
P2	2	5	6
P3	1	4	3
P4	4	2	5
P5	6	9	7
P6	5	4	4
P7	7	10	10

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในตารางที่ 2-3 ประกอบไปด้วย 7 กระบวนการ ได้แก่ P1, P2, P3, P4, P5, P6 และ P7 นำตารางที่ 2-3 มาเขียนแผนภูมิแกนต์ จากตารางที่ 2-3 แสดงการรอของแต่ละกระบวนการสามารถเขียนแผนภูมิแกนต์ ได้ดังรูปที่ 2.2 ข้างล่าง



รูปที่ 2-2 แผนภูมิแกนต์แสดงการจัดกระบวนการแบบไม่มีพรีเอ็มทีฟ

จากแผนภูมิแกนต์จะเห็นว่าไม่มีกระบวนการมาแทรกได้ จะทำงานโดยกระบวนการที่มาก่อนจะได้ทำงานก่อนให้เสร็จสิ้นแล้วค่อยมีกระบวนการอื่นมาต่อแถวเรียงเป็นลำดับตามเวลาที่กำหนดของแต่ละกระบวนการ รอแต่ละกระบวนการในรูปที่ 2.2 สามารถอธิบายได้ดังนี้

1. P1 เข้ามาในระบบและได้รับการจัดสรรให้ใช้ซีพียูทันที จึงไม่ต้องเสียเวลาในการรอซีพียู เวลาในการรอซีพียูจึงเท่ากับ 0 หน่วยเวลา

2. P3 ต้องรอจนกว่า P1 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 3 หน่วยเวลา

3. P6 ต้องรอจนกว่า P3 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 7 หน่วยเวลา

4. P4 ต้องรอจนกว่า P6 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 11 หน่วยเวลา

5. P2 ต้องรอจนกว่า P4 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 13 หน่วยเวลา

6. P5 ต้องรอจนกว่า P2 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 18 หน่วยเวลา

7. P7 ต้องรอจนกว่า P5 ทำงานเสร็จเรียบร้อยและคืนซีพียูให้กับระบบ เวลาในการรอเท่ากับ 27 หน่วยเวลา

การคำนวณเวลาครบวงงานของแต่ละกระบวนการและการคำนวณการรอเวลาจะคำนวณได้จากสูตรที่ (1), (2) จะได้ผลลัพธ์ดังตารางที่ 2-4

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 2-4 แสดงลำดับของกระบวนการทำงานของซีพียูแบบไม่มีพรีเอ็มทีฟและผลลัพธ์จากการคำนวณจากสมการ (1), (2) และเวลาที่กำหนดเสร็จ [10]

Process Id	Arrival Time	Burst Time	Priority	Completion Time	Turnaround Time	Waiting Time
P1	0	3	2	3	3	0
P2	2	5	6	18	16	11
P3	1	4	3	7	6	2
P4	4	2	5	13	9	7
P5	6	9	7	27	21	12

ตารางที่ 2-4 แสดงลำดับของกระบวนการทำงานของซีพียูแบบไม่มีพรีเอ็มทีฟและผลลัพธ์จากการคำนวณจากสมการ (1), (2) และเวลาที่กำหนดเสร็จ [10] (ต่อ)

Process Id	Arrival Time	Burst Time	Priority	Completion Time	Turnaround Time	Waiting Time
P6	5	4	4	11	6	2
P7	7	10	10	37	30	18

จากตารางที่ 2-4 ได้ผลลัพธ์เวลาที่ใช้ในการรอคอยแล้วนำมาคำนวณหาค่าเฉลี่ยเวลาที่ใช้ในการรอคอยจากสูตร (3) จะได้ Average Waiting Time = (0+11+2+7+12+2+18)/7 = 52/7 หน่วยเวลา

จากระบบปฏิบัติการเลือกว่าจะอนุญาตให้มีการจัดกระบวนการแบบมีพรีเอ็มทีฟหรือการจัดกระบวนการแบบไม่มีพรีเอ็มทีฟ ระบบปฏิบัติการก็จะเลือกวิธีการจัดการซึ่งมีกระบวนการทั้งหมด 4 ประเภท ซึ่งแต่ละประเภทจะเหมาะกับสถานการณ์ที่แตกต่างกันไป และจะใช้กับการจัดกระบวนการแบบพรีเอ็มทีฟหรือการจัดกระบวนการแบบไม่มีพรีเอ็มทีฟอย่างใดอย่างหนึ่งหรือใช้ทั้ง 2 แบบก็ได้ ซึ่งจะแสดงรายละเอียดในหัวข้อ 2.1.3

2.1.3 นโยบายการทำกำหนดการเวลา

นโยบายการทำกำหนดการเวลา (Scheduling Policy) [9] เป็นวิธีหรือหลักการที่สามารถจัดลำดับความสำคัญของรายการเวลาบนระบบเวลาจริง ซึ่งจะเหมาะกับสถานการณ์ต่าง ๆ ในระบบปฏิบัติการ ซึ่งแบ่งออกเป็น 4 ประเภท

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1. First Come First Served Scheduling (FCFS) มาก่อนได้รับบริการก่อน ข้อดีของกระบวนการนี้จะมีคิวชัดเจน ข้อเสียคือถ้ากระบวนการที่มาก่อนทำงานค่อนข้างนาน ทำให้กระบวนการที่รอต่อคิวอยู่ต้องคอยนาน ทำให้การรอเวลานาน กระบวนการนี้ไม่มีการจัดการพรีเอ็มทีฟเพราะไม่มีกระบวนการอื่นมาแทรกกลางคัน

2. Short First Scheduling (SJF) กระบวนการนี้จะใช้เวลาน้อยที่สุดทำงานก่อน ถ้ามีกระบวนการมาที่มีเวลาเท่ากัน กระบวนการนี้จะทำงานแบบ วิธีมาก่อนได้รับบริการก่อน แต่ข้อเสียคือ จะไม่สามารถคาดเดาได้ว่า ซีพียูของกระบวนการนี้ใช้เวลาเท่าไหร่ ทำให้ไม่สามารถใช้งานในเวลาจริงได้ กระบวนการนี้มีทั้งการจัดการแบบไม่มีพรีเอ็มทีฟเมื่อกระบวนการมาถึงพร้อม ๆ กัน และการ

จัดการแบบมีพรีเอ็มทีฟเมื่อกระบวนการมาถึงต่างเวลากัน กระบวนการที่มาที่หลังมีเวลาน้อยกว่ากระบวนการที่กำลังทำงานอยู่ สามารถเข้าแทรกได้

3. Round-Robin Scheduling กระบวนการนี้จะรองรับผู้ใช้จำนวนมากและทำงานเป็นรอบ ๆ โดยจะกำหนดเวลาที่สั้นและเท่ากันจนกว่ากระบวนการจะทำงานเสร็จสมบูรณ์ ช่วงเวลาที่กำหนดการเข้าทำงานเรียกว่า เวลาควอนตัม (Time Quantum) หรือการแบ่งเวลา (Time Slice) โดยกระบวนการจะเข้าไปทำงาน แต่ละกระบวนการจะทำงานในเวลาที่กำหนด กระบวนการที่ยังทำงานไม่เสร็จก็ต้องออกมาต่อแถวแล้วทำงานต่อไปเรื่อย ๆ ข้อดีของวิธีนี้คือ จะเหมาะกับซีพียูที่มีงานเข้าเป็นจำนวนมาก กระบวนการใดใช้เวลาน้อย ๆ ก็จะไม่ต้องต่อแถว ข้อเสียของวิธีนี้คือ หากมีกระบวนการมีระยะเวลาที่มากเกินไปจะทำวิธีมาก่อนได้รับบริการ กระบวนการนี้เป็นการจัดการแบบมีพรีเอ็มทีฟเท่านั้นเพราะถ้ากระบวนการไม่เสร็จในเวลาที่กำหนดจะโดนแทรกตลอด

4. Priority Scheduling กระบวนการนี้จะนึกถึงเรื่องลำดับความสำคัญเป็นอันดับแรก การทำงานของกระบวนการ หากมีกระบวนการมาพร้อมกัน ระบบปฏิบัติการก็จะให้กระบวนการที่มีความสำคัญมากกว่าเข้ามาทำงานก่อน ข้อดีของวิธีนี้คือ จะได้กระบวนการที่สำคัญ ๆ ทำก่อนกระบวนการที่ได้ลำดับความสำคัญน้อย ข้อเสียของวิธีนี้คือ กระบวนการจะถูกแช่แข็ง (Starvation) กล่าวคือ กระบวนการที่มีความสำคัญน้อยกว่าจะรอคิวนาน จนไม่ได้ทำงานเพราะถูกกระบวนการที่มีลำดับความสำคัญมากกว่ามาแย่งซีพียูไปใช้งาน ทำให้กระบวนการที่มีลำดับความสำคัญน้อยกว่าไม่มีโอกาสเข้าใช้งานซีพียู วิธีนี้มีทั้งมีการจัดการแบบไม่มีพรีเอ็มทีฟ เมื่อกระบวนการมาถึงพร้อม ๆ กัน และการจัดการแบบมีพรีเอ็มทีฟเมื่อกระบวนการมาถึงต่างเวลากัน กระบวนการมาทีหลังมีลำดับความสำคัญสูงกว่าก็จะเข้าแทรกกลางคัน จากรายละเอียดข้างต้นจะสรุปได้ดังตารางที่ 2-5

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 2-5 สรุปการจัดกำหนดรายการจัดกระบวนการแบบพรีเอ็มทีฟและกระบวนการแบบไม่มีพรีเอ็มทีฟ

การจัดกำหนดรายการเวลา	Preemptive	Non Preemptive
1. กระบวนการมาก่อนได้รับบริการก่อน (First Come First Served Scheduling)	✕	✓
2. ให้กระบวนการที่ใช้เวลาน้อยที่สุดได้ทำงานก่อน (Short First Scheduling)	✓	✓

ตารางที่ 2-5 สรุปการจัดกำหนดรายการจัดกระบวนการแบบพรีเอ็มทีฟและกระบวนการแบบไม่มีพรีเอ็มทีฟ (ต่อ)

การจัดกำหนดรายการเวลา	Preemptive	Non Preemptive
3. สลับให้แต่ละกระบวนการได้ทำงานคนละหนึ่งช่วงเวลา (Round-Robin Scheduling)	✓	✗
4. ให้กระบวนการที่มีความสำคัญมากกว่าได้ทำงานก่อนกระบวนการที่มีลำดับความสำคัญน้อยกว่า (Priority Scheduling)	✓	✓

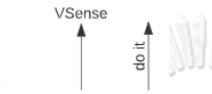
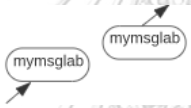
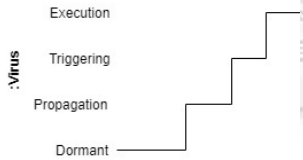
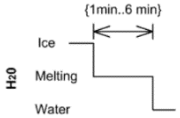
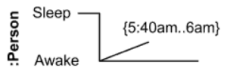
ผู้วิจัยสนใจข้อ 4 คือ Priority Scheduling เนื่องจากผู้วิจัยคำนึงถึงระดับความสำคัญเป็นหลักตามตารางกำหนดเวลา จึงต้องให้งานที่มีลำดับความสำคัญมากกว่ามาก่อน เช่น ในการคำนึงถึงเรื่องความปลอดภัย รถฉุกเฉินมีคนเจ็บควรที่จะให้รถฉุกเฉินไปก่อนเพราะมีคนเจ็บต้องรีบนำส่งโรงพยาบาล เป็นต้น ลำดับถัดไปจะพูดถึง แผนภาพเวลา ซึ่งจะใช้ดูพฤติกรรมการปฏิสัมพันธ์ที่เกิดขึ้นเพื่อแสดงการเปลี่ยนแปลงหรือเงื่อนไขของคลาส (Class) ในระบบเวลาจริง

2.1.4 แผนภาพเวลา

แผนภาพเวลา [7] เป็นแผนภาพแบบปฏิสัมพันธ์ (Interaction Diagram) ใน UML เวอร์ชัน 2.0 ซึ่งวัตถุประสงค์หลักของแผนภาพจะเกี่ยวข้องกับเวลา แผนภาพเวลาใช้อธิบายพฤติกรรมของคลาสและการปฏิสัมพันธ์ที่เกิดขึ้นภายในระหว่างเส้นชีวิตตามแกนเวลาเชิงเส้นเพื่อแสดงการเปลี่ยนแปลงของสถานะ ลักษณะของแผนภาพเวลาจะเพิ่มขึ้นจากซ้ายไปขวา ตัวอย่างสัญลักษณ์ในแผนภาพเวลามีดังตารางที่ 2-6

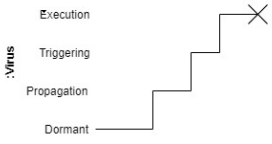
จุฬาลงกรณ์มหาวิทยาลัย CHULALONGKORN UNIVERSITY

ตารางที่ 2-6 ตัวอย่างสัญลักษณ์ที่มักพบในแผนภาพเวลา [7], [12]

ชื่อ	สัญลักษณ์	คำอธิบาย
Frame		สัญลักษณ์แสดงกรอบสี่เหลี่ยมล้อมรอบแผนภาพ บนซ้ายมือเป็นชื่อของแผนภาพเวลา
Message		สัญลักษณ์แสดงข้อความที่ระบุชนิดการติดต่อปฏิสัมพันธ์ เช่น อะซิงโครนัส (Asynchronous Message) การเรียก (Call) หรือ การตอบกลับ (Reply) เป็นต้น
Message Label		สัญลักษณ์แสดงป้าย (Labels) กำกับเพื่อไม่ให้เกิดความซ้ำซ้อนในแผนภาพเวลา
State		สัญลักษณ์แสดงลักษณะสถานะแต่ละประเภทแอตทริบิวต์ (Attribute) เงื่อนไขทดสอบ (Testable Condition) สถานะต่อเนื่อง (Continuous) หรือไม่ต่อเนื่อง (Discrete)
Duration Constraint		สัญลักษณ์บอกถึงช่วงเวลาที่จำกัด หากมีการละเมิดข้อจำกัดซึ่งหมายความว่าระบบนั้นถือว่าล้มเหลว จากตัวอย่าง น้ำแข็งควรละลายในน้ำ 1 ถึง 6 นาที
Time Constraint		สัญลักษณ์บอกถึงข้อจำกัดเวลาหากมีการการละเมิดข้อจำกัด ซึ่งหมายความว่าระบบนั้นถือว่าล้มเหลว ตัวอย่าง บุคคลควรตื่นนอนระหว่าง 5:40 น. ถึง 6:00 น.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

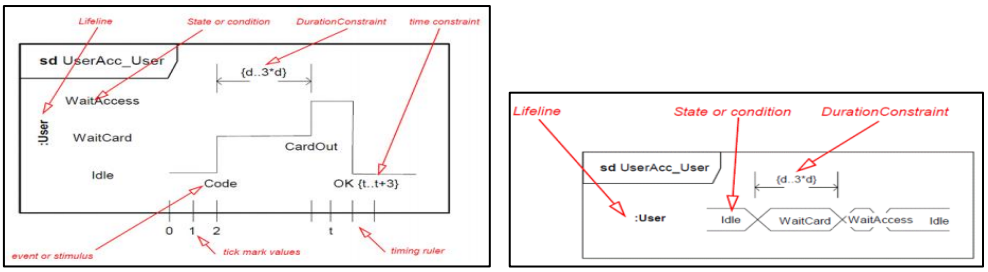
ตารางที่ 2-6 ตัวอย่างสัญลักษณ์ที่มักพบในแผนภาพเวลา (ต่อ)

ชื่อ	สัญลักษณ์	คำอธิบาย
Lifeline		สัญลักษณ์เส้นชีวิตแสดงชื่อของลักษณนาม (Classifier) หรือกรณีตัวอย่าง (Instance) แต่ละประเภทตามแนวตั้ง
Destruction Occurrence Specification		สัญลักษณ์นี้เรียกว่า เหตุการณ์การทำลาย (Destruction Event) หรือหยุด (Stop) เหตุการณ์การทำลายจะแสดงให้เห็นในรูปแบบของสัญลักษณ์ X สุดท้ายของไทม์ไลน์ (Timeline) ถ้าสัญลักษณ์นี้เกิดขึ้นซึ่งหมายถึง ไม่มีเหตุการณ์ใดเกิดขึ้นหลังจากเกิดสัญลักษณ์นี้
General Value Lifeline		แสดง ส่วนย่อย (Element) ที่เชื่อมต่อกันตามแนวนอนที่มีการตัดกันและข้างในจะระบุสถานะ

จากตารางสัญลักษณ์แผนภาพเวลาข้างต้นจะแสดงตัวอย่างให้เห็นภาพรวมดังรูปข้างล่าง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตัวอย่างที่ 2.3 องค์ประกอบแผนภาพเวลา



(1) แผนภาพเวลาเต็ม (2) แผนภาพเวลาแบบย่อ

รูปที่ 2-3 ตัวอย่างแผนภาพเวลา [7]

จาก รูปที่ 2-3 (1) แผนภาพชื่อ UserAcc_User แสดงถึงการเปลี่ยนแปลงสถานะของเส้นชีวิตในช่วงเวลา ณ ขณะหนึ่ง แนวแกน X แสดงถึงเวลาที่ผ่านไปของแต่ละสถานะ จะเห็นได้ว่ามีตัวเลข 0 ถึง t, ส่วนแนวแกน Y ถูกระบุด้วยสถานะต่าง ๆ ที่เกิดขึ้นตามลำดับ ได้แก่ สถานะ Idle, WaitCard, WaitAccess

จากรูปที่ 2-3 (2) แสดงการเปลี่ยนแปลงสถานะของเส้นชีวิต รูปที่มีลักษณะเป็นแผนภาพแบบย่อ จะเห็นว่าส่วนย่อยข้างในจะเป็นการเรียงสถานะกันตามเส้นชีวิต ได้แก่ Idle, WaitCard, WaitAccess, Idle

2.1.5 ออโตมาตา

ออโตมาตา (Automata) [13] เป็นตัวแบบทางคณิตศาสตร์ (Mathematical Model) ที่ใช้อธิบายขั้นตอนการทำงานในรูปแบบของเครื่องจักร ออโตมาตาจำกัดแบ่งได้เป็น 2 ประเภท คือ ออโตมาตาจำกัดเชิงกำหนด (Deterministic Finite Automata) และ ออโตมาตาจำกัดเชิงไม่กำหนด (Non-Deterministic Finite Automata)

1. ออโตมาตาจำกัดเชิงกำหนด สามารถรับสัญลักษณ์ตัวหนึ่งเข้ามาและจะเปลี่ยนสถานะไปยังสถานะหนึ่งตามแต่เครื่องกำหนดไว้

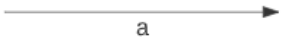
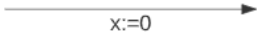
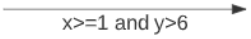
2. ออโตมาตาจำกัดเชิงไม่กำหนด สามารถมีทางเลือกว่าจะเปลี่ยนไปอยู่ในสถานะใดได้มากกว่าหนึ่งสถานะ และเครื่องสามารถที่จะรู้ได้ว่าทางเลือกใดในทางเลือกทั้งหมดที่มี จะทำให้เครื่องทำงานจนจบได้ แล้วเลือกเปลี่ยนสถานะไปทางนั้น สามารถตัดสินใจเลือกสถานะเพื่อให้ทำงานจนจบได้ ความสามารถตัดสินใจแบบนี้ เรียกว่า การเปลี่ยนสถานะเชิงไม่กำหนด (Nondeterministic State Transition)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.1.5.1 ไทมด์ออโตมาตา

ไทมด์ออโตมาตา (Timed Automata) [6] เป็นออโตมาตาจำกัดเชิงไม่กำหนด ที่มีการขยายเพิ่มเติมในเรื่องเวลามาเกี่ยวข้อง และนำมาใช้ในการวิเคราะห์พฤติกรรมและสร้างแบบจำลองเกี่ยวกับเวลา เช่น ระบบเวลาจริงในระบบคอมพิวเตอร์ ตลอดจนใช้เป็นวิธีการทวนสอบคุณสมบัติที่เกี่ยวกับเวลา ลักษณะของไทมด์ออโตมาตา จะมีค่าตัวแปรเวลา เมื่อมีการเปลี่ยนสถานะ ตัวแปรเวลาจะเพิ่มด้วยความเร็วคงที่ ซึ่งถูกคิดค้นโดย Rajeev Alur และ David L. Dill

ตารางที่ 2-7 แผนภาพสัญลักษณ์ไทมด์ออโตมาตา [13]

สัญลักษณ์	ชื่อสัญลักษณ์	คำอธิบาย
	สถานะ q หรือ โหนด q	โหนดแสดงถึงสถานะและจะเรียกเงื่อนไขเวลาที่กำหนดไว้ในโหนดว่า ค่าคงที่ (Invariant)
	สถานะเริ่มต้น s	โหนดที่ชี้ด้วยลูกศรต่อกับวงกลมแทนสถานะเริ่มต้น
	สถานะหยุด g	โหนดเส้นคู่แทนสถานะสิ้นสุด g
	ทรานซิชั่น (Transition) หรือการกระทำ (Action)	แสดงเส้นเชื่อมระหว่างโหนด ทำให้มีการเปลี่ยนสถานะ จะแสดง input action เป็น action name? และ output action เป็น action name ! ส่วนการกระทำในไทมด์ออโตมาตา มี 2 ประเภท ได้แก่ Visible Action, Not Visible Action
	Reset	เป็นการรีเซ็ตค่าหรือกำหนดค่าเริ่มต้นของตัวแปร รีเซ็ตจะอยู่บนเส้นทรานซิชั่น ค่ารีเซ็ตจะเป็น 0 และจะมีการรีเซ็ตระหว่างเปลี่ยนสถานะ
	Guards (เงื่อนไข)	เป็นการบอกเงื่อนไขเพื่อที่ทำให้เกิดการเปลี่ยนแปลงสถานะ การ์ดจะอยู่บนเส้นทรานซิชั่น ซึ่งการ์ดจะทำให้เปิดหรือปิดการใช้งานของทรานซิชั่นที่อยู่ระหว่างสถานะ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วากยสัมพันธ์ (Syntax) ของไทมด์ออโตมาตา

1. ไทมด์ออโตมาตอน (Timed Automaton (TA)) หมายถึง ทูเพิล (Tuple) ซึ่งประกอบไปด้วย 6 ส่วน โดยจะเขียนสมการได้ดังนี้

$$TA = L, \ell, A, C, E, I \quad (4)$$

โดยที่ข้างล่างจะแสดงตารางตัวแปรของสมการที่ (4) ของไทมด์ออโตมาตอน [6] ตัวแปรของไทมด์ออโตมาตอน

L เซตของโลเคชัน (Location)

A เป็นแชนเนล (Channel) และการกระทำ ซึ่งการกระทำจะประกอบด้วย 2 ส่วน

1. Visible Action ซึ่ง input action เป็น action name? และ output action เป็น action name! ทีอยู่บนแชนเนล โดยที่ (action name?, action name!) ∈ Channel

2. Not Visible เมื่อ **τ** ∈ Channel ได้แก่ การกระทำที่มาหาสถานะเดิม Action

C เป็นเซตของนาฬิกา

E เป็นเซตของเส้นเชื่อม $E \subseteq L \times B(C) \times Act \times 2^c \times L$ ซึ่งจะเขียนให้เข้าใจง่าย ได้ดังนี้ $\ell \xrightarrow{g,a,r} \ell'$, $(\ell, g, a, r, \ell') \epsilon E$ โดยที่

ℓ เป็นโลเคชันต้นทาง

g เป็นการ์ด

a เป็นการกระทำ

r เป็นเซตของนาฬิกาที่ถูกรีเซ็ต

ℓ'เป็นโลเคชันเป้าหมาย

I เป็นค่าคงที่ กำหนดไว้ที่โลเคชัน ซึ่ง $I: L \rightarrow B(C)$ โดย $B(C)$ เป็นเงื่อนไขของเวลาที่อยู่บนโลเคชัน

2. เซต $\mathrm{B}(C)$ ของข้อจำกัดนาฬิกา (Clock Constraints) หรือ การ์ด, นาฬิกา C จะนิยามโดย

$$g, g_1, g_2 ::= x \bowtie n \mid g_1 \wedge g_2 \quad (5)$$

โดยที่ $x \in C$ ซึ่ง C เป็นนาฬิกา

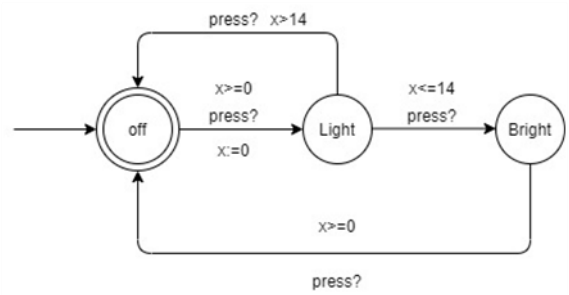
$n \in \mathbb{N}$ และ $\bowtie \in \{\leq, <, =, >, \geq\}$

เมื่อ $C = \{x, y, z\}$ เช่น $x \leq 5, x \geq 0 \wedge x < 5$ เป็นต้น

เราจะแสดงตัวอย่างไทมด์ออโตมาตาให้เข้าใจมากขึ้น ดังต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตัวอย่างที่ 2.4 การเขียนไทมด์ออโตมาตอนของสวิตช์ไฟ



รูปที่ 2-4 สวิตช์ไฟ [14]

จากรูปที่ 2.4 นาฬิกา x จะเท่ากับ 0 เมื่อค่า Off (สถานะเริ่มต้น) เข้าสู่สถานะ Light เมื่อ $x \geq 0$ และจะถูกรีเซ็ต $x := 0$ ตัวอย่างรูปที่ 2.4 สวิตช์ไฟ จะสามารถอธิบายได้ดังนี้

โดยให้ $g_t = x \geq 0$ เป็นการ์ด

$C = \{x\}$, C เป็นเซตของนาฬิกา

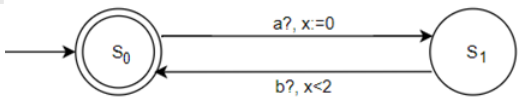
$L = \{off, Light, Bright\}$, L เป็นเซตของโลเคชัน

$E = \left\{ \begin{matrix} off \xrightarrow{g_t, press, \{x\}} Light, Light \xrightarrow{x>14, press, \emptyset} off, \\ Light \xrightarrow{x \leq 14, press, \emptyset} Bright, Bright \xrightarrow{g_t, press, \emptyset} off \end{matrix} \right\}$, E เป็นเซตของเส้นเชื่อม

$I(off) = I(Light) = I(Bright) = g_t$, g_t เป็นการ์ด

ตัวอย่างที่ 2.5 ออโตมาตอนที่มีนาฬิกา 1 ตัว

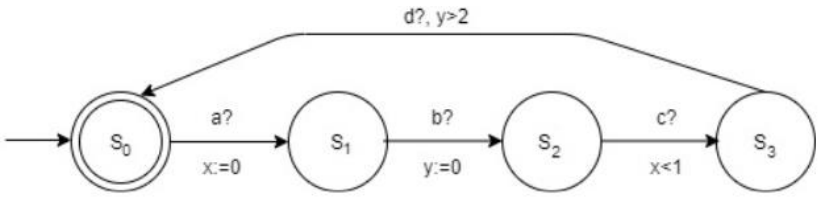
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 2-5 ออโตมาตอนที่มีนาฬิกา 1 ตัว [6]

จากรูปที่ 2-5 ออโตมาตอนเริ่มที่สถานะ s_0 มีนาฬิกาเป็น x บอกเงื่อนไข จากสถานะ s_0 ทรานซิชั่นไปยัง s_1 นาฬิกา x จะเซตเท่ากับ 0 และอ่านค่านำเข้าสัญลักษณ์ a หลังจากนั้นทรานซิชั่นจากสถานะ S_1 ไปยังสถานะ S_0 ได้จะต้องผ่านเงื่อนไขว่า $x < 2$ ถ้าเป็นไปตามเงื่อนไขจะไปยังสถานะ S_0 ได้ และจะทำซ้ำไปเรื่อย ๆ

ตัวอย่างที่ 2.6 ออโตมาตอนที่มีนาฬิกา 2 ตัว



รูปที่ 2-6 การทรานซิชั่นที่มีนาฬิกา 2 ตัว [3]

จากรูปที่ 2-6 มีสถานะทั้งหมด 4 สถานะ s_0, s_1, s_2, s_3

นาฬิกา x จะมีสถานะ s_0 ทรานซิชั่นไปยัง s_1 เซตค่า $x := 0$ และอ่านค่านำเข้า a หลังจากนั้นจากสถานะ s_2 ทรานซิชั่นไปยัง s_3 ตรวจสอบว่าเงื่อนไข $x < 1$ ไหม ถ้าเงื่อนไขเป็นจริงจะอ่านค่านำเข้า c

นาฬิกา y จะมีสถานะ s_1 ทรานซิชั่นไปยัง s_2 เซตค่า $y := 0$ และอ่านค่านำเข้า b หลังจากนั้นจากสถานะ s_3 ทรานซิชั่นไปยัง s_0 ตรวจสอบว่าเงื่อนไข $y > 2$ ถ้าเงื่อนไขเป็นจริงจะอ่านค่านำเข้า d จะสังเกตได้ว่า จะมีการหน่วงเวลาระหว่าง a และ c, ระหว่าง b และ d ดังนั้นประโยชน์ของการมีนาฬิกาหลายตัวแปรทำให้มีอิสระกันของแบบจำลองออโตมาตอน

2.2 งานวิจัยที่เกี่ยวข้อง

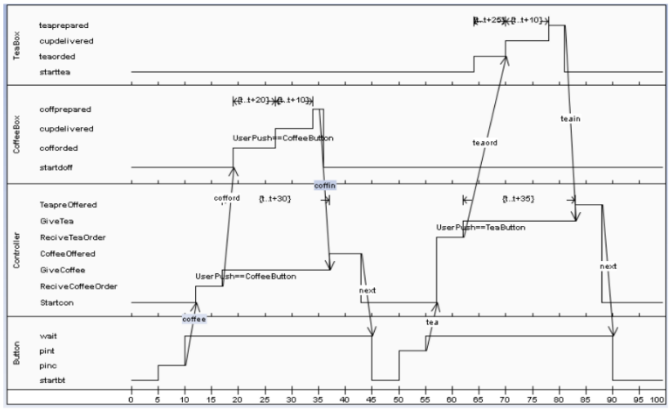
2.2.1 งานวิจัย “Unifying Modeling and Simulation Based on UML Timing Diagram and UPPAAL โดย Kangle Cui, Zongyuan Yang, Jinkui Xie, Kaiyu Wan ปี ค.ศ. 2010 [3]

งานวิจัยนี้จะกล่าวถึงแผนภาพเวลาในระบบความจริงซึ่งไม่สามารถทวนสอบได้ จึงนำเสนอการสร้างแผนภาพเวลามาเป็นไทมด์ออโตมาตา มีการออกแบบกฎการแปลงแผนภาพเวลามาเป็นไทมด์ออโตมาตา หลังจากสร้างแบบจำลองไทมด์ออโตมาตาเสร็จจะทวนสอบด้วยเครื่องมือ UPPAAL เพื่อทวนสอบว่า ระบบเป็นไปตามความต้องการหรือไม่ ตัวอย่างกรณีศึกษาฉบับนี้ใช้เป็นระบบควบคุมเครื่องทำกาแฟ สำหรับงานวิจัยจะอธิบายดังรายละเอียดข้างล่างนี้ นิยามการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาใน UPPAAL จากสูตรจะได้

$$\alpha \rightarrow \beta \text{ ที่ } \alpha \in E_{UML}, \beta \in E_{UPA} \quad (6)$$

โดย E_{UML} เป็นองค์ประกอบในแผนภาพเวลา

E_{UPA} เป็นไทมด์ออโตมาตาใน UPPAAL



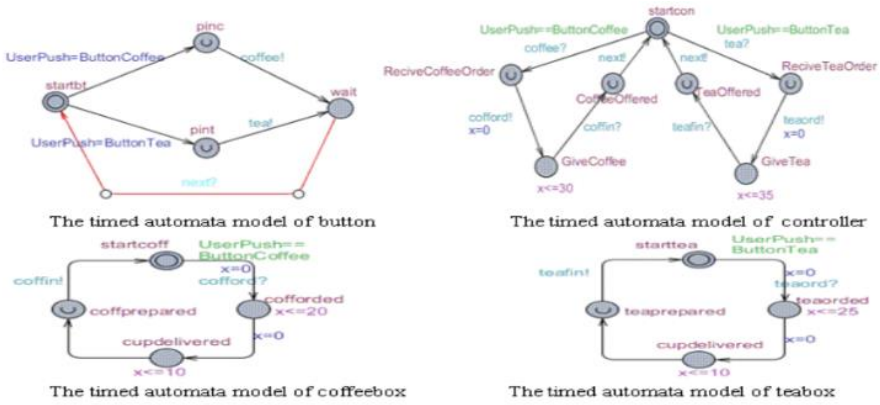
รูปที่ 2-7 แผนภาพเวลา [3]

นำแผนภาพเวลา จากรูปที่ 2-7 มาแปลงเป็นไทมด์ออโตมาตาโดยมีกฎที่ออกแบบไว้ในตารางที่ 2-8 มีทั้งหมด 7 กฎ แสดงดังข้างล่างนี้

ตารางที่ 2-8 กฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตา

หมายเลขกฎ	กฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตา
กฎที่ 1	Lifeline ⟶ Template
กฎที่ 2	State or Condition ⟶ Location
กฎที่ 3	Duration Constraint ⟶ Invariant
กฎที่ 4	Time Constraint ⟶ Guard
กฎที่ 5	Event ⟶ Guard
กฎที่ 6	Timeline ⟶ Clock
กฎที่ 7	Message⟶ Channel

จากกฎ 7 ข้อข้างต้น เมื่อนำแผนภาพเวลาจากรูปที่ 2-8 มาแปลงเป็นไทมด์ออโตมาตา โดยมาจากกรณีศึกษาจะได้เส้นชีวิตทั้งหมด 4 เส้นชีวิต ดังนั้นการแปลงเป็นไทมด์ออโตมาตอนทั้งหมด 4 เทมเพลต ได้แก่ button มี 4 เงื่อนไข, controller มี 7 เงื่อนไข, coffeebox มี 4 เงื่อนไข, teabox มี 4 เงื่อนไข ดังรูปที่ 2-8 ข้างล่าง



รูปที่ 2-8 ไทมด์ออโตมาตอนทั้งหมด 4 เทมเพลต

จากรูปที่ 2-8 นำไทมด์ออโตมาตามาทวนสอบด้วยเครื่องมือ UPPAAL เช่น ทวนสอบคุณสมบัติการเข้าถึง E <> button.wait: หมายความว่า การกดปุ่มสามารถรอเมื่อขายกาแฟหรือชา

<u>สิ่งที่นำมาใช้กับงานวิทยานิพนธ์</u> : ศึกษาตัวอย่างการแปลงแผนภาพเวลามาเป็นไทมด์ออโตมาตาและและทวนสอบด้วยเครื่องมือ UPPAAL

<u>สิ่งที่แตกต่างจากงานวิทยานิพนธ์</u> : สำหรับงานวิจัยนี้ข้อมูลนำเข้าเป็นแผนภาพเวลา 1 แผนภาพเวลาเพื่อมาแปลงไทมด์ออโตมาตา และไม่รองรับการจัดลำดับความสำคัญและสนับสนุนการทำพรีเอ็มทีฟ สำหรับงานวิทยานิพนธ์ที่จะทำจะนำเข้าแผนภาพเวลามากกว่า 1 แผนภาพเวลาแปลงมาเป็นไทมด์ออโตมาตา และรองรับการจัดลำดับความสำคัญและสนับสนุนการทำพรีเอ็มทีฟ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.2.3 งานวิจัย “Formalizing Real-Time Embedded System into Promela” โดย Punwess Sukvanich, Arthit Thongtak, Wiwat Vatanawood ปี ค.ศ. 2015 [4]

วิจัยนี้เป็นการกล่าวถึงระบบเวลาจริงแบบฝังตัว (Real-Time Embedded System) ซึ่งพบมากในปัจจุบันและมีการขยายอย่างรวดเร็วในอุตสาหกรรมตัวอย่างเช่น การบิน อุสาหกรรมยา เป็นต้น งานวิจัยนี้จึงเสนอการแปลงพฤติกรรมเวลาของระบบเวลาจริงไปเป็นโปรเมลาซึ่งเป็นภาษารูปนัยที่ใช้สร้างแบบจำลองภาษาหนึ่งและทวนสอบด้วยเครื่องมือสปิน แบบจำลองรูปนัยวิจัยนี้สนับสนุนคุณสมบัติที่สำคัญของระบบเวลาจริงแบบฝังตัว ได้แก่ การจัดการข้อจำกัดทรัพยากร, การจัดลำดับ

ความสำคัญของกระบวนการ, การทำพรีเอ็มทีฟในระบบเวลาจริง, งานที่ต้องพึ่งพาอาศัยกัน (Dependent Task)

ในขั้นตอนงานวิจัยนี้ : จะมีการกำหนดพารามิเตอร์ที่จำเป็นสำหรับระบบเวลาจริงแบบฝังตัวในแบบจำลอง ได้แก่ Name (ชื่องาน), การใช้ซีพียู, I2C (บัส), SPI (บัส), Periodic (คาบเวลาของงาน), Duration (จำนวน tick ที่ต้องใช้), ลำดับความสำคัญ และกำหนดค่านาฬิการะบบเพื่อกำหนดเวลาเริ่มในระบบเวลาจริง แบบจำลองนี้จะเขียนเป็นแบบไทม์ออโตมาตา โดยจะแบ่งการทำงานเป็น 4 สถานะ ได้แก่ open, schedule, execute, close เพื่อจัดกำหนดรายการเวลาให้เกิดการซิงโครไนซ์และมีการวิเคราะห์ความต้องการของระบบที่ทำหน้าที่แตกต่างกันไป ได้แก่ การจัดลำดับความสำคัญของกระบวนการและการครอบครองทรัพยากรหากเท่ากัน, การขัดจังหวะการทำงานโดยการทำพรีเอ็มทีฟ, แบบจำลองจองทรัพยากรได้, งานที่ต้องพึ่งพากัน

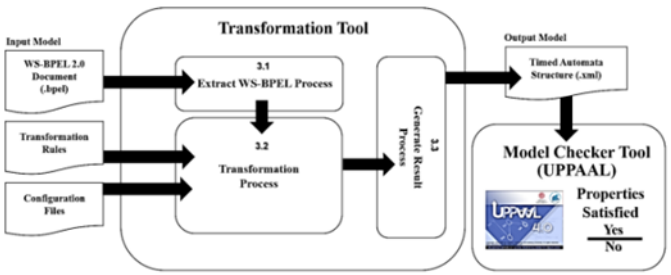
<u>สิ่งที่นำมาใช้กับงานวิทยานิพนธ์</u> : เป็นตัวอย่างการแปลงพฤติกรรมเวลาของระบบเวลาจริงแบบฝังตัวไปเป็นโปรเมลา ภาษาโปรเมลาใช้สร้างแบบจำลองภาษาหนึ่งและทวนสอบด้วยเครื่องมือสปิน และมีการวิเคราะห์คุณสมบัติความต้องการของระบบที่สำคัญในระบบเวลาจริงแบบฝังตัว ได้แก่ การทำพรีเอ็มทีฟ, การจัดลำดับความสำคัญของกระบวนการ, การจัดการทรัพยากร, งานที่ต้องพึ่งพากัน เป็นต้น

<u>สิ่งที่แตกต่างจากงานวิทยานิพนธ์</u> : ผู้วิจัยจะใช้ภาษารูปนัยเป็นไทมด์ออโตมาตาและใช้ UPPAAL ในการทวนสอบคุณสมบัติความต้องการของระบบเวลาจริงผู้วิจัยคำนึงถึงการทำพรีเอ็มทีฟ, การจัดลำดับความสำคัญของกระบวนการและงานที่เป็นอิสระต่อกัน (Independent Task) ซึ่งต่างจากงานวิจัยนี้คือ งานวิจัยงานที่ทำจะต้องพึ่งพากัน

2.2.4 งานวิจัย “Transforming WS-BPEL into Timed Automata” โดย Pramate Chandratarat และ Wiwat Vatanawood ปี ค.ศ. 2019 [5]

ในงานวิจัยนี้กล่าวถึง ดับเบิลยูเอส-บีเพล (WS-BPEL) ซึ่งเป็นแบบจำลองที่ใช้อธิบายกระบวนการธุรกิจอีกทั้งยังเป็นตัวกลางเรียกใช้เซอร์วิสต่าง ๆ มีการสร้างกระบวนการที่มีความซับซ้อนและยังมีการควบคุมกระบวนการภายในด้วยกระแสงาน (Workflow) รวมไปถึงเพื่อตอบสนองถึงความต้องการธุรกิจของลูกค้าและผู้ให้บริการ ในปัจจุบันกระบวนการทางธุรกิจเป็นบริการออกแบบที่

ซับซ้อน มีปฏิสัมพันธ์, มีการระบุข้อจำกัดด้านเวลา (Time Constraints) และมีกระแสงานที่หลากหลาย งานวิจัยนี้จะพิสูจน์ว่า ข้อจำกัดด้านเวลา และส่วนของการควบคุมกระแสงานมีความถูกต้องและสอดคล้องกันหรือไม่ ถ้าเกิดมีความซ้ำซ้อนโดยเสนอการทวนสอบข้อจำกัดด้านเวลาและส่วนของการควบคุมกระแสงานของกระบวนการดับเบิลยูเอส-บีเพลไปเป็นไทมด์ออโตมาตาเพื่อทวนสอบคุณสมบัติการคงอยู่, ความปลอดภัยและสภาวะการติดตาย โดยทวนสอบเครื่องมือชื่อว่า UPPAAL ในงานวิจัยนี้ใช้ไทมด์ออโตมาตาเป็นแบบจำลองสำหรับทวนสอบคุณสมบัติของดับเบิลยูเอส-บีเพล และมีการเสนอกฎการแปลงดับเบิลยูเอส-บีเพลแปลงเป็นไทมด์ออโตมาตา มีกรณีศึกษากระแสงานการติดตามหนี้ ตัวอย่างเช่น รับคำขอจากผู้ใช้, การเรียกใช้บริการเว็บเซอร์วิสและการตอบกลับผู้ใช้ เป็นต้น วิธีวิจัยของงานวิจัยนี้ ดังรูปที่ 2-9



รูปที่ 2-9 วิธีงานวิจัย [5]

จากรูปที่ 2-9 มีการ Extract WS-BPEL Process เป็นการวิเคราะห์โครงสร้างไฟล์นำเข้า เพื่อทำการจำแนกความสัมพันธ์ของโครงสร้างดับเบิลยูเอส-บีเพลเป็นข้อมูลสัญลักษณ์, Transformation Process เป็นการนำเอาข้อมูลที่วิเคราะห์ได้จากไฟล์นำเข้าที่อยู่ในรูปแบบของออบเจกต์ที่สืบทอดมาจากคลาสมาหาความสัมพันธ์, Adjustment-Configuration Process เป็นขั้นตอนของการปรับแต่งไฟล์ไทมด์ออโตมาตาที่สร้างขึ้นมา, Generate result Process เครื่องมือจะทำการบันทึกไฟล์ผลลัพธ์ที่ได้จากการแปลงและส่งผ่านไปยังเครื่องมือที่ใช้ในการทวนสอบแบบจำลองให้อัตโนมัติ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

<u>สิ่งที่นำมาใช้กับงานวิทยานิพนธ์</u> : ศึกษาตัวอย่างการสร้างไทมด์ออโตมาตา และการทวนสอบคุณสมบัติที่สำคัญในระบบเวลาจริง สิ่งที่นำมาใช้ในงานวิจัยคือ มีแนวคิดการสนับสนุนการใช้เครื่องมือการแปลงไทมด์ออโตมาตาแบบอัตโนมัติ

สิ่งที่นำมาใช้กับงานวิทยานิพนธ์ : ในงานวิทยานิพนธ์นี้ ผู้วิจัยจะนำเข้าแผนภาพเวลาเป็นไทมด์ออโตมาตาที่มีพรีเอ็มทีฟ แต่ในวิจัยนี้ใช้ดับเบิลยูเอส-บีเพิลเป็นการนำเข้า

2.2.5 เครื่องมือ UPPAAL

เป็นเครื่องมือสำหรับทวนสอบ (Verification) [15] ของระบบเวลาจริง โดยร่วมมือพัฒนาระหว่างมหาวิทยาลัยอุปซอลา (**Upp**sala University) ในประเทศสวีเดนและมหาวิทยาลัยอัลบอร์ (**Aal**borg University) ในประเทศเดนมาร์ก ซึ่งชื่อเครื่องมือจะนำตัวอักษรข้างหน้าของมหาวิทยาลัย 3 ตัวอักษรมารวมกัน ชื่อว่า UPPAAL และเปิดใช้งานในเวอร์ชันแรกในปี ค.ศ. 1988 เครื่องมือถูกออกแบบมาเพื่อทวนสอบความถูกต้องของการสร้างแบบจำลองที่เป็นเครือข่ายของไทมด์ออโตมาตา ปัจจุบันเป็นเวอร์ชัน 4.1.24 ปี ค.ศ.2019 เครื่องมือ UPPAAL ประกอบด้วย 2 ส่วนหลัก ได้แก่ Graphical User Interface (GUI) และ Model-Checker Engine

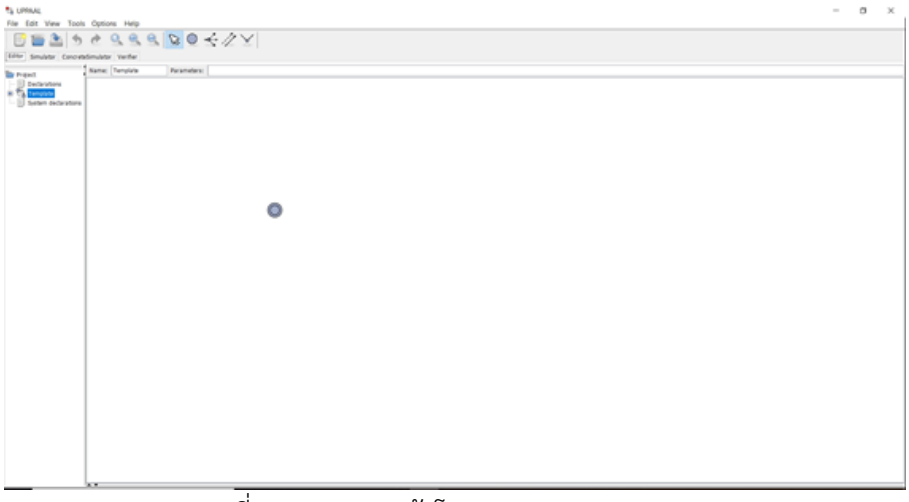
ในส่วนของ Model-Checker Engine จะประกอบด้วย 3 ส่วนหลัก ได้แก่

1. ส่วนภาษาคำอธิบาย (Description Language) เป็นการอธิบายเพิ่มเติมในส่วนของรูปแบบตัวแปร เช่น int, char, double, array เพื่ออธิบายพฤติกรรมของไทมด์ออโตมาตา

2. ส่วนการจำลอง (Simulation) เป็นการทวนสอบรูปแบบประมวลผลแบบไดนามิกระหว่างการออกแบบจำลอง

3. ส่วนการทวนสอบแบบจำลอง (Model Checker) เป็นการทวนสอบค่าคงที่, ภาวะติดตาย, การเข้าถึงในแบบจำลองของปริภูมิสถานะ (State Space)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

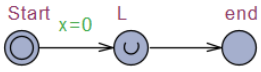
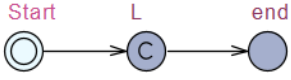


รูปที่ 2-10 แสดงหน้าโปรแกรม UPPAAL [15]

ในเครื่องมือ UPPAAL ไทมด์ออโตมาตาแสดงในรูปแบบกราฟมีทิศทาง (Directed Graph) ดังรูปที่ 2-11 ข้างล่าง ซึ่งมีโลเคชัน (Location) เป็น โหนด (Node) และเส้นเชื่อม (Edge) จะอยู่ระหว่างโลเคชัน

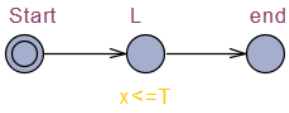
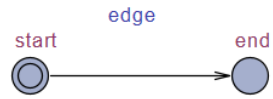
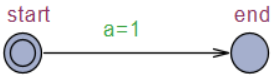
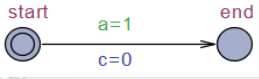
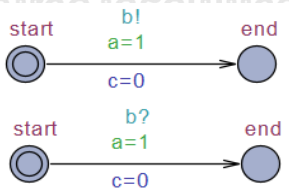
รูปที่ 2-11 แบบจำลองไทมด์ออโตมาตา [15]

ตารางที่ 2-9 แสดงสัญลักษณ์ไทมด์ออโตมาตาในเครื่องมือ UPPAAL [15]

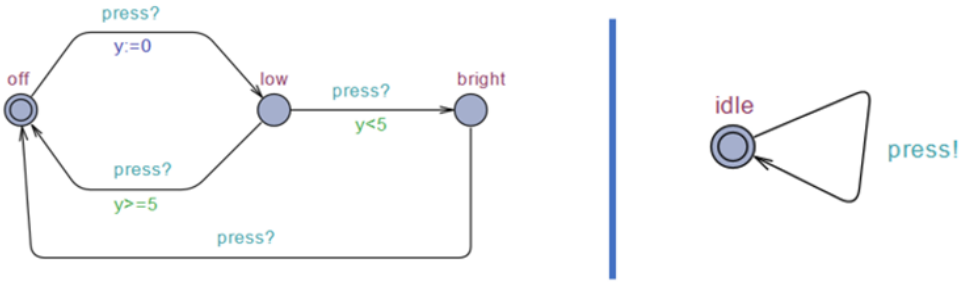
ชื่อ	สัญลักษณ์	คำอธิบาย
1.Location		โลเคชันลักษณะเป็นโหนดมี 4 ชนิดได้แก่ Initial, Urgent, Committed, Normal
1.1 Initial Location		แต่ละเทมเพลต (Template) จะต้องมี Initial ดังนั้นเทมเพลตแต่ละเทมเพลตจะต้องมี 1 โลเคชันที่เป็น Initial
1.2 Normal Location		จากโลเคชันหนึ่งไปยังโลเคชันหนึ่งแบบปกติและในโลเคชันสามารถมีหรือไม่มีความล่าช้า (Delay), การรอ (Wait) ได้
1.3 Urgent Location		นาฬิกา x ที่เข้ามาจะถูกรีเซ็ตบนเส้นเชื่อมที่เข้ามาทั้งหมดและเวลาจะไม่มีการนับภายใน Urgent Location หลังจากเกิด Urgent Location จะสามารถผ่านได้ปกติ
1.4 Committed Location		มีข้อจำกัดมากขึ้นกว่า Urgent Location ใน Committed Location เมื่อนาฬิกาผ่านเข้ามาและส่งออกไป ค่านาฬิกา x=0 เสมอและไม่สามารถมีความล่าช้าและการรอได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 2-9 แสดงสัญลักษณ์ไทมด์ออโตมาตาในเครื่องมือ UPPAAL (ต่อ)

ชื่อ	สัญลักษณ์	คำอธิบาย
2. Invariant		ค่าคงที่จะเป็นเงื่อนไขของโลเคชัน เช่น x<T หรือ x<=T เมื่อ x เป็นนาฬิกาและ T เป็นจำนวนเต็ม
3. Edges		เส้นเชื่อมจะอยู่ระหว่างโลเคชัน 2 โลเคชัน ซึ่งเส้นเชื่อมใช้อธิบาย การ์ด, อัปเดต, การซิงโครไนซ์
3.1 Guard		การ์ดเป็นการแสดงตัวแปรและนาฬิกาของแบบจำลองเพื่อระบุว่าเมื่อใดเปิดใช้งาน (Enabled) เช่น ตัวแปร a = 1
3.2 Update		เป็นการอัปเดตค่าเพื่อที่จะเป็นการยิง (Fired) ไปอีกสถานะนึง การอัปเดตทำก็ต่อเมื่อเปลี่ยนสถานะระบบเช่น ตัวแปร a = 1 จะมีการอัปเดตตัวแปร c = 0
3.3 Synchronisation		การประกาศตัวแปร : Chan b บนเส้นเชื่อมจะกำกับด้วย b! ซึ่งจะซิงโครไนซ์ไปยังเส้นเชื่อมที่กำกับด้วย b?

จากรูปที่ 2-13 ข้างล่างแสดงตัวอย่างแบบจำลองไทมด์ออโตมาตอนของโคมไฟในเครื่องมือ UPPAAL รูปที่ (1) ด้านซ้ายแสดงโคมไฟมี 3 โลเคชัน ได้แก่ off, low และ bright และ รูปที่ (2) ด้านขวาแสดงโลเคชัน 1 โลเคชัน ซึ่งแสดงผู้ใช้กดปุ่มโคมไฟ มีการทำงานแบบซิงโครไนซ์กันของแบบจำลองทั้ง 2 ออโตมาตอนผ่านทาง press เพื่อส่งสัญญาณการทำงานระหว่างระบบย่อยของแบบจำลองเดียวกัน ดังรูปที่ 2-13



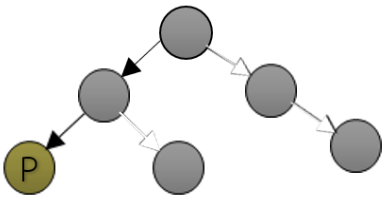
(1) โคมไฟ (2) ผู้ใช้

รูปที่ 2-12 ตัวอย่างการทำงานประสานกันของแบบจำลองโคมไฟ 2 ออโตมาตอน [15]

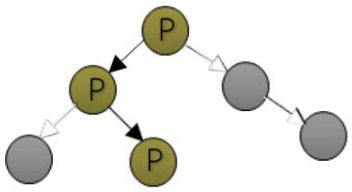
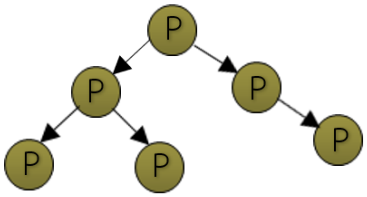
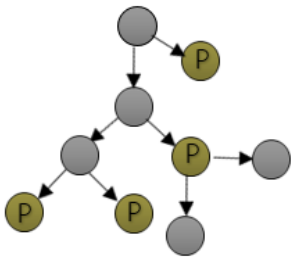
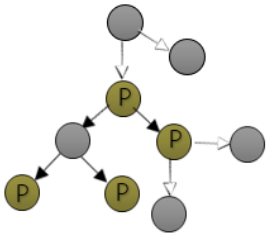
ในการทวนสอบคุณสมบัติของเครื่องมือ UPPAAL จะจัดอยู่ในรูปแบบภาษาสืบค้น (Query) ซึ่งเป็นสับเซต TCTL (Timed Computation Tree Logic) โดยมีการนิยามสัญลักษณ์ดังนี้

- E – มีเส้นทาง (Exists a Path) (“**E**” ในเครื่องมือ UPPAAL)
- A - สำหรับทุกเส้นทาง (All Paths) (“**A**” ในเครื่องมือ UPPAAL)
- G - สำหรับทุกสถานะในเส้นทาง (All States in a Path) (“**[]**” ในเครื่องมือ UPPAAL)
- F – บางสถานะในเส้นทาง (Some State in a Path) (“**<>**” ในเครื่องมือ UPPAAL)

ตารางที่ 2-10 การอธิบายคุณสมบัติการทวนสอบของ TCTL [12]

คุณสมบัติการสืบค้น	คำอธิบาย
1. คุณสมบัติการเข้าถึง (Reachability Properties) 	คุณสมบัติการเข้าถึง สนใจเงื่อนไขที่มีบางสถานะของพฤติกรรมที่เป็นไปได้ในแบบจำลอง ตัวอย่าง ระบบนั้นมีเส้นทางอย่างน้อย 1 เส้นทางที่ไปยังบางสถานะ p ซึ่งจะตรงกับคุณสมบัติการทวนสอบ ตัวอย่าง การเขียน TCTL : E<> p

ตารางที่ 2-10 การอธิบายคุณสมบัติการทวนสอบของ TCTL [12] (ต่อ)

คุณสมบัติการสืบค้น	คำอธิบาย
2. คุณสมบัติความปลอดภัย (Safety Properties) ตัวอย่าง (1)  ตัวอย่าง(2) 	สนใจเงื่อนไขที่มีสถานะทั้งหมดของเส้นทางที่การปฏิบัติตาม (Execution) ตัวอย่าง (1) ระบบมีเส้นทางอย่างน้อย 1 เส้นทางสำหรับทุกสถานะ p ตัวอย่างการเขียน TCTL : E [] p ตัวอย่าง (2) สำหรับระบบทุก ๆ เส้นทาง และสำหรับทุกสถานะ p ในเส้นทาง ตัวอย่างการเขียน TCTL : A [] p
3. คุณสมบัติการคงอยู่ (Liveness Properties) ตัวอย่าง (1)  ตัวอย่าง (2) 	คุณสมบัติการคงอยู่ ระบบที่สนใจนั้นในทุก ๆ เส้นทางที่ท้ายที่สุดแล้วจะไปสู่สถานะที่ตรงกับคุณสมบัติที่ต้องการทวนสอบ ตัวอย่าง (1) ระบบทุก ๆ เส้นทาง ที่มีบางสถานะ p ตัวอย่างการเขียน TCTL : A <> p ตัวอย่าง (2) เส้นทางใดก็ได้ สถานะเริ่มต้นที่ q มายังสถานะ p ตัวอย่างการเขียน TCTL : q-->p

ตารางที่ 2-10 การอธิบายคุณสมบัติการทวนสอบของ TCTL [12] (ต่อ)

คุณสมบัติการสืบค้น	คำอธิบาย
4. คุณสมบัติภาวะติดตาย (Deadlock Properties)	คำอธิบาย จะทวนสอบการหยุดที่เป็นไปได้ของแบบจำลอง ภาวะติดตาย ในแบบจำลองจะมีการรอและทรานซิชั่นระหว่างโลเคชัน เช่น ไม่มีการเปิดใช้งานในทรานซิชั่นนั้น ตัวอย่าง การเขียน TCTL : – E<> deadlock หมายถึง มีเส้นทางเป็นภาวะติดตาย – A[] not deadlock หมายถึง สำหรับทุกเส้นทางไม่มีภาวะติดตาย

ตัวอย่างที่ 3.1 การกำหนดตัวแปรในเครื่องมือ UPPAAL

- นาฬิกา

  สัญลักษณ์ clock x1, ..., xn ; เช่น : clock x, y; การกำหนดนาฬิกา 2 ตัวเป็นค่า x, y

- ตัวแปรข้อมูล (Data Variable)

  สัญลักษณ์ : int n1, ... ; เช่น int a, b;

  int[l,u] n1, ... ; เช่น int[0,1]a;

  int n1[m], ... ; เช่น b[5];

- แชนเนลหรือการกระทำ

  สัญลักษณ์ : chan a, ... ; เช่น chan a, b[2];

  urgent chan b, ... ; เช่น urgent chan c;

- ค่าคงที่ (Constant)

  สัญลักษณ์ : const int c1 = n1; เช่น const int[0,1] YES = 1;

  const bool NO = false;

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 3
วิธีการวิจัยของการสร้างแบบจำลองแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

ในหัวข้อนี้พูดถึงวิธีการสร้างแบบจำลองแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟ ผู้วิจัยจะกล่าวถึงขั้นตอนการดำเนินงานของงานวิจัยที่อธิบายดังรูปที่ 3-1 ในรูปภาพข้างล่างเป็นขั้นตอนแรก จะอธิบายกฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟด้วยแผนภาพเวลา ขั้นตอนที่สองเป็นการนำเข้าแผนภาพเวลาที่เป็นอิสระต่อกันอยู่ในรูปแบบไฟล์เอกซ์เอ็มแอล และในไฟล์เอกซ์เอ็มแอล จะบอกถึงลำดับความสำคัญในแต่ละแผนภาพนั้นด้วย ขั้นตอนที่สามเป็นการนำแผนภาพเวลามาแปลงเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟโดยอ้างอิงจากกฎที่เราออกแบบไว้ และในขั้นตอนสุดท้าย ผลการแปลงในรูปแบบไทมด์ออโตมาตาที่ได้จากกฎ นำมาเข้าเครื่องมือที่ชื่อว่า UPPAAL เพื่อนำไทมด์ออโตมาตานั้นไปทวนสอบลำดับต่อไป

รูปที่ 3-1 ภาพรวมของการทำวิทยานิพนธ์

3.1 การออกแบบและสร้างกฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟ

การนำเข้าไฟล์ต้นทางแผนภาพเวลา มีคุณสมบัติว่าแผนภาพเวลาสามารถมีหลายแผนภาพเวลาซึ่งเป็นอิสระต่อกันและจะมีลำดับความสำคัญกำกับแต่ละแผนภาพเวลา ผู้วิจัยนิยามกฎของแผนภาพเวลา

นิยาม แผนภาพเวลาที่เป็นอิสระต่อกัน มีค่าลำดับความสำคัญกำกับ

แผนภาพเวลาที่เป็นอิสระต่อกันและมีค่าลำดับความสำคัญกำกับ (Timing Diagram of an Individual Task with Priority Value (TDTP)) จะสามารถเขียนสมการได้ดังนี้

$$TDTP = (S, t, F, S_0, p) \quad (7)$$

เมื่อ

S เป็นเซตสถานะจำกัด (Finite Set of States) ที่เป็นอิสระต่อกัน

t เป็นเวลาที่เฉพาะเจาะจง (Specific Time) โดย $t \in [0, \infty)$

$F: t \rightarrow S$ สถานะฟังก์ชันของแผนภาพเวลาที่วนกลับมาในเวลาที่กำหนด

S_0 เป็นสถานะเริ่มต้นของแผนภาพเวลา (Initial State) โดย $S_0 \in S$

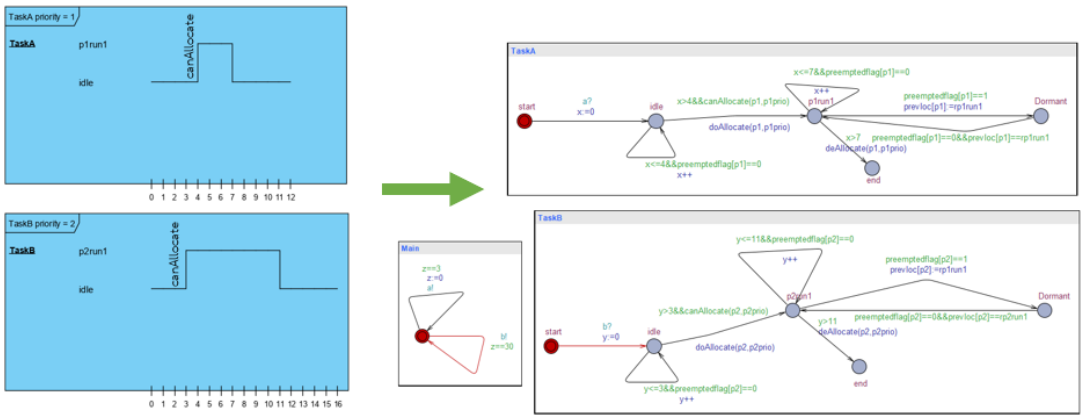
P เป็นค่าลำดับความสำคัญของแต่ละแผนภาพเวลา

วิทยานิพนธ์เสนอกฎการแปลงแต่ละองค์ประกอบของแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟ ต่อไปนี้ผู้วิจัยจะมาสร้างกฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟมีทั้งหมด 7 กฎ โดยมีรายละเอียดกฎดังนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

กฎข้อที่ 1 แผนภาพเวลาที่เป็นอิสระต่อกันและมีค่าลำดับความสำคัญ จะจับคู่กับไทมด์ออโตมาตา

แผนภาพเวลาแต่ละแผนภาพเวลาจะบ่งบอกลำดับความสำคัญแต่ละแผนภาพเวลา โดยแผนภาพเวลาต้องเป็นอิสระต่อกันและมีค่าลำดับความสำคัญกำกับ จะแปลงไปเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟได้ 3 แบบ ตัวอย่างเช่น $TDTP_{TaskA}$ และ $TDTP_{TaskB}$ จะแปลงเป็น TA_{TaskA} ,TA_{TaskB} และ $Main$ ซึ่งแสดงในรูป 3-2

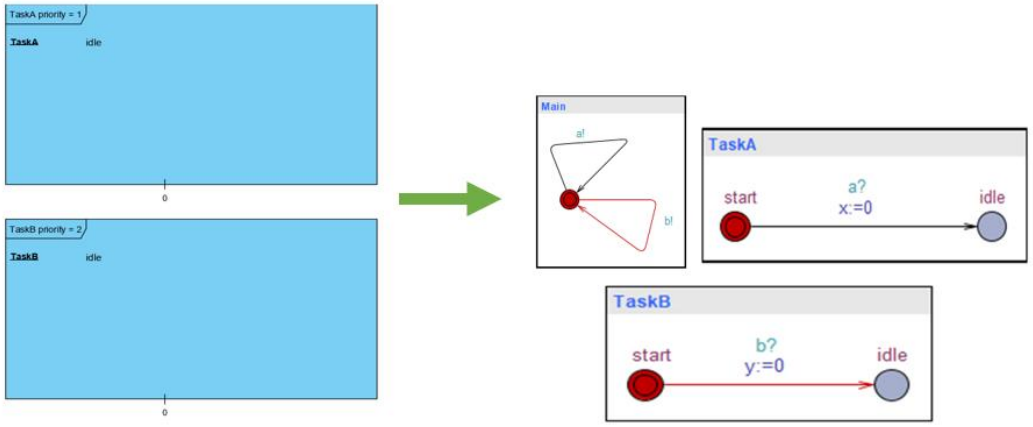


รูปที่ 3-2 แผนภาพเวลา Task A และ Task B จะจับคู่ออโตมาตอนได้ 3 แบบได้แก่ Task A, Task B และ Main

<u>กฎข้อที่ 2</u> สถานะเริ่มต้นของแผนภาพเวลาจะแปลงเป็นโหนดเริ่มต้นของไทม์ออโตมาตา

สถานะเริ่มต้น (S_0) ใน $TDTP$ จะเป็นการแปลงโหนดเริ่มต้น (Start Node) หรือเรียกอีกอย่างหนึ่งว่า โลเคชันเริ่มต้น จะอยู่ในรูปแบบของ $l_0 \in L$ และเส้น (Edge) จะอยู่ในรูปแบบของ $e \in E$ เป็นการเชื่อมโยงของโหนดโลเคชันซึ่งอยู่ในรูปแบบของ $l \in L$ (หลังสถานะเริ่มต้นในแผนภาพเวลา $TDTP$) ตัวอย่างเช่น สถานะเริ่มต้นของ $TDTP_{TaskA}$ จะเรียกว่า สถานะ $"idle"$ จะแปลงเป็นโหนดเริ่มต้น $"start"$ และเส้นสัญลักษณ์จะเขียนได้เป็น $e = ("start", \{"x := 0"\}, True, \{\}, "idle") \in E$ เชื่อมต่อโหนดและมี การกระทำ $"x := 0" \in Act$ เป็นค่ารีเซ็ตของตัวแปรที่กำกับไว้บนเส้น e ซึ่งแสดงในรูป 3-3

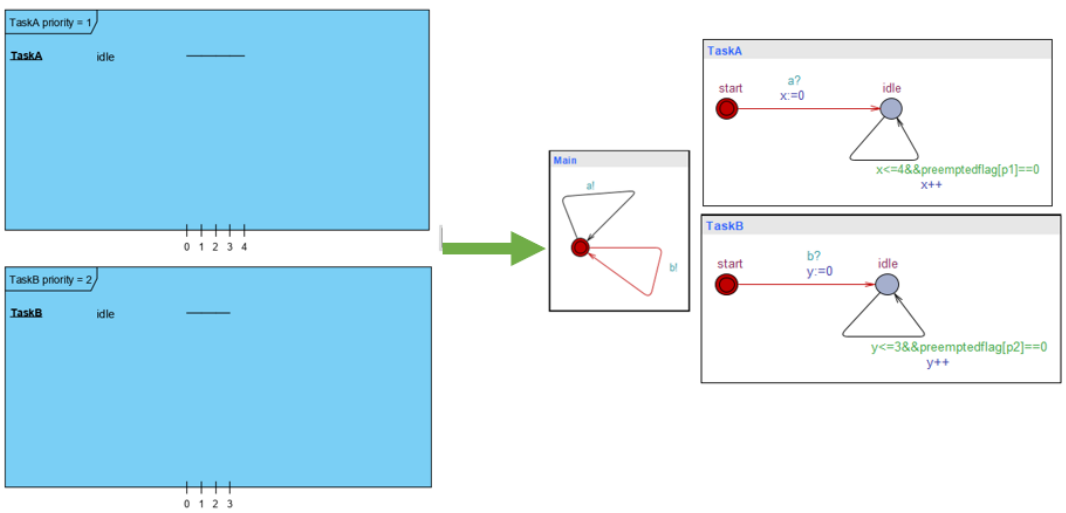
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



รูปที่ 3-3 การจับคู่สถานะเริ่มต้นของแผนภาพเวลา TDTP

<u>กฎข้อที่ 3</u> เส้นช่วงเวลา ณ สถานะเริ่มต้นในแผนภาพเวลาที่มีลำดับความสำคัญจะจับคู่กับไทมด์ออโตมาตาจะมีเส้นเชื่อมที่สถานะ “idle” ที่วนมายังสถานะตัวมันเอง

เส้นช่วงเวลา (Time Period (k)) ณ สถานะเริ่มต้น (S_0) ใน $TDTP$ จะจับคู่กับไทมด์ออโตมาตาจะมีเส้น e “$idle$” ที่วนมายังสถานะตัวมันเอง (วนไปยังโหนด “$idle$”) ที่เส้น $e \in E$ เส้นเชื่อมต่อจากโหนดและวนมาหาตัวเองจะเขียนเงื่อนไขการ์ด (Guard) เป็น $k(\leq k)$ ตัวอย่างเช่น ณ สถานะเริ่มต้น ใน $TDTP_{TaskA}$ จะเรียกโหนด “$idle$” ช่วงเวลาเงื่อนไขการ์ดจะคิดเป็น $k = t_2 - t_1$ ถ้า $F(t_1) \rightarrow S_0$ และ $F(t_2) \rightarrow S_i$ ที่ $i \geq 0$ ตัวอย่างเช่น สถานะ “$idle$” ช่วงเวลา $k = 4$ และสัญลักษณ์เส้นจะเป็น $e = (\text{"}idle\text{"}, \{\}, \text{"}\ x \leq 4\text{"}, \{\}, \text{"}idle\text{"}) \in E$ เชื่อมต่อกับโหนด “$idle$” ที่เข้าหาตัวเอง การ์ดจะเท่ากับ $\text{"}x \leq 4\text{"} \in B(C)$ โดย $B(C)$ เป็นการ์ดที่แสดงบนเส้น e ซึ่งแสดงในรูปที่ 3-4 นั่นคือเป็นการจับคู่ช่วงเวลาของสถานะ “$idle$” ของแผนภาพเวลา $TDTP$

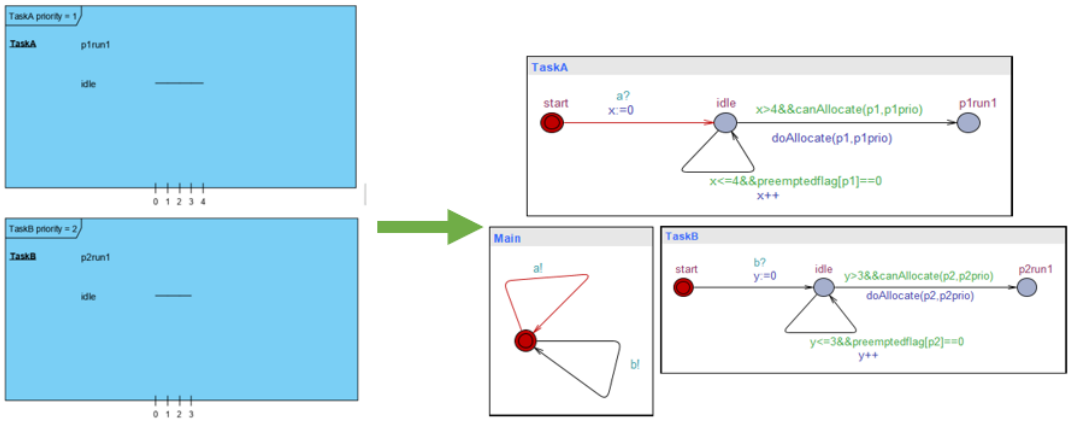


รูปที่ 3-4 การจับคู่ช่วงเวลาของสถานะเริ่มต้นของแผนภาพเวลา TDTP

<u>กฎข้อที่ 4</u> เมื่อมีหลายสถานะต่อไปจะเขียนสัญลักษณ์ S_i และถ้ามีงานดำเนินการพร้อมกันจะเกิดการขัดจังหวะ

เมื่อมีหลายสถานะต่อไปจะเขียนสัญลักษณ์ S_i จากสถานะเริ่มต้น S_0 ใน $TDTP$ จะจับคู่กับไทมด์ออโตมาตา เส้น $e \in E$ ที่เชื่อมต่อกับสถานะเริ่มต้นไปยังโหนด I (จะเรียกสถานะ S_i) และเงื่อนไขการ์ดจะมากกว่าหรือเท่ากับช่วงเวลาสถานะเริ่มต้น $k(> k)$ ดังนั้นจะเขียนด้วยฟังก์ชันได้เป็น $canAllocate(self\ , mypriority)$ จะเพิ่มการ์ดบนเส้น e เพื่อตรวจสอบว่า TA อนุญาตเข้าไปประมวลผลการคำนวณ (Computing Processor) หรือไม่และประกาศชื่อและค่าลำดับ

ความสำคัญให้งานดำเนินไปตามเวลา ถ้างานที่ดำเนินการตามเวลามีลำดับความสำคัญต่ำเข้ามาพร้อมกับงานที่ดำเนินการตามเวลามีลำดับความสำคัญสูง งานที่ดำเนินการตามเวลามีลำดับความสำคัญต่ำใน TA จะเกิดการขัดจังหวะ (Interrupt) และจะเข้าสู่โหมดการทำพรีเอ็มทีฟและพองานที่มีค่าความลำดับความสำคัญที่มีค่าสูงดำเนินงานเสร็จสิ้น ทำให้งานที่อยู่ในโหมดพรีเอ็มทีฟกลับไปทำงานเสร็จสิ้น ตัวอย่างเช่น สถานะเริ่มต้นของ TA_{TaskA} ที่เรียกว่า $“idle”$ ช่วงเวลา $k = 4$ และเขียนสัญลักษณ์เส้นเป็นรูปแบบดังนี้ $e = (“idle”, \{\}, “x > 4 \&\& canAllocate(sel, mypriority)”, \{\}, \text{"}run\text{"}) \in E$ ที่เชื่อมต่อโหนด $“idle”$ ไปยังโหนด $“run”$ การ์ด "$x > 4$" $B(C)$ เป็นกำกับด้วยเส้นบนเส้น e ถึงแสดงดังรูป 7 ฟังก์ชัน $canAllocate(TA_{TaskA}, 1)$ เป็นส่วนที่เพิ่มไว้ในการ์ด ซึ่งแสดงในรูปที่ 3-5 เป็นการจับคู่สถานะถัดไปของสถานะเริ่มต้นแผนภาพเวลา $TDTP$ นั่นคือ p1run1 และ p2run2

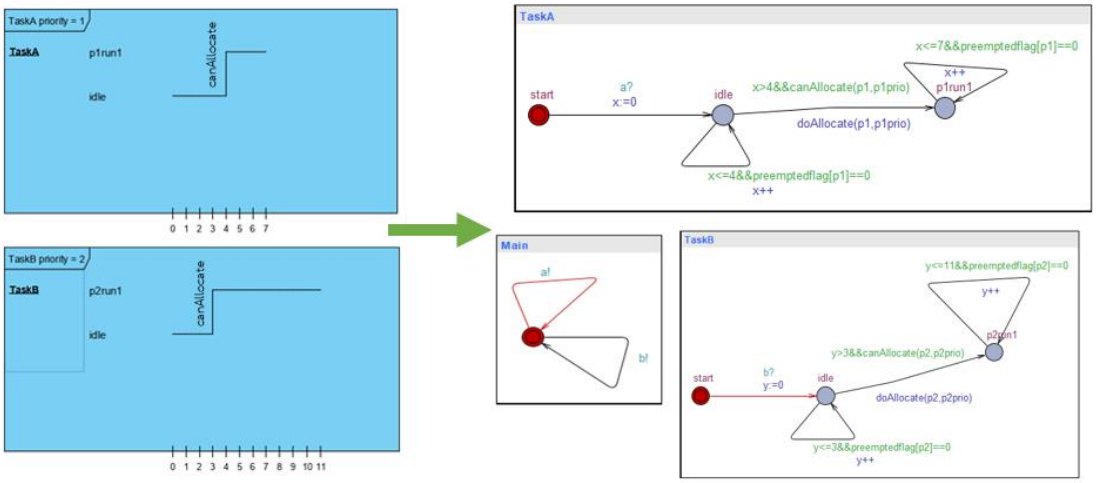


รูปที่ 3-5 การจับคู่สถานะถัดไปของสถานะเริ่มต้นแผนภาพเวลา TDTP

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

<u>กฎข้อที่ 5</u> ช่วงเวลาของสถานะ S_i ใน TDTP จับคู่กับไทม์ออโตมาตา เส้นเชื่อมไปหาโหนดตัวเอง

ช่วงเวลา k ของสถานะ S_i ใน *TDTP* จับคู่กับไทม์ออโตมาตา เส้นเชื่อมไปหาโหนดตัวเอง $e \in E$ และเงื่อนไขพิเศษ $k \leq k$ ตัวอย่างเช่น สถานะ "run" ของ $TDTP_{TaskA}$ ช่วงเวลาสถานะจะคิดเป็น $k = t_2 - t_1$ ถ้า $F(t_1) \rightarrow S_0$ และ $F(t_2) \rightarrow S_j$ ที่ $j \geq j$ สถานะ "run" ช่วงเวลา $k = 7$ และสัญลักษณ์บนเส้น $e = (“run”, \{\}, “x \leq 7”, \{\}, \text{"}run\text{"}) \in E$ เส้นเชื่อมต่อกับโหนดตัวมันเองนั่นคือ โหนด "run" การ์ด "$x \leq 7$" $B(C)$ เป็นกำกับด้วยเส้นบนเส้น e ถึงแสดงดังรูป 3-6



รูปที่ 3-6 จับคู่ช่วงเวลาของสถานะ p1run1 และ p2run1 แผนภาพเวลา TDTP

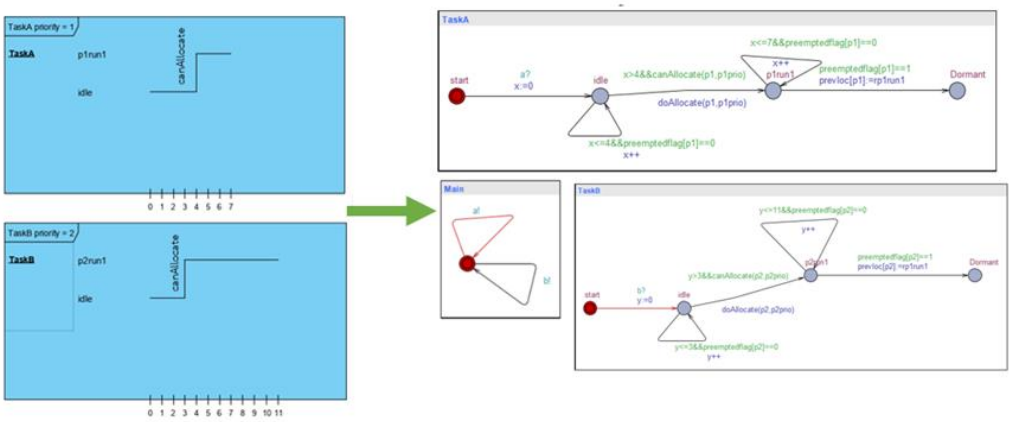
<u>กฎข้อที่ 6</u> การเพิ่มสถานะโดเมนในไทม์ออโตมาตา

ในผลของ TA โลเคชันโดเมน (Dormant Location (dL)) จะมีการเพิ่มเมื่อจัดกำหนดการเชิงพรีเอ็มทีฟ แต่ละโหนด (l) ที่แอคทีฟ $L - \{l_0, "idle"\}$ เพิ่มเส้นขาออก $e_1 \in E$ เชื่อมต่อกับโหนด (l) ไปยังโหนดโดเมน (d) อยู่บนเส้น การ์ดจะเป็นฟังก์ชัน $ispreempted(sel\ f\ , l)$ เปิดใช้งานเมื่อ TA เกิดการขัดจังหวะ และถูกบล็อก ดังนั้น เพิ่มเส้นเข้ามา $e_2 \in E$ เชื่อมต่อกับเส้นโหนดโดเมนกลับมายังโหนด l เช่นเดิม บนเส้นจะถูกกำกับด้วยการ์ดเป็นฟังก์ชัน $(not\ ispreempted(sel\ f\ , l))$ ซึ่งเปิดเมื่อ TA ไม่เกิดการขัดจังหวะและไม่ถูกบล็อก บนเส้นที่มีการ์ดเขียนด้วยฟังก์ชัน $ispreempted(self\ , l)$ จะเช็คเมื่อ TA เกิดการทำพรีเอ็มทีฟหรือไม่ ตัวอย่างเช่น โหนดสถานะโดเมน $"dormant" \in L$ เพิ่มผลลัพธ์ของไทม์ออโตมาตา TA_{TaskA} โหนดที่มีการแอคทีฟไปยังสถานะ $"run"$ พิจารณาทั้ง 2 เส้น ได้แก่

$e_1 = ("run", \{\}, "\ ispreempted(TA_{TaskA}, "run")", \{\}, "dormant") \in E$ และเส้น

$e_2 = ("dormant", \{\}, "\ not\ ispreempted(TA_{TaskA}, "run")", \{\}, "run") \in E$ แสดงในรูปที่ 3-7 และ 3-8 อธิบายนั่นคือ 3-7 ถ้า TaskA มีลำดับความสำคัญเท่ากับ 1 และ TaskB มีลำดับความสำคัญเท่ากับ 2 ถ้า Task A ไปยังสถานะ “p1run1” อยู่แล้ว TaskB

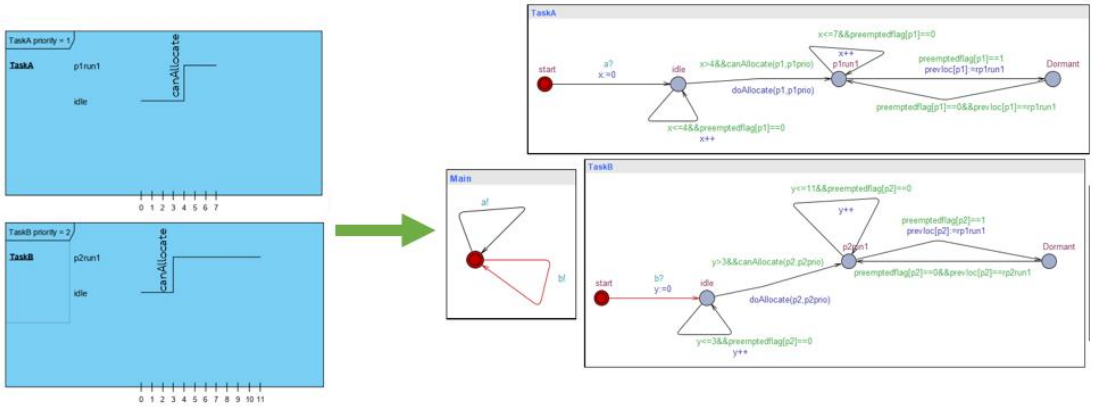
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 3-7 เพิ่มสถานะโดเมนในแผนภาพเวลา TDTP เมื่อมีลำดับความสำคัญสูงเข้าแทรก

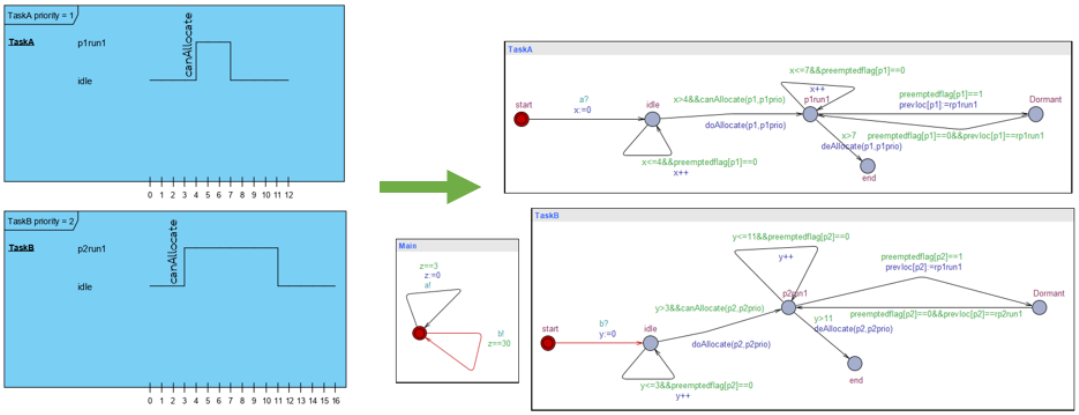
<u>กฎข้อที่ 7</u> เส้นสิ้นสุดของแผนภาพเวลาจับคู่กับโหนด end

เพิ่มโหนดสถานะ n (End Node) ไปยังผลลัพธ์ไทม์ออโตมาตาถ้าเส้นสิ้นสุดจาก TDTP สำหรับโหนดล่าสุด (Last Node (IL)) จะเพิ่มเส้นออก $e \in E$ เชื่อมกับโหนด l ไปยังโหนด end บนเส้นกำกับจะกำกับการ์ดด้วย $deallocate(self, l)$ ซึ่งกู้คืนเธรดที่ถูกทำพรีเอ็มทีฟไว้ก่อนหน้ากลับไปยังประมวลผลและรีเซ็ตสถานะค่าพรีเอ็มทีฟ ตัวอย่างเช่น โหนดสุดท้าย a $"end" \in L$ เพิ่มผลลัพธ์ไทม์ออโตมาตา TA_{TaskA} รวมถึงพิจารณาโหนดสถานะล่าสุด $"run"$ และบนเส้น $e = ("run", \{\}, x > 7 \&\& deAllocate(TA_{TaskA}, "run")", \{\}, "end") \in E$ จะแสดงในรูป 3-9



รูปที่ 3-8 เพิ่มสถานะโดเมนในแผนภาพเวลา TDTP กลับมาสถานะ “run”

ผลที่ได้ตามกฎการแปลงที่กำหนดไว้ข้างบน จากแผนภาพเวลา 2 รูป $TDTP_{TaskA}$ และ $TDTP_{TaskB}$ จะมีค่าลำดับความสำคัญกำกับเราจะได้ไทมด์ออโตมาตา TA_{TaskA} และ TA_{TaskB} ตามรูปด้านล่าง



รูปที่ 3-9 ผลลัพธ์ของไทมด์ออโตมาตา Task A และ Task B จากข้อมูลนำเข้าของแผนภาพเวลา

3.2 อธิบายการแปลงแผนภาพเวลาและไทมด์ออโตมาตาในรูปแบบของโครงสร้างเอกซ์เอ็มแอล

3.2.1 การแปลงแผนภาพเวลาให้อยู่ในรูปแบบของไฟล์เอกซ์เอ็มแอล

เพื่อให้การแปลงแผนภาพเวลาให้อยู่ในรูปแบบของสกุลเอกซ์เอ็มแอลเป็นไปตามแนววิธีการวิทยานิพนธ์นี้ รูปแบบของไฟล์สกุลเอกซ์เอ็มแอลจะต้องอยู่ในรูปแบบโครงสร้างที่เป็นมาตรฐานรวมไปถึงไฟล์นั้นต้องเข้าสู่เครื่องมือในรูปแบบที่ถูกต้องตามโครงสร้างของแท็กเอกซ์เอ็มแอลที่ผู้วิจัยได้สร้างขึ้น ก่อนที่ผู้วิจัยจะทำการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟได้ แผนภาพเวลาในรูปแบบสกุลเอกซ์เอ็มแอลจะต้องผ่านการสกัดองค์ประกอบออกก่อน โดยองค์ประกอบเหล่านี้จะอยู่ในรูปของแท็กเอกซ์เอ็มแอล ซึ่งจะประกอบไปด้วยแท็กเอกซ์เอ็มแอลดังตารางที่ 3-1 โดยแท็กเหล่านี้จะแบ่งออกเป็นแต่ละประเภทตามตารางที่ 3-1

ตารางที่ 3-1 แท็กเอกซ์เอ็มแอลของแผนภาพเวลาไฟล์สกุลต้นทาง

แท็กเอกซ์เอ็มแอลไฟล์สกุลต้นทาง	ประเภท	คำอธิบายไฟล์สกุลต้นทาง
<TimingDiagramS> ... </TimingDiagramS>	TimingDiagramS	รากของแผนภาพเวลาหลายแผนภาพเวลา
<TimingDiagram name="..." priority="..." syc="..." int="..." clock="...">	TimingDiagram name	ชื่องานแต่ละงานของแผนภาพเวลา เช่น TaskA
	priority	ค่าลำดับความสำคัญถ้ามีเลขน้อยหมายความว่ามีลำดับความสำคัญมาก เป็นตัวเลข เช่น 1
	syc	Synchronize เป็นการคุยกันระ หว่างเส้นแทนตัวอักษร เช่น a
	int	จำนวนเต็ม (Integer) แทนตัวอักษร เช่น x
	clock	ชื่อนาฬิกาแทนตัวอักษร เช่น z
<state id="..." name="..."></state>	id	กำหนดงานเป็นตัวเลข เช่น 1
	name	ชื่อสถานะแผนภาพเวลา
<timesequence> ... </timesequence>	timesequence	รากของลำดับเวลา
<seq no="..." state="..." tickerevent="..."> <duration begin="..." end="..."> </duration> </seq>	seq no	ลำดับสถานะระบุเป็นตัวเลข เช่น 1
	state	ชื่อสถานะ เช่น idle
	tickerevent	เหตุการณ์สำคัญ เช่น can Allocate
	duration begin	ระยะเวลาเริ่มต้น เช่น 0
	end	ระยะเวลาสิ้นสุด เช่น 2

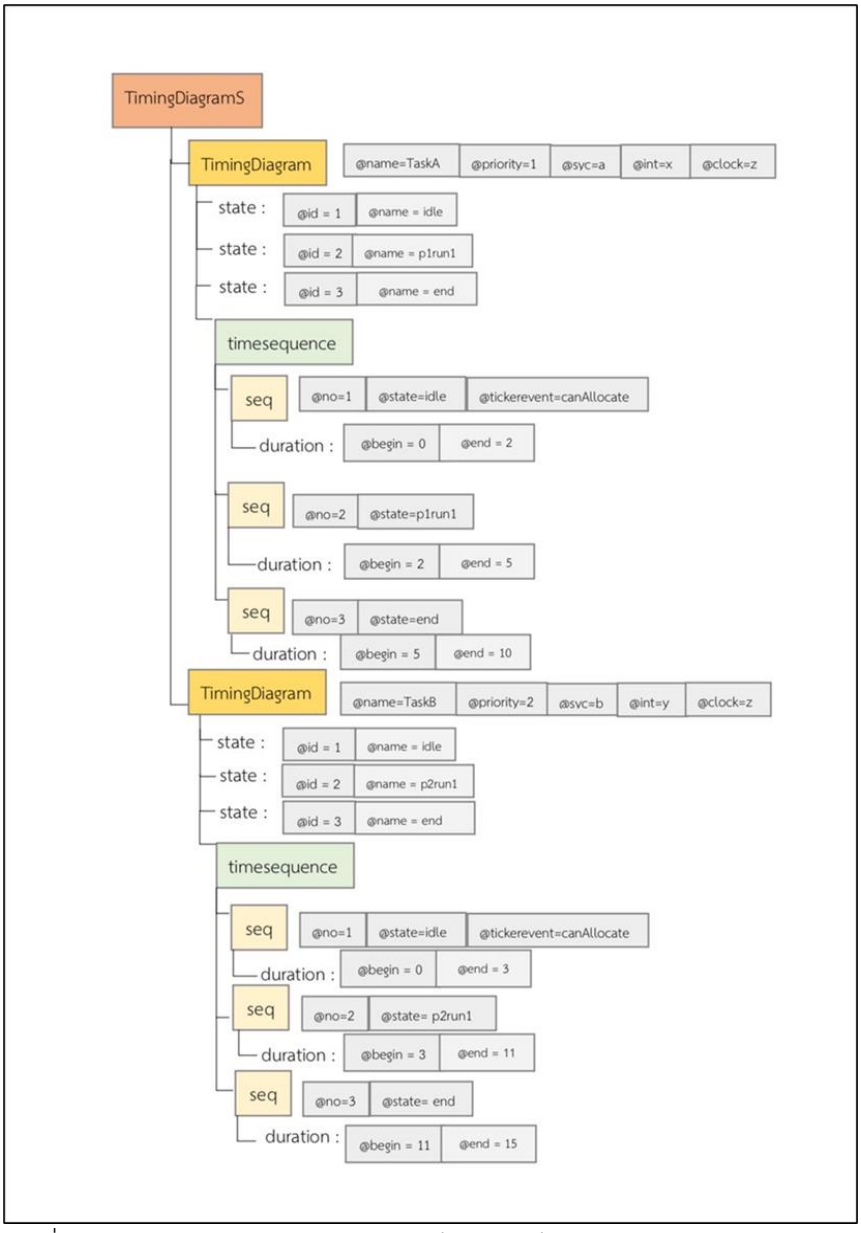
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
<TimingDiagramS>
  <TimingDiagram name="TaskA" priority="1" syc="a" int="x" clock="z">
  <state id="1" name="idle"></state>
  <state id="2" name="p1run1"></state>
  <state id="3" name="end"></state>
  <timesequence>
    <seq no="1" state="idle" tickerevent="canAllocate">
        <duration begin="0" end="4"></duration>
    </seq>
    <seq no="2" state="p1run1">
     <duration begin="4" end="7"></duration>
    </seq>
    <seq no="3" state="end">
     <duration begin="7" end="12"></duration>
    </seq>
  </timesequence>
 </TimingDiagram>
 <TimingDiagram name="TaskB" priority="2" syc="b" int="y" clock="z">
  <state id="1" name="idle"></state>
  <state id="2" name="p2run1"></state>
  <state id="3" name="end"></state>
  <timesequence>
    <seq no="1" state="idle" tickerevent="canAllocate">
        <duration begin="0" end="3"></duration>
    </seq>
    <seq no="2" state="p2run1">
     <duration begin="3" end="11"></duration>
    </seq>
    <seq no="3" state="end">
     <duration begin="11" end="16"></duration>
    </seq>
  </timesequence>
 </TimingDiagram>
</TimingDiagramS>
```

รูปที่ 3-10 ตัวอย่างแท็กเอกซ์เอ็มแอลของแผนภาพเวลา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จะได้ไฟล์สกุลเอกซ์ดังตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของแผนภาพเวลาดังรูปที่ 3-11



รูปที่ 3-11 ตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของแผนภาพเวลา

3.2.2 การแปลงไทมด์ออโตมาตาในรูปแบบของไฟล์เอกซ์เอ็มแอล

ผู้วิจัยได้ใช้โปรแกรม Notepad++ ในการสกัดแต่ละองค์ประกอบของไทมด์ออโตมาตา โดยตารางที่ 3-2 จะแสดงถึงองค์ประกอบของไทมด์ออโตมาตาที่สกัดได้ในไฟล์ต้นทาง

ตารางที่ 3-2 แท็กเอกซ์เอ็มแอลไทมด์ออโตมาตาไฟล์สกุลปลายทาง

แท็กเอกซ์เอ็มแอลของไฟล์สกุลปลายทาง	ประเภท	คำอธิบายของไฟล์สกุลปลายทาง
<nta>...</nta>	nta	Root แท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตา
<declaration>...</declaration>	declaration	แท็กไฟล์สำหรับกำหนดตัวแปรและเขียนโปรแกรมภาษาซี
<template>...</template>	template	ระบุรายละเอียดทั้งหมดภายในออโตมาตอน
<location id="..." x="..." y="..."> <name x="..." y="...">...</name> </location>	location	ระบุรายละเอียดต่าง ๆ ของ Location
	id	ระบุ id ของโลเคชัน
	x	ขนาดแกน x ของเส้น
	y	ขนาดแกน y ของเส้น
	name	ชื่อโลเคชัน
<init ref=""/>	Initial location	ระบุโลเคชัน ที่เป็นจุดเริ่มต้นของออโตมาตอน
<transition>...</transition>	transition	ระบุรายละเอียดของเส้น
<source ref=""/>	source	ระบุโลเคชันก่อนหน้า
<target ref=""/>	target	ระบุโลเคชันถัดไป
<label kind="guard" x="141" y="221">z==0</label>	label kind	ชนิดของ Label ต่าง ๆ เช่น Guard และ Channel
<nail x="173" y="180"/>	nail	ระบุขนาดของเส้น
<label kind="" x="" y="">...</label>	label kind	ระบุรายละเอียดบนเอดจ์ต่าง ๆ เช่น Guard และ Channel
<system>...</system>	system	กำหนดค่าการเรียกออโตมาตอนหรือเทมเพลตเข้าสู่ระบบ Model Checker
<queries/>...<queries/>	queries	สำหรับทวนสอบ TCTL

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ผู้ทำวิจัยได้วิเคราะห์โครงสร้างไฟล์ไทมด์ออโตมาตาในรูปแบบเอกสารเอกซ์เอ็มแอลออกมาเป็นแผนภาพ ดังตารางที่ 3-2 เพื่อจุดประสงค์ในการนำไปใช้ในขั้นตอนการจับคู่ ตามกฎการแปลงที่ระบุไว้ข้างต้น รูปที่ 3-12 ตัวอย่างการอธิบายการเขียนภาษาซีในเครื่องมือ UPPAAL เพื่อทวนสอบการเปรียบเทียบค่าลำดับความสำคัญและเงื่อนไขที่จำเป็นในการเข้าไปใช้ในซีพียู

<declaration>// Place global declarations here.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nta PUBLIC '-//Uppaal Team//DTD Flat System 1.1//EN' 'http://www.it.uu.se/research/group/darts/uppaal/flat-1_2.dtd'>
<declaration>// Place global declarations here.
int x;
int y;
clock z;
chan a;
chan b;
int runningcpu = 0;
int runningpprio =0;
const int p1 =1;
const int p1prio=1;
const int p2 =2;
const int p2prio=2;
int ppreempted[10]; // array
int ppriopreempted[10]; // array
int preemptedflag[10];// preem
int prevloc[10];
const int rp1run1=1;
const int rp2run1=1;
bool canAllocate(int p,int pprio)
{
 if (runningcpu==0 or runningpprio&lt;pprio){
    return true;
 }
 else{
    return false;
 }
}
void doAllocate(int p,int pprio){
 if (runningcpu==0 or runningpprio&lt;pprio){
     preemptedflag[runningcpu]=1;
     ppreempted[p]=runningcpu;
     ppriopreempted[p]=runningpprio;
     runningcpu=p;
     runningpprio=pprio;
 }
}
void deAllocate(int p,int pprio){
 if(ppreempted[p]!=0){
    runningcpu=ppreempted[p];
    runningpprio=ppriopreempted[p];
    preemptedflag[ppreempted[p]]=0;
    ppreempted[p]=0;
    ppriopreempted[p]=0;
 }
}
</declaration>
```

รูปที่ 3-12 ตัวอย่างไทมด์ออโตมาตาในรูปแบบของเอกสารเอกซ์เอ็มแอล

ตารางที่ 3-3 ตัวอย่างไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟในรูปแบบของเอกสารเอกซ์เอ็มแอล

ตัวประกอบไฟล์เอกซ์เอ็มแอล

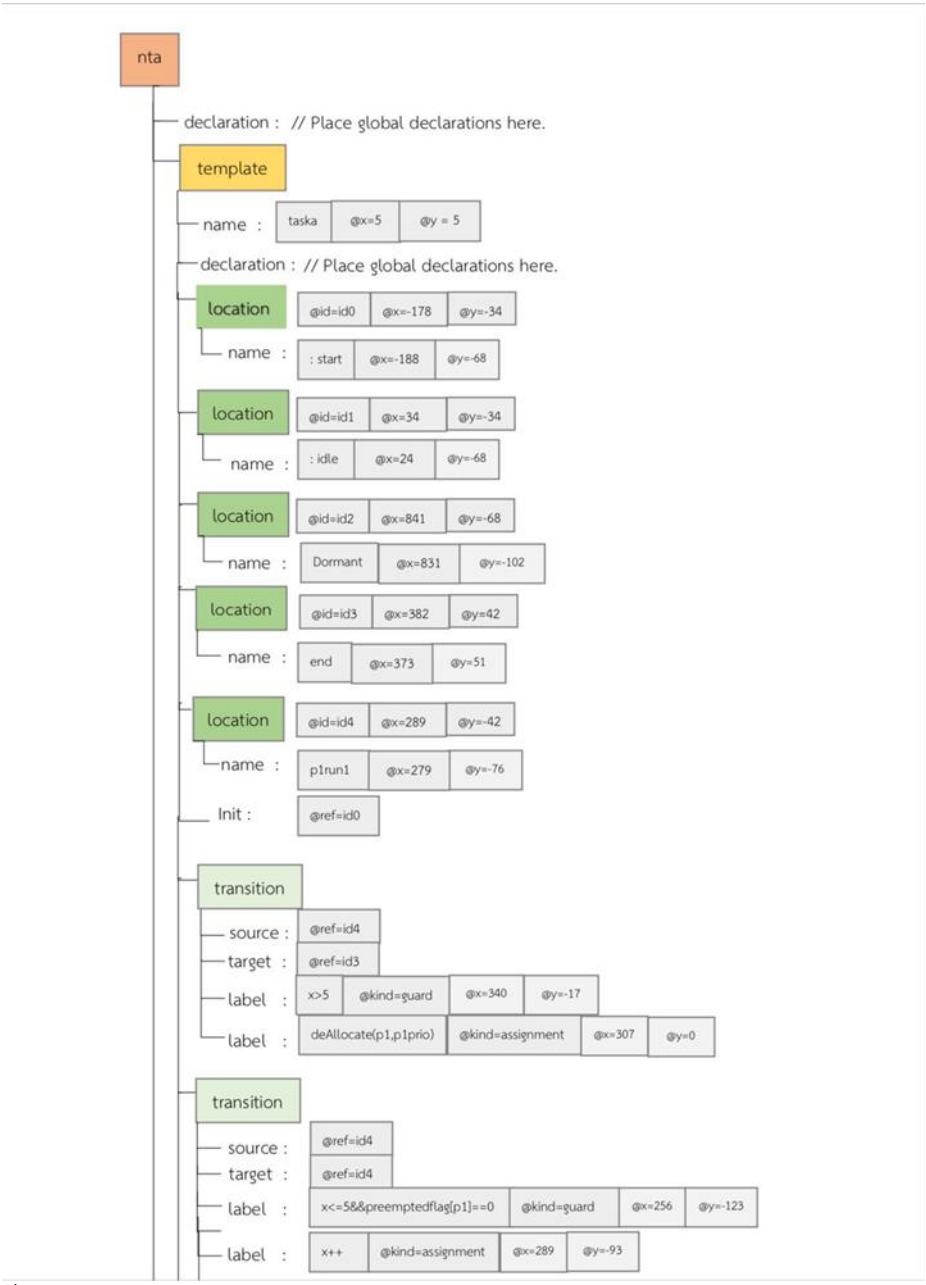
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE nta PUBLIC '-//Uppaal Team//DTD Flat System 1.1//EN' 'http://www.it.uu.se/research/group/darts/uppaal/flat-1_2.dtd'>
<nta>
    <declaration>// Place global declarations here.
******ใส่ code ที่นี่ได้ทำงาน*******
</declaration>
    <template>
        <name x="5" y="5">taska</name>
        <declaration>// Place local declarations here.
</declaration>
        <location id="id0" x="-178" y="-34">
            <name x="-188" y="-68">start</name>
        </location>
        <location id="id1" x="34" y="-34">
            <name x="24" y="-68">idle</name>
        </location>
        <location id="id2" x="841" y="-68">
            <name x="831" y="-102">Dormant</name>
        </location>
        <location id="id3" x="382" y="42">
            <name x="373" y="51">end</name>
        </location>
        <location id="id4" x="289" y="-42">
            <name x="279" y="-76">p1run1</name>
        </location>
        <init ref="id0"/>
        <transition>
            <source ref="id4"/>
            <target ref="id3"/>
            <label kind="guard" x="340" y="-17">x&gt;5</label>
            <label kind="assignment" x="307" y="0">deAllocate(p1,p1prio)</label>
        </transition>
        <transition>
            <source ref="id4"/>
            <target ref="id4"/>
            <label kind="guard" x="256" y="-123">x&lt;=5&amp;&amp;preemptedflag[p1]==0</label>
            <label kind="assignment" x="289" y="-93">x++</label>
            <nail x="238" y="-94"/>
            <nail x="365" y="-85"/>
        </transition>
        <transition>
            <source ref="id2"/>
            <target ref="id4"/>
            <label kind="guard" x="425" y="-17">preemptedflag[p1]==0&amp;&amp;prevloc[p1]==rp1run1</label>
            <nail x="535" y="-17"/>
        </transition>
        <transition>
            <source ref="id4"/>
            <target ref="id2"/>
            <label kind="guard" x="510" y="-102">preemptedflag[p1]==1</label>
            <label kind="assignment" x="527" y="-85">prevloc[p1]:=rp1run1</label>
        </transition>
        <transition>
            <source ref="id1"/>
            <target ref="id4"/>
            <label kind="guard" x="68" y="-76">x&gt;2&amp;&amp;canAllocate(p1,p1prio)</label>
            <label kind="assignment" x="119" y="-34">doAllocate(p1,p1prio)</label>
            <nail x="153" y="-42"/>
            <nail x="195" y="-42"/>
        </transition>
        <transition>
            <source ref="id1"/>
            <target ref="id1"/>
            <label kind="guard" x="17" y="25">x&lt;=2&amp;&amp;preemptedflag[p1]==0</label>
            <label kind="assignment" x="68" y="42">x++</label>
            <nail x="-17" y="25"/>
            <nail x="68" y="25"/>
        </transition>
        <transition>
            <source ref="id0"/>
            <target ref="id1"/>
            <label kind="synchronisation" x="-85" y="-76">a?</label>
            <label kind="assignment" x="-93" y="-59">x:=0</label>
        </transition>
    </template>
    <template>
        <name>taskb</name>
        <location id="id5" x="-365" y="-136">
            <name x="-375" y="-170">start</name>
        </location>
        <location id="id6" x="-195" y="-136">
            <name x="-205" y="-170">idle</name>
        </location>
        <location id="id7" x="612" y="-204">
            <name x="602" y="-238">Dormant</name>
        </location>
        <location id="id8" x="144" y="-102">
            <name x="135" y="-85">end</name>
        </location>
        <location id="id9" x="34" y="-195">
            <name x="24" y="-229">p2run1</name>
        </location>
        <init ref="id5"/>
        <transition>
            <source ref="id9"/>
            <target ref="id8"/>
            <label kind="guard" x="102" y="-187">y&gt;11</label>
            <label kind="assignment" x="85" y="-170">deAllocate(p2,p2prio)</label>
```

ตารางที่ 3-3 ตัวอย่างไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟในรูปแบบของเอกสารเอกซ์เอ็มแอล (ต่อ)

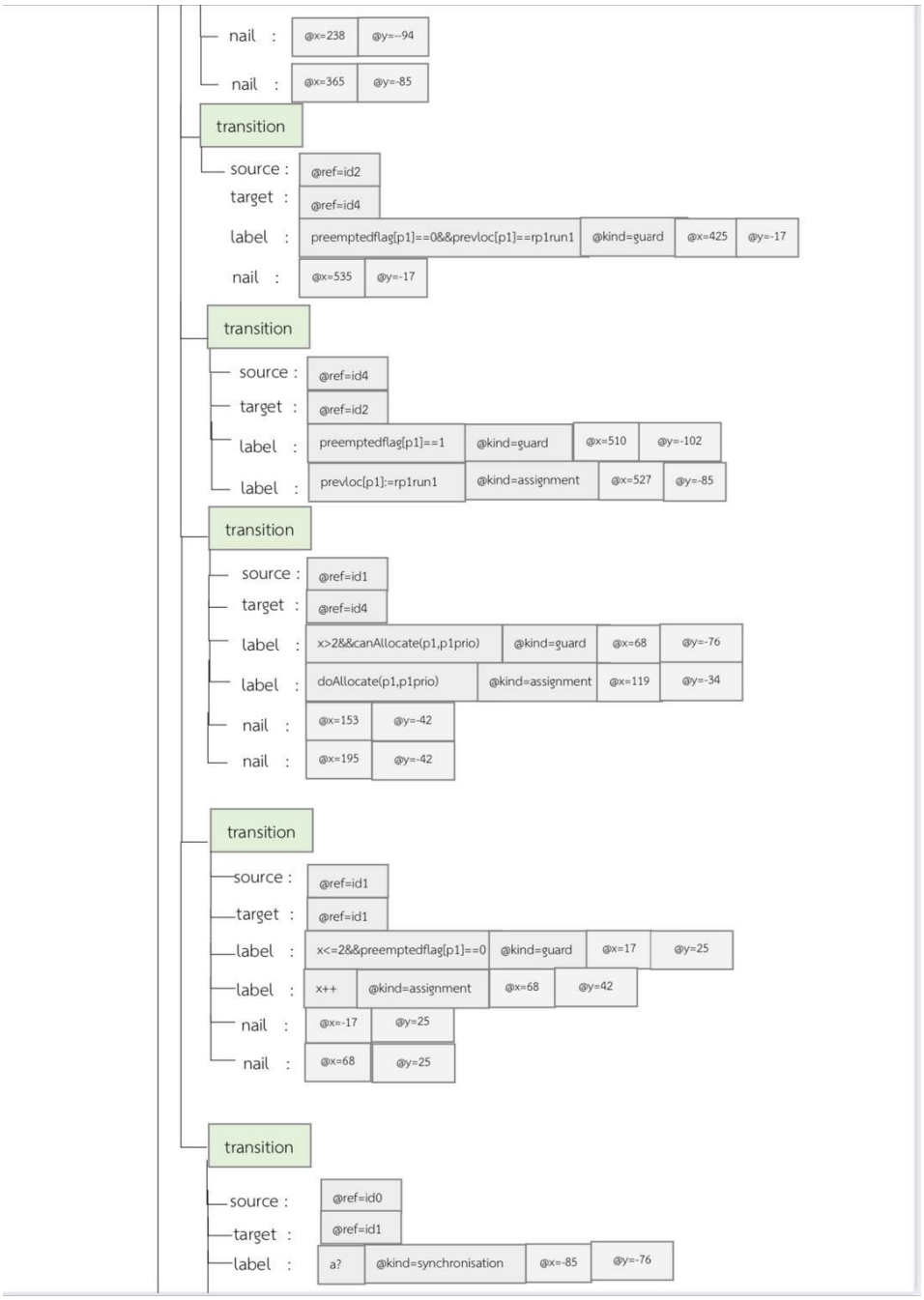
ตัวประกอบไฟล์เอกซ์เอ็มแอล

```
        </transition>
        <transition>
            <source ref="id9"/>
            <target ref="id9"/>
            <label kind="guard" x="-59" y="-323">y&lt;11&amp;&amp;preemptedflag[p2]==0</label>
            <label kind="assignment" x="17" y="-289">y++</label>
            <nail x="-59" y="-289"/>
            <nail x="85" y="-297"/>
        </transition>
        <transition>
            <source ref="id7"/>
            <target ref="id9"/>
            <label kind="guard" x="187" y="-221">preemptedflag[p2]==0&amp;&amp;prevloc[p2]==p2run1</label>
        </transition>
        <transition>
            <source ref="id9"/>
            <target ref="id7"/>
            <label kind="guard" x="212" y="-331">preemptedflag[p2]==1</label>
            <label kind="assignment" x="212" y="-289">prevloc[p2]:=p1run1</label>
            <nail x="280" y="-314"/>
        </transition>
        <transition>
            <source ref="id6"/>
            <target ref="id6"/>
            <label kind="guard" x="-212" y="-76">y&lt;11&amp;&amp;preemptedflag[p2]==0</label>
            <label kind="assignment" x="-153" y="-59">y++</label>
            <nail x="-246" y="-77"/>
            <nail x="-152" y="-77"/>
        </transition>
        <transition>
            <source ref="id6"/>
            <target ref="id9"/>
            <label kind="guard" x="-187" y="-204">y&gt;=3&amp;&amp;canAllocate(p2,p2prio)</label>
            <label kind="assignment" x="-187" y="-144">doAllocate(p2,p2prio)</label>
            <nail x="-110" y="-153"/>
        </transition>
        <transition>
            <source ref="id5"/>
            <target ref="id6"/>
            <label kind="synchronisation" x="-280" y="-178">b?</label>
            <label kind="assignment" x="-290" y="-161">y:=0</label>
        </transition>
    </template>
    <template>
        <name>main</name>
        <location id="id10" x="-144" y="17">
        </location>
        <init ref="id10"/>
        <transition>
            <source ref="id10"/>
            <target ref="id10"/>
            <label kind="guard" x="-25" y="25">z==30</label>
            <label kind="synchronisation" x="-8" y="8">b!</label>
            <nail x="-34" y="17"/>
            <nail x="-51" y="85"/>
        </transition>
        <transition>
            <source ref="id10"/>
            <target ref="id10"/>
            <label kind="guard" x="-144" y="-110">z==3</label>
            <label kind="synchronisation" x="-136" y="-76">a!</label>
            <label kind="assignment" x="-136" y="-93">z:=0</label>
            <nail x="-170" y="-51"/>
            <nail x="-68" y="-59"/>
        </transition>
    </template>
    <system>// Place template instantiations here.
TaskA= taska();
TaskB =taskb();
Main=main();
// List one or more processes to be composed into a system.
system TaskA, TaskB,Main;
    </system>
    <queries>
        <query>
            <formula>E&lt;&gt;</formula>
            <comment></comment>
        </query>
        <query>
            <formula>E&lt;&gt;z&lt;5 and TaskA.Dormant</formula>
            <comment></comment>
        </query>
        <query>
            <formula></formula>
            <comment></comment>
        </query>
        <query>
            <formula></formula>
            <comment></comment>
        </query>
    </queries>
</nta>
```

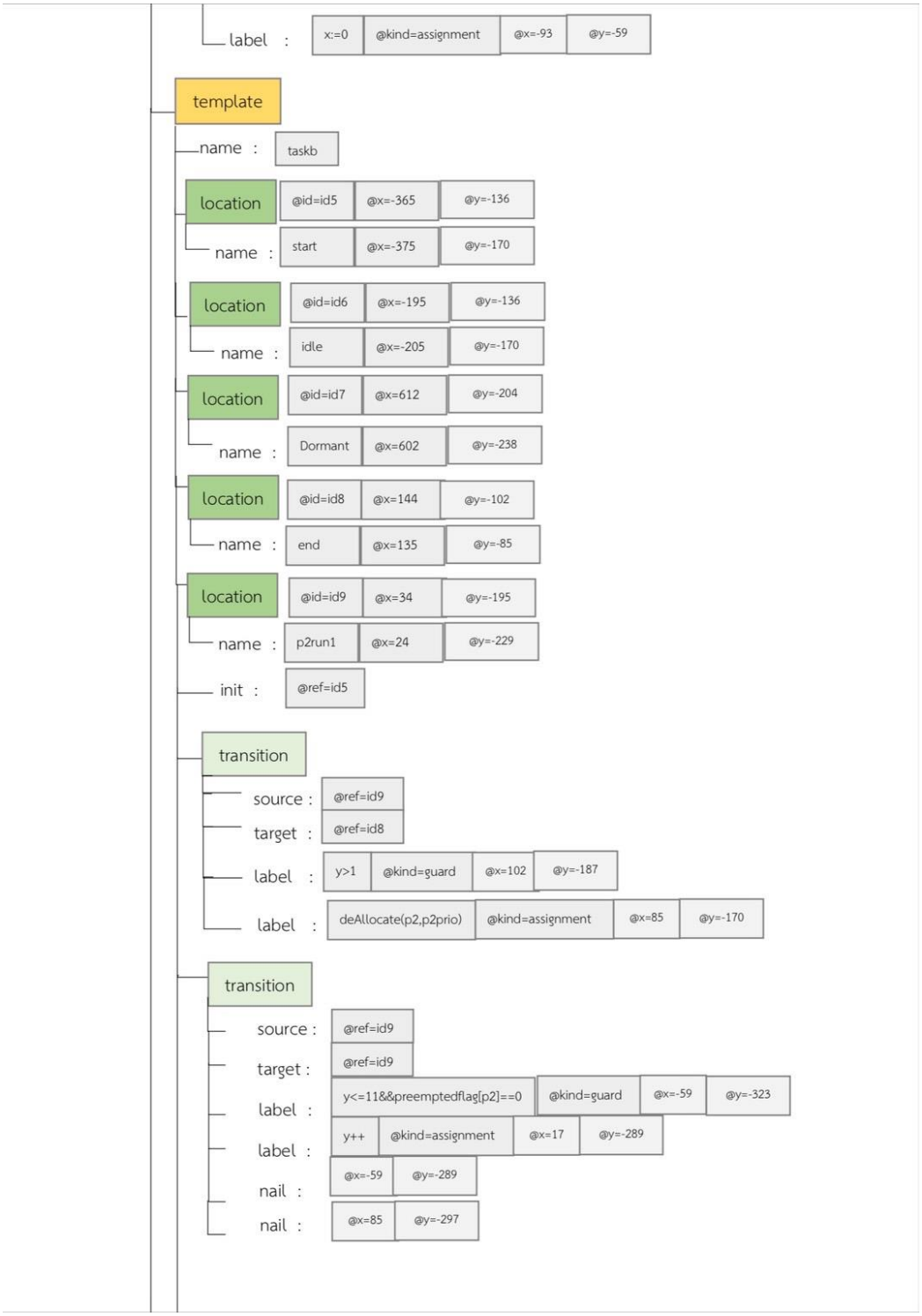
จะได้ไฟล์สกุลเอกซ์ดังตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟดังรูปที่ 3-13



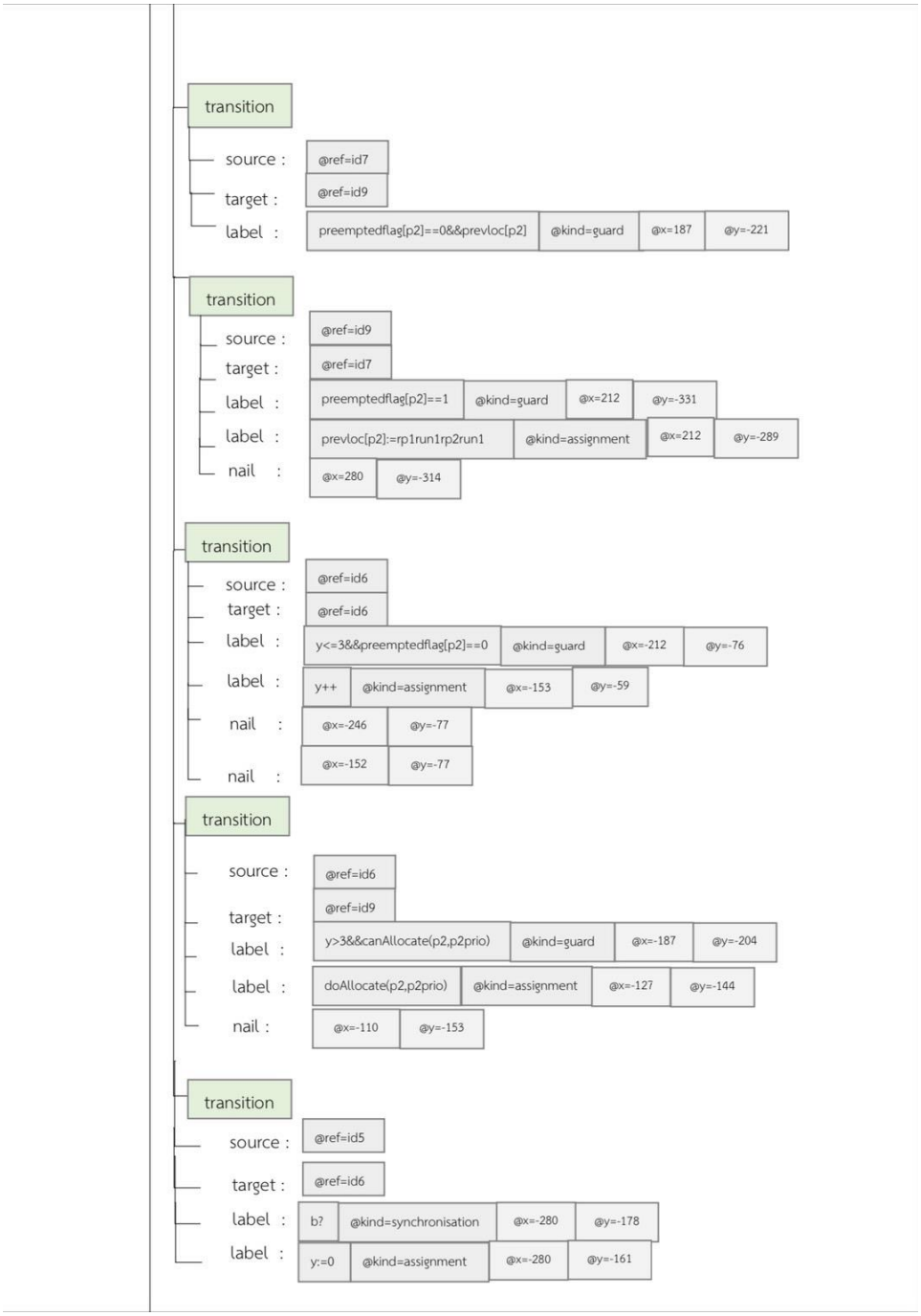
รูปที่ 3-13 ตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ



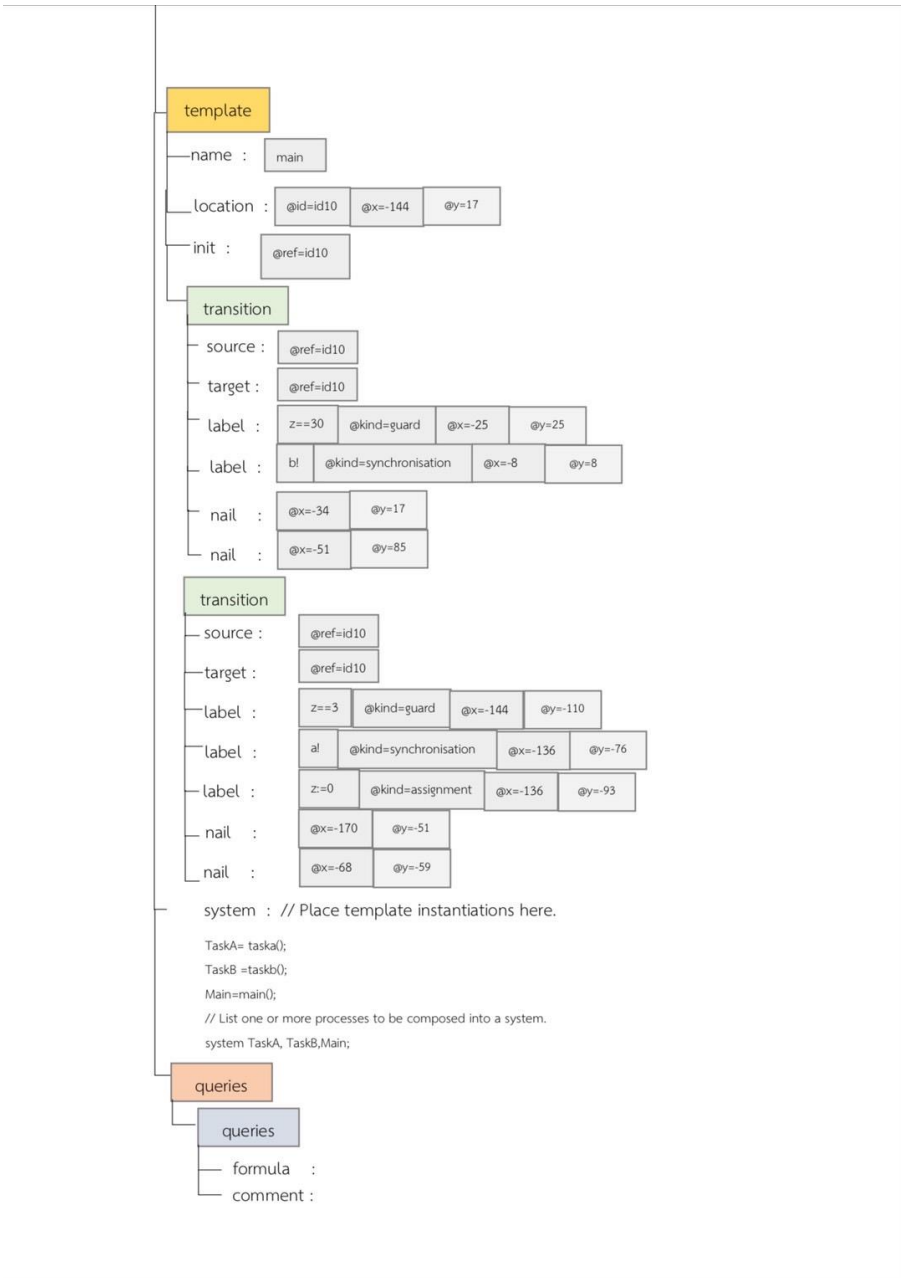
รูปที่ 3-13 ตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ (ต่อ)



รูปที่ 3-13 ตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ (ต่อ)

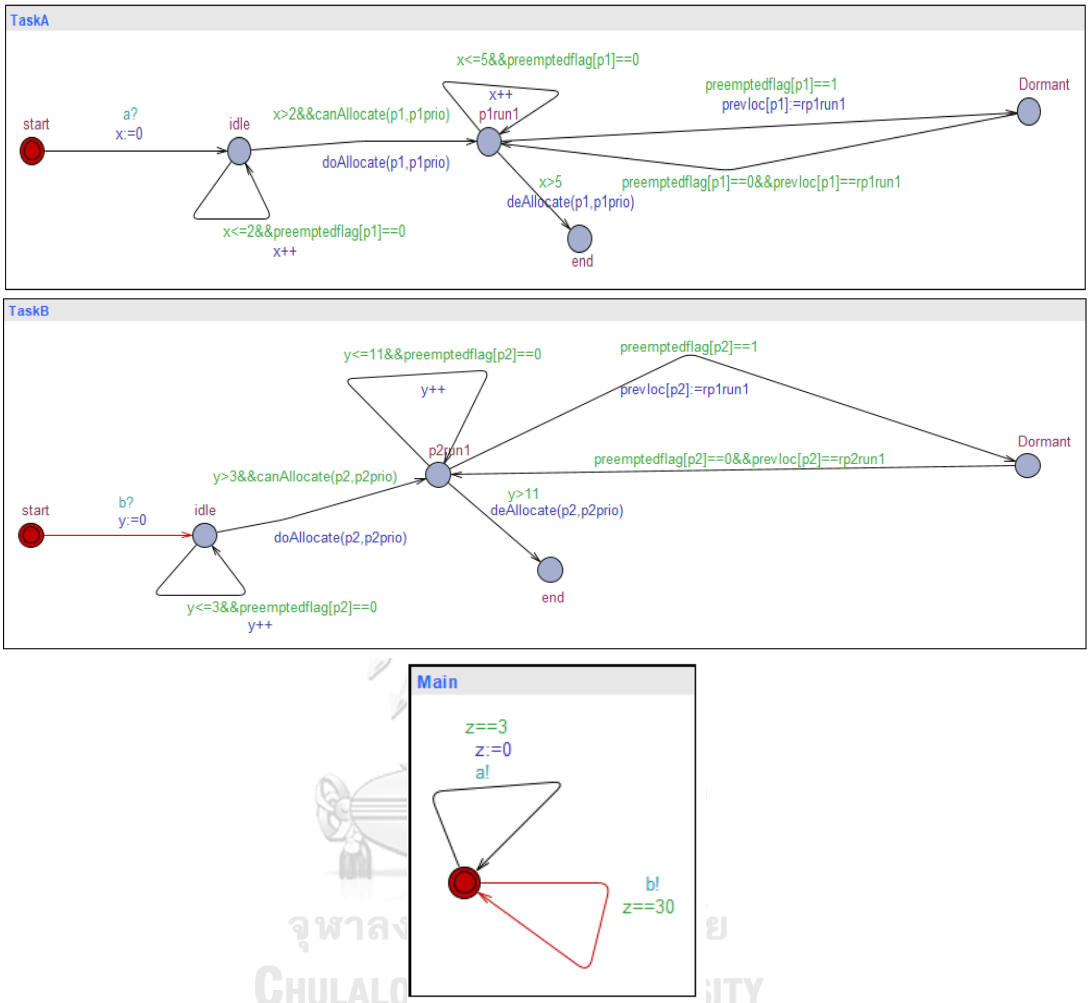


รูปที่ 3-13 ตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ (ต่อ)



รูปที่ 3-13 ตัวอย่างรูปภาพต้นไม้ของแท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ (ต่อ)

ผลลัพธ์การแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ จะได้ดังรูปที่ 3-14



รูปที่ 3-14 ผลลัพธ์การแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

บทที่ 4
การออกแบบและพัฒนาเครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตา สำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

4.1 ภาพรวมของการพัฒนาเครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

สำหรับหัวข้อนี้ จะอธิบายการออกแบบและพัฒนาเครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ ผู้วิจัยได้นำวิธีการการศึกษาข้อมูลในส่วนของวิจัยที่เกี่ยวข้องและแนวทางการการแปลงที่ได้อธิบายในบทที่ 3 นำมาพัฒนาเครื่องมือสำหรับการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ เพื่อที่จะสนับสนุนวิธีการการแปลงนี้ ก่อนการทำวิจัยนี้ ผู้วิจัยได้แสดงถึงแนวคิดของการพัฒนาเครื่องมือ ซึ่งมีรายละเอียดดังต่อไปนี้

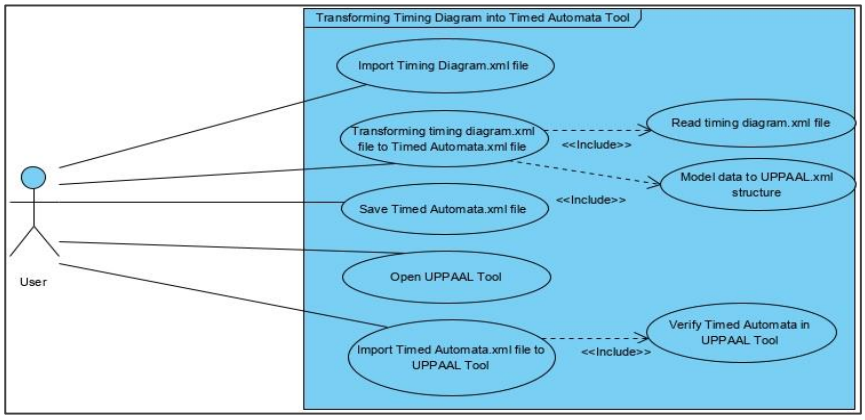
4.1.1 การสร้างตัวแบบซอฟต์แวร์ (Software Modeling)

การสร้างตัวแบบซอฟต์แวร์ในงานวิจัยนี้จะใช้เครื่องมือเป็นแผนภาพยูเอ็มแอล (UML Diagrams) ซึ่งเป็นภาษามาตรฐานของการสร้างตัวแบบด้วยภาพ (Standard visual modeling languge) [7] โดยประกอบด้วย 5 แผนภาพต่อไปนี้

<u>แบบจำลองเชิงหน้าที่ (Function Model)</u>

1) แผนภาพยูสเคส (User-case Diagram) และคำอธิบายยูสเคส (User-case Description) เป็นแผนภาพแสดงปฏิสัมพันธ์ระหว่างระบบกับสภาพแวดล้อมของระบบได้อย่างชัดเจนและเข้าใจได้ง่าย เป็นการเล่าเรื่องราวทั้งหมดของระบบว่ามีการทำงานอะไรบ้างและขอบเขตของระบบและฟังก์ชันการทำงานของระบบ แสดงดังรูปที่ 4-1

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



รูปที่ 4-1แผนภาพยูสเคสของ “Transforming Timing Diagram into Timed Automata Converter”

จากรูปที่ 4-1 สามารถอธิบายยูสเคสทั้งหมดได้ดังตารางที่ 4-1

ตารางที่ 4-1แสดงรายละเอียดของยูสเคสไอดีและยูสเคสทั้งหมดของเครื่องมือ “Transforming Timing Diagram into Timed Automata Converter”

No.	Use-Case ID:	Use-Case Name
1	UC-Timing2UPPAAL-01	Import Timing Diagram.xml file
2	UC-Timing2UPPAAL-02	Transforming Timing Diagram.xml file to Timed Automata.xml file
3	UC-Timing2UPPAAL-03	Read Timing Diagram.xml file
4	UC-Timing2UPPAAL-04	Model data to UPPAAL .xml structure
5	UC-Timing2UPPAAL-05	Save Timed Automata.xml file
6	UC-Timing2UPPAAL-06	Open UPPAAL Tool
7	UC-Timing2UPPAAL-07	Export Timed Automata.xml file
8	UC-Timing2 UPPAAL-08	Verify Timed Automata in UPPAAL Tool

ตารางที่ 4-1 แสดงรายละเอียดของยูสเคสไอดีและยูสเคสทั้งหมดของเครื่องมือแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาที่มีการสนับสนุนการทำพรีเอ็มทีฟ และผู้วิจัยจะแสดงรายละเอียดคำอธิบายยูสเคสซึ่งมีรายละเอียดดังตารางที่ 4-2 ต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 4-2 คำอธิบายยูสเคสของการนำเข้าแผนภาพเวลาในรูปแบบไฟล์เอ็กซ์เอ็มแอล

Use-Case Description	
Use-Case Name: Import timing diagram.xml file	Use-Case ID: UC-Timing2UPPAAL-01
Actor: ผู้ใช้งาน	
Description: ผู้ใช้งานนำเข้าไฟล์แผนภาพเวลาที่อยู่ในรูปแบบไฟล์เอ็กซ์เอ็มแอล	
Precondition: ผู้ใช้งานใช้เครื่องมือ visual paradigm และ Notepad++ สร้างแบบจำลองแผนภาพยูเอ็มแอลนั่นคือ แผนภาพเวลา ที่มีการการบอกลำดับความสำคัญ	
Include: ไม่มี	
Postcondition: เครื่องมือ “Transforming Timing Diagram into Timed Automata converter” จะทำการตรวจสอบว่า ไฟล์แผนภาพเวลาที่นำเข้าเป็นไฟล์เอ็กซ์เอ็มแอลหรือไม่	
Normal Flows: 1. ผู้ใช้งานเปิดเครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” 2. ผู้ใช้งานเลือกปุ่ม “Browse” เพื่อเลือกไฟล์แผนภาพเวลาที่ต้องการจะแปลง	
Alternative Flows: 1a: กรณีที่ผู้ใช้งานไม่เลือกไฟล์แผนภาพเวลา ผู้ใช้งานต้องกดปุ่ม “Cancel” เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการยกเลิกการเลือกไฟล์แผนภาพเวลา และกลับสู่หน้าหลักของเครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” 2a: กรณีที่ผู้ใช้งานไม่เลือกกดปุ่ม “Browse” แต่กดปุ่ม “Submit” เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะแจ้งเตือนผู้ใช้ว่า “Please select a file to Upload”	

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4-3 คำอธิบายยูสเคสของการแปลงไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลาไปเป็นไฟล์เอ็กซ์เอ็มแอลของไทมด์ออโตมาตา

Use-Case Description	
Use-Case Name: Transforming timing diagram.xml file to Timed Automata.xml file	Use-Case ID: UC-Timing2UPPAAL-02
Actor: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter”	
Description: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการ Mapping กฎที่ผู้วิจัยออกแบบของแต่ละกฎของแผนภาพเวลาที่อ่านเข้ามา	
Precondition: ผู้ใช้งานเลือกไฟล์แผนภาพเวลาที่จะนำเข้าสู่เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” เพื่อนำไปแปลง	
Include: Read Timing Diagram.xml file และ Model data to UPPAAL .xml structure	
Postcondition: เมื่อเครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” ทำการแปลงตามกฎที่ผู้วิจัยออกแบบขึ้นมาเรียบร้อยแล้ว เครื่องมือจะทำการส่งออกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอล	
Normal Flows: 1. หลังจากผู้ใช้งานกดปุ่ม “Submit” เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการจับคู่กฎทั้ง 7 ข้อของแต่ละกฎของแผนภาพเวลาไปเป็นไทมด์ออโตมาตา 2. เมื่อเครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” ทำการแปลงตามกฎที่ผู้วิจัยออกแบบขึ้นมาเรียบร้อยแล้ว เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะส่งออกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูป ไฟล์เอ็กซ์เอ็มแอล	
Alternative Flows: 2a: กรณีผู้ใช้งานกดปุ่มกากบาท สัญลักษณ์ “X” เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการยกเลิกการแปลงนั้น	

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4-4 คำอธิบายยูสเคสของการอ่านไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลา

Use-Case Description	
Use-Case Name: Read timing diagram.xml file	Use-Case ID: UC-Timing2UPPAAL-03
Actor: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter”	
Description: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะอ่านไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลาเข้ามา ที่ผู้ใช้งานได้เลือกไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลาไว้	
Precondition: ผู้ใช้งานต้องเลือกไฟล์สกุลเอ็กซ์เอ็มแอลของแผนภาพเวลาที่เป็นไปตามกฎที่ผู้วิจัยได้ออกแบบไว้	
Include: ไม่มี	
Postcondition: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการอ่านไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลา	
Normal Flows: 1.เมื่อผู้ใช้งานกดปุ่ม “Submit” เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการอ่านและจับคู่กฎของแต่ละกฎของแผนภาพเวลาไปเป็นไทมด์ออโตมาตา	
Alternative Flows: 1a: กรณีผู้ใช้งานกดปุ่มกากบาท สัญลักษณ์ “X” เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะไม่ทำการอ่านไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลา	

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 4-5 คำอธิบายยูสเคสของการสร้างข้อมูลแบบจำลองไปเป็นโครงสร้างไฟล์เอ็กซ์เอ็มแอลของเครื่องมือ UPPAAL

Use-Case Description	
Use-Case Name: Model data to UPPAAL.xml structure	Use-Case ID: UC-Timing2UPPAAL-04
Actor: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter”	
Description: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการจับคู่กฎของแต่ละกฎของแผนภาพเวลาไปเป็นไทมด์ออโตมาตา	
Precondition: ผู้ใช้งานต้องเลือกไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลาที่เป็นไปตามกฎที่ผู้วิจัยได้ออกแบบไว้ เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการอ่านไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลามา	
Include: ไม่มี	
Postcondition: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะทำการจับคู่กฎทั้ง 7 ข้อของแต่ละกฎของแผนภาพเวลาไปเป็นไทมด์ออโตมาตา	
Normal Flows 1. เมื่อเครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” ทำการแปลงตามกฎที่ผู้วิจัยออกแบบขึ้นมาเรียบร้อยแล้ว เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะส่งออกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอล	
Alternative Flows: 1a: กรณีผู้ใช้งานกดปุ่มกากบาท สัญลักษณ์ “X” เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะไม่ทำการอ่านไฟล์เอ็กซ์เอ็มแอลของแผนภาพเวลา เครื่องมือจะไม่สามารถส่งออกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอลได้	

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 4-6 คำอธิบายยูสเคสของการบันทึกไฟล์เอ็กซ์เอ็มแอลของไทมด์ออโตมาตา

Use-Case Description	
Use-Case Name: Save Timed Automata.xml file	Use-Case ID: UC-Timing2UPPAAL-05
Actor: ผู้ใช้งาน	
Description: ผู้ใช้งานจะได้ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอล ตามที่อยู่ที่ผู้ใช้บันทึก เพื่อนำเข้าเครื่องมือ UPPAAL	
Precondition: เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” ต้องทำการแปลงตามกฎที่ผู้วิจัยออกแบบขึ้นมาเรียบร้อยแล้วก่อน แล้วส่งออกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอลให้ดาวน์โหลดได้	
Include: ไม่มี	
Postcondition: ผู้ใช้งานจะได้ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอล ตามที่อยู่ที่ผู้ใช้บันทึก	
Normal Flows: 1. เมื่อเครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” ทำการแปลงตามกฎที่ผู้วิจัยออกแบบขึ้นมาเรียบร้อยแล้ว เครื่องมือ “Transforming Timing Diagram into Timed Automata Converter” จะส่งออกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูป ไฟล์เอ็กซ์เอ็มแอล 2. ผู้ใช้งานจะได้ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอล ตามที่อยู่ที่ผู้ใช้บันทึก	
Alternative Flows: 2a: เมื่อผู้ใช้ไม่กดปุ่ม “Download” จะไม่มีไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอลปรากฏ	

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 4-7 คำอธิบายยูสเคสของการเปิดเครื่องมือ UPPAAL

Use-Case Description	
Use-Case Name: Open UPPAAL Tool	Use-Case ID: UC-Timing2UPPAAL-06
Actor: ผู้ใช้งาน	
Description: ผู้ใช้งานเปิดเครื่องมือ UPPAAL	
Precondition: ผู้ใช้งานจะต้องมีไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอลตามที่อยู่ที่ผู้ใช้บันทึก	
Include: ไม่มี	
Postcondition: ผู้ใช้งานสามารถเปิดเครื่องมือ UPPAAL ได้	
Normal Flows: 1.ผู้ใช้งานเปิดเครื่องมือ UPPAAL	
Alternative Flows: 1a: ผู้ใช้งานต้องมีเครื่องมือ UPPAAL ที่ดาวน์โหลดไว้อยู่ในเครื่องคอมพิวเตอร์	

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4-8 คำอธิบายยูสเคสของการนำเข้าไฟล์เอ็กซ์เอ็มแอลของไทมด์ออโตมาตาไปยังเครื่องมือ UPPAAL

Use-Case Description	
Use-Case Name: Import Timed Automata .xml file to UPPAAL Tool	Use-Case ID: UC-Timing2UPPAAL-07
Actor: ผู้ใช้งาน (User)	
Description: ผู้ใช้งานนำเข้าไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปแบบเอ็กซ์เอ็มแอลตามที่อยู่ที่ผู้ใช้บันทึก	
Precondition: ผู้ใช้งานจะได้ไทมด์ออโตมาตาที่อยู่ในรูปไฟล์เอ็กซ์เอ็มแอล ตามที่อยู่ที่ผู้ใช้บันทึก เพื่อนำเข้าเครื่องมือ UPPAAL	
Include: ไม่มี	
Postcondition: ผู้ใช้งานต้องได้ไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปเอ็กซ์เอ็มแอล ตามที่อยู่ที่ผู้ใช้บันทึก	
Normal Flows: 1. ผู้ใช้งานเปิดเครื่องมือ UPPAAL ที่อยู่ในเครื่องคอมพิวเตอร์ 2. ผู้ใช้งานเลือกปุ่ม “import” ในเครื่องมือ UPPAAL แล้วเลือกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปเอ็กซ์เอ็มแอล ตามที่อยู่ที่ผู้ใช้บันทึก เข้ามาในเครื่องมือ UPPAAL	
Alternative Flows: 2a: กรณีที่ผู้ใช้งานไม่เลือกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปเอ็กซ์เอ็มแอล ผู้ใช้งานต้องกดปุ่ม “Cancel” เครื่องมือ UPPAAL จะทำการยกเลิกการเลือกไฟล์ และกลับสู่หน้าหลักของเครื่องมือ UPPAAL	

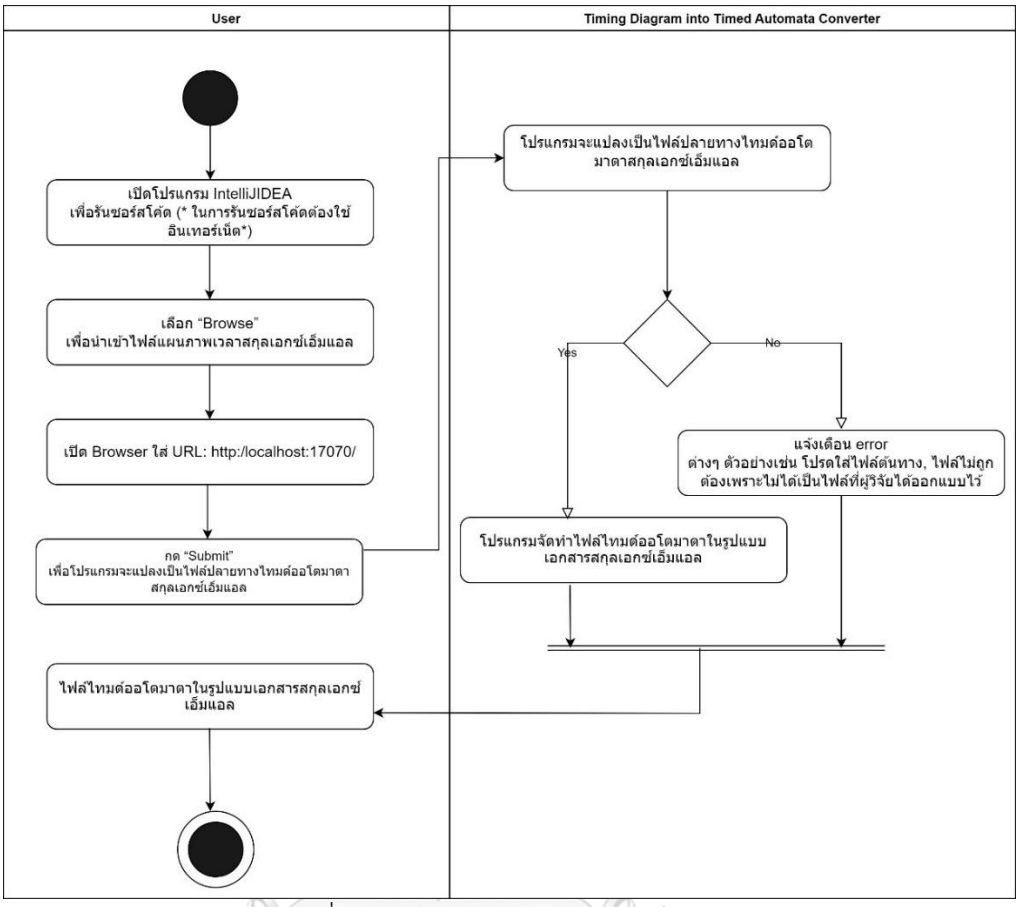
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4-9 คำอธิบายยูสเคสของการทวนสอบการออกแบบไทมด์ออโตมาตาในเครื่องมือ UPPAAL

Use-Case Description	
Use-Case Name: Verify Timed Automata in UPPAAL Tool	Use-Case ID: UC-Timing2UPPAAL-08
Actor: ผู้ใช้งาน	
Description: ผู้ใช้งานนำไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปเอ็กซ์เอ็มแอลที่นำเข้ามาในเครื่องมือ UPPAAL สามารถทวนสอบคุณสมบัติ TCTL ได้	
Precondition: ผู้ใช้งานต้อง“import” ในไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปเอ็กซ์เอ็มแอล ตามที่อยู่ที่ผู้ใช้บันทึกเข้ามาในเครื่องมือ UPPAAL	
Include: Verify Timed Automata in UPPAAL Tool	
Postcondition: ผู้ใช้งานจะทวนสอบไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปเอ็กซ์เอ็มแอลที่นำเข้ามาในเครื่องมือ UPPAAL ได้	
Normal Flows: 1. ผู้ใช้งานเลือกปุ่ม “import” ในเครื่องมือ UPPAAL แล้วเลือกไฟล์ไทมด์ออโตมาตาที่อยู่ในรูปเอ็กซ์เอ็มแอล ตามที่อยู่ที่ผู้ใช้บันทึก เข้ามาในเครื่องมือ UPPAAL 2. ผู้ใช้งานป้อนป้อนข้อมูลคุณสมบัติทวนสอบเพื่อทวนสอบไฟล์ไทมด์ออโตมาตาในเครื่องมือ UPPAAL	
Alternative Flows: 2a: ถ้าผู้ใช้งานไม่ป้อนข้อมูลคุณสมบัติทวนสอบ ผู้ใช้งานจะไม่ได้ทวนสอบในเครื่องมือ UPPAAL	

2) แผนภาพกิจกรรม (Activity Diagram) แสดงให้เห็นการไหลของการทำงานและแสดงขั้นตอนการทำงานของระบบ รูปที่ 4-2

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 4-2 แผนภาพกิจกรรมของเครื่องมือแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูป 4-2 แสดงการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ จะสามารถอธิบายรายละเอียดขั้นตอนได้ดังนี้

1) ผู้ใช้ทำการเตรียมแผนภาพเวลาเข้าเครื่องมือ “Timing Diagram into Timed Automata Converter” ไฟล์สกุลเอ็กซ์เอ็มแอลของแผนภาพเวลา

2) นำเข้าไฟล์หัวข้อที่ 1) เข้าสู่เครื่องมือ เครื่องมือจะตรวจสอบว่าไฟล์อยู่รูปแบบสกุลไฟล์เอกซ์เอ็มแอลและมีการเขียนลำดับความสำคัญ

3) เครื่องแปลง “Timing Diagram into Timed Automata Converter” การแปลงแผนภาพเวลาที่มีลำดับความสำคัญไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

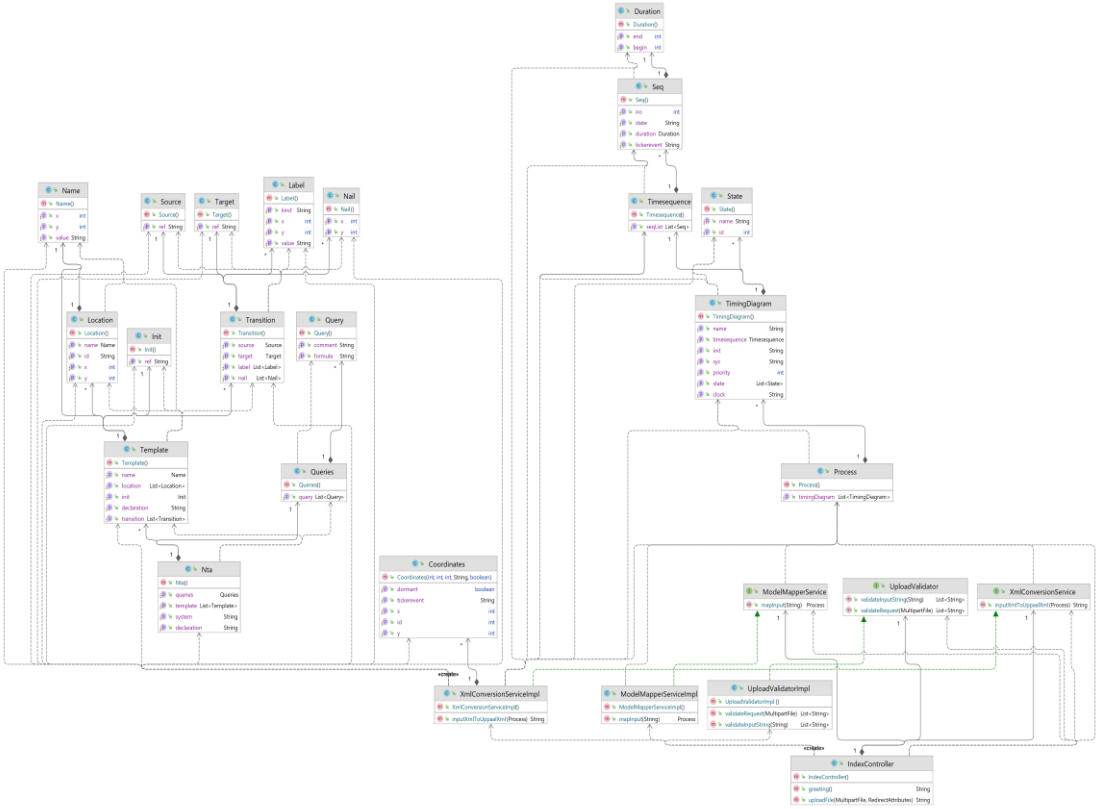
3.1 ในกรณีที่เครื่องมือตรวจสอบพบว่าผู้ใช้ยังไม่ได้เลือกที่อยู่ของไฟล์แผนภาพเวลาสกุลไฟล์เอกซ์เอ็มแอลหรือที่อยู่บันทึกไฟล์ผลลัพธ์ส่งออก เครื่องมือ “Timing Diagram into Timed Automata Converter”จะหยุดการทำงานและแจ้งข้อผิดพลาดให้ทราบ

3.2 ในกรณีที่เครื่องมือ “Timing Diagram into Timed Automata Converter” ตรวจสอบพบว่าไฟล์นำเข้านั้นไม่ถูกต้องหรือไม่ใช่ไฟล์สกุลเอกซ์เอ็มแอล ตัวอย่างเช่น โครงสร้างไฟล์สกุลไฟล์เอกซ์เอ็มแอล เครื่องมือแปลงจะหยุดการทำงานและแจ้งข้อความข้อผิดพลาดให้ผู้ใช้ทราบ

4) เครื่องมือส่งออกไฟล์ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟสกุลไฟล์เอกซ์เอ็มแอล

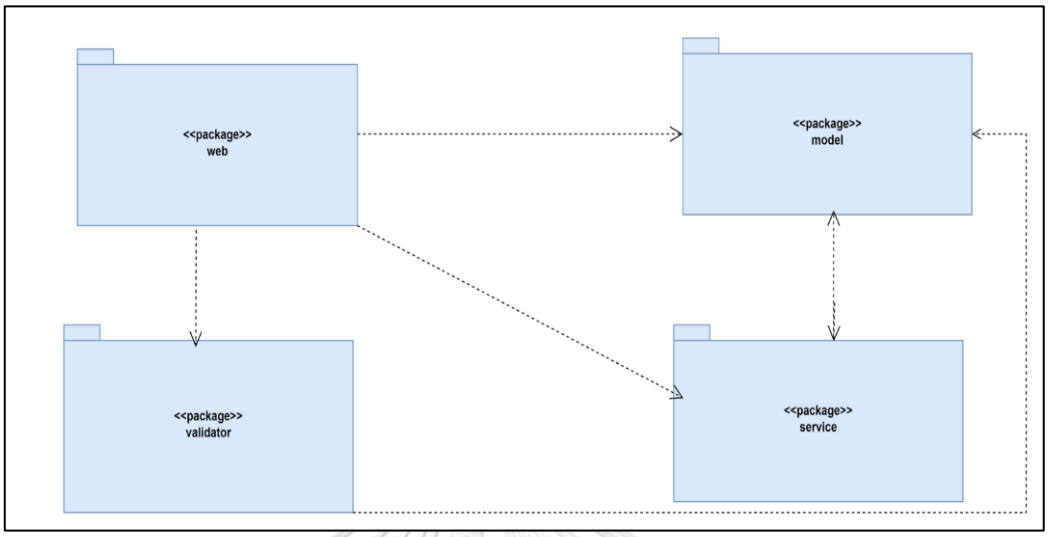
<u>แบบจำลองเชิงโครงสร้าง (Structural Model)</u>

3) แผนภาพคลาส (Class Diagram) สามารถใช้แสดงความสัมพันธ์ระหว่างคลาสของระบบ ทั้งความสัมพันธ์เชิงสถิตย์ (Static Relationship) และความสัมพันธ์เชิงกิจกรรม (Dynamic Relationship) ในส่วนนี้จะแสดงแผนภาพคลาสของเครื่องมือ “Timing Diagram into Timed Automata Converter” เพื่อแสดงให้เห็นถึงความสัมพันธ์ของแต่ละคลาสในเครื่องมือ และจะเป็นการอธิบายโครงสร้างการจัดเก็บข้อมูลของเครื่องมือที่พัฒนาขึ้นโดยใช้แผนภาพคลาส



รูปที่ 4-3 ภาพรวมของแผนภาพคลาสเครื่องมือ “Timing Diagram into Timed Automata Converter”

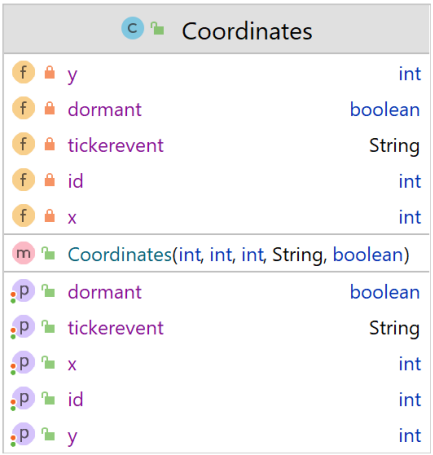
รูปที่ 4-4 แผนภาพคลาสประกอบไปด้วยแพ็กเกจหลัก ซึ่งแบ่งออกเป็น 4 แพ็กเกจ ได้แก่ แพ็กเกจ web, แพ็กเกจ model, แพ็กเกจ service, แพ็กเกจ validator



รูปที่ 4-4 แผนภาพแพ็กเกจของเครื่องมือ “Timing Diagram into Timed Automata Converter” ที่พัฒนา

3.1 แพ็กเกจ model ประกอบด้วยคลาสหลักๆ 3 คลาส ได้แก่ คลาส Coordinates, คลาส Input, คลาส Output ในคลาสจะประกอบด้วยสัญลักษณ์ตัวอักษรภาษาอังกฤษข้างหน้า อันได้แก่ F หมายถึง ฟิลด์ (Fields), M* หมายถึง คอนสตรัคเตอร์ (Constructure), m หมายถึง เมธอด (Method), P หมายถึง คุณสมบัติ (Properties), I หมายถึง ภายในคลาส(Inner class)

• คลาส Coordinates ทำหน้าที่จัดเก็บตัวแปรตำแหน่งหรือที่อยู่ที่มีร่วมกันของคลาส Input และ คลาส Output ซึ่งคลาส Coordinates มีฟิลด์ 5 ฟิลด์ ได้แก่ x หมายถึงตำแหน่งแกน x, y หมายถึงตำแหน่งแกน y, dormant หมายถึงหยุดนิ่ง, tickerevent หมายถึงเหตุการณ์สำคัญ เช่น canAllocate, id หมายถึงหมายเลขกำหนดงาน



รูปที่ 4-5 แผนภาพคลาสย่อยในโมเดลชื่อ Coordinates

• คลาส Input มีทั้งหมด 6 คลาส ได้แก่ คลาส Process, คลาส TimingDiagram, คลาส State, คลาส Timesequence, คลาส Seq, คลาส Duration

1. คลาส Process มีหน้าที่จัดเก็บตัวแปรฟิลด์ของแผนภาพเวลา

2. คลาส TimingDiagram มีหน้าที่จัดเก็บตัวแปรเกี่ยวกับแผนภาพเวลา มี 5 ฟิลด์ ได้แก่ ลำดับความสำคัญ (priority), จุดเริ่มต้นโลเคชัน (init), ตัวแปรนาฬิกา (clock), สถานะ (state), ลำดับเวลา (timesequence), ชื่อสถานะแผนภาพเวลา (name), การคุยกันระหว่างเส้น (syc)

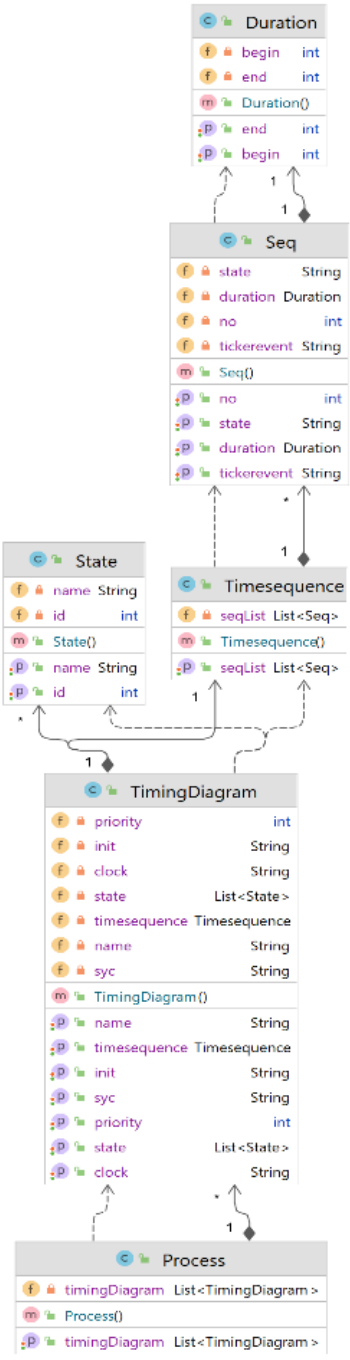
3. คลาส State มีหน้าที่จัดเก็บตัวแปรเกี่ยวกับสถานะ มี 2 ฟิลด์ ได้แก่ ชื่อสถานะ (name), กำหนดหมายเลขงาน (id)

4. คลาส Timesequence มีหน้าที่จัดเก็บตัวแปรเกี่ยวกับลำดับเวลา มี 1 ฟิลด์ ได้แก่ ลำดับรายการ (seqList)

5. คลาส Seq มีหน้าที่จัดเก็บตัวแปรเกี่ยวกับลำดับ มี 4 ฟิลด์ ได้แก่ สถานะ (state), ระยะเวลา (duration), หมายเลข (no), เหตุการณ์สำคัญ (tickerevent)

6. คลาส Duration มีหน้าที่จัดเก็บตัวแปรเกี่ยวกับระยะเวลา มี 2 ฟิลด์ ได้แก่ เริ่มต้น (begin), สิ้นสุด (end)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



รูปที่ 4-6 แผนภาพคลาสย่อยในโมเดลชื่อ Input

• คลาส output เป็นคลาสสำหรับแสดงผลลัพธ์หรือนำออกของแบบจำลองไทมด์ออโตมาตา ประกอบด้วยคลาสทั้งหมด 12 คลาส ได้แก่ คลาส Nta, คลาส Query, คลาส Queries, คลาส

Template, คลาส Location, คลาส Nail, คลาส lnit, คลาส Name, คลาส Target, คลาส Transition, คลาส Source, คลาส Label, คลาส Nail

1. คลาส Nta มีหน้าที่จัดเก็บตัวแปรแท็กเอกซ์เอ็มแอลของไทมด์ออโตมาตา มีทั้งหมด 4 ฟิลด์ ได้แก่ กำหนดตัวแปร (declaration), สำหรับทวนสอบ TCTL (queries), ระบุรายละเอียดทั้งหมดภายในออโตมาตอน (template), การเรียกออโตมาตอน (system)

2. คลาส Query มีหน้าที่จัดเก็บตัวแปรสำหรับทวนสอบ TCTL มีทั้งหมด 2 ฟิลด์ ได้แก่ สูตรการทวนสอบ (formula), ความคิดเห็น (comment)

3. คลาส Queries มีหน้าที่จัดเก็บตัวแปรสำหรับทวนสอบ TCTL มีทั้งหมด 1 ฟิลด์ ได้แก่ รายการคิวรี (query)

4. คลาส Template มีหน้าที่จัดเก็บตัวแปรระบุรายละเอียดทั้งหมดภายในออโตมาตอน มี 5 ฟิลด์ ได้แก่ จุดเริ่มต้นโลเคชัน (init), โลเคชัน (location), กำหนดตัวแปร (declaration), ชื่อเทมเพลต (name), รายละเอียดของเส้น (transition)

5. คลาส Location มีหน้าที่จัดเก็บตัวแปรโลเคชัน มี 4 ฟิลด์ ได้แก่ ตำแหน่งแกน x (x), ตำแหน่งแกน y (y), กำหนดหมายเลขงาน (id), ชื่อโลเคชัน (name)

6. คลาส Nail มีหน้าที่จัดเก็บตัวแปรขนาดของเส้นมี 2 ฟิลด์ ได้แก่ ตำแหน่งแกน x (x), ตำแหน่งแกน y (y)

7. คลาส lnit มีหน้าที่จัดเก็บตัวแปรจุดเริ่มต้นโลเคชันมี 1 ฟิลด์ ได้แก่ โลเคชันอ้างอิง (ref)

8. คลาส Name มีหน้าที่จัดเก็บตัวแปรชื่อของไทมด์ออโตมาตา มีทั้งหมด 3 ฟิลด์ ได้แก่ ตำแหน่งแกน x (x), ตำแหน่งแกน y (y), ค่า (value)

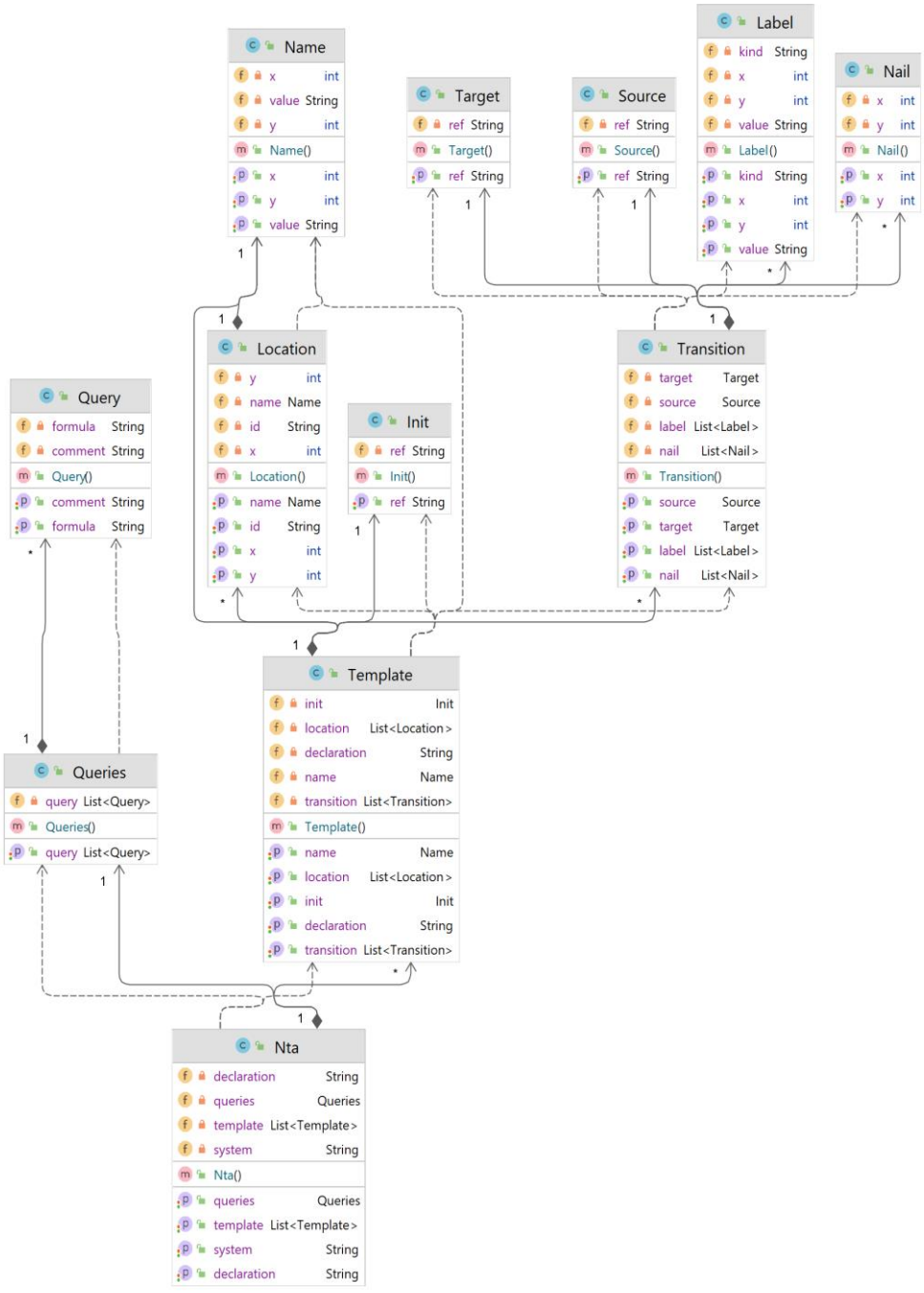
9. คลาส Target มีหน้าที่จัดเก็บตัวแปรระบุโลเคชันถัดไป มีทั้งหมด 1 ฟิลด์ ได้แก่ อ้างอิงโลเคชันถัดไป (ref)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

10. คลาส Transition มีหน้าที่จัดเก็บตัวแปรระบุรายละเอียดของเส้น มีทั้งหมด 4 ฟิลด์ ได้แก่ ระบุโลเคชันถัดไป (target), ระบุโลเคชันก่อนหน้า (source), ตัวแปรชนิดของ Label ต่าง ๆ (label), ขนาดของเส้น (nail)

11. คลาส Source มีหน้าที่จัดเก็บตัวแปรระบุโลเคชันก่อนหน้า มีทั้งหมด 1 ฟิลด์ ได้แก่ ระบุโลเคชันก่อนหน้า (ref)

12. คลาส Label มีหน้าที่จัดเก็บตัวแปรชนิดของ Label ต่าง ๆ เช่น Guard และ Channel มีทั้งหมด 4 ฟิลด์ ได้แก่ ชนิดของ Label (kind), ค่า (value), ตำแหน่งแกน x (x), ตำแหน่งแกน y (y)



รูปที่ 4-7 แผนภาพคลาสย่อยในโมเดลชื่อ Output

หลังจากได้คลาสในคลาส input ซึ่งเป็นแพ็กเกจเกี่ยวกับการนำเข้าข้อมูลแผนภาพเวลา และคลาส output เกี่ยวกับข้อมูลนำออกของไทมด์ออโตมาตา

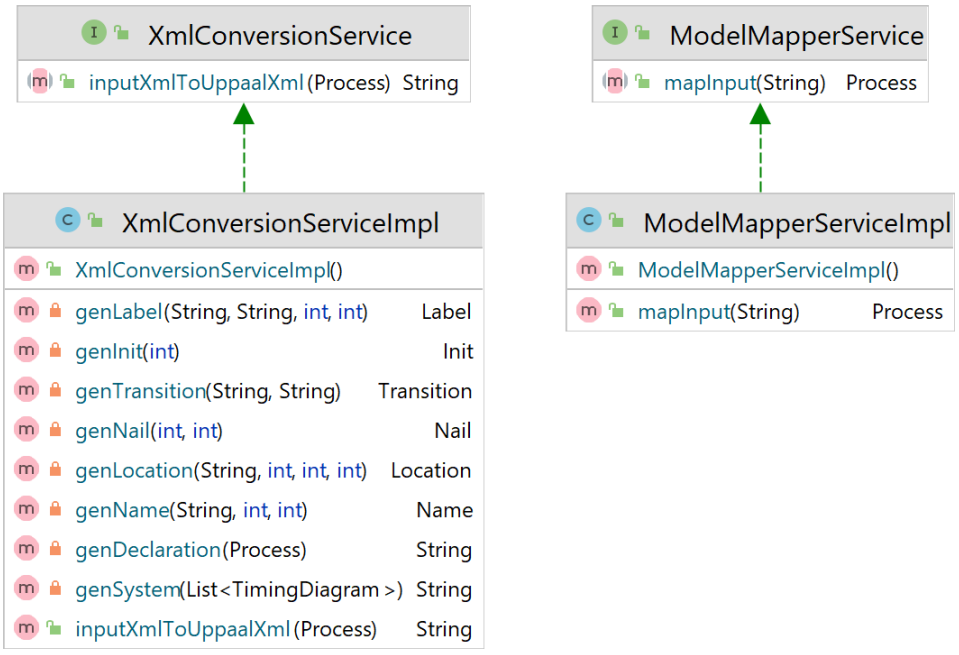
3.2 แพ็กเกจ service เป็นคลาสสำหรับเซอร์วิสหลักของเครื่องมือที่มีฟังก์ชันใช้งานในการแปลงแผนภาพเวลาไปเป็นไฟล์ไทมด์ออโตมาตาจะทำหน้าที่เซอร์วิสตอนนำเข้าไฟล์แผนภาพเวลาและเซอร์วิสสำหรับจับคู่กฎ 7 ข้อที่ผู้วิจัยได้ออกแบบไว้ รายละเอียดดังรูปที่ 4-7 ซึ่งแพ็กเกจ service ประกอบด้วยคลาสทั้งหมด 4 คลาส ได้แก่ คลาส ModelMapperService, คลาส XmlConversionService, คลาส ModelMapperServiceImpl, คลาส XmlConversionServiceImpl

1. คลาส ModelMapperService มีหน้าที่จัดเก็บตัวแปรบริการการจับคู่โมเดล

2. คลาส ModelMapperServiceImpl มีหน้าที่จัดเก็บตัวแปรบริการการดำเนินการจับคู่โมเดล

3. คลาส XmlConversionService มีหน้าที่จัดเก็บตัวแปรบริการการแปลงเอกซ์เอ็มแอล

4. คลาส XmlConversionServiceImpl หน้าที่จัดเก็บตัวแปรบริการดำเนินการการแปลงเอกซ์เอ็มแอล

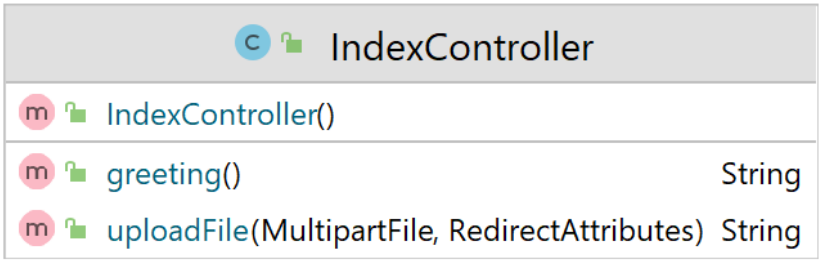


รูปที่ 4-8 แผนภาพคลาสย่อยในเซอร์วิส

3.3 แพ็กเกจ Web จัดการข้อมูลในหน้าเว็บ ซึ่งเป็นส่วนควบคุมการแสดงผลให้ผู้ใช้เห็น ตัวอย่างเช่น ปุ่ม “Browse”, ปุ่ม “Download” เป็นต้น โดยรับข้อมูลนำเข้าจากผู้ใช้งานระบบก็คือ

แผนภาพเวลา แล้วมาตัดสินใจว่าทำงาน นำออกไฟล์อย่างไร แพ็กเกจ Web จะมีคลาส 1 คลาส อันได้แก่ คลาส IndexController

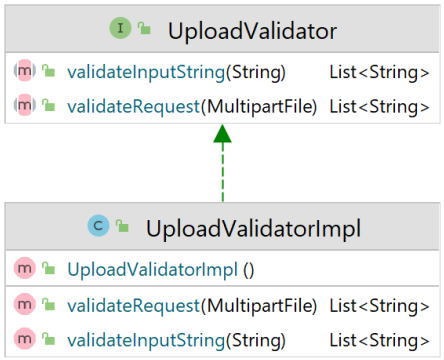
1. คลาส IndexController มีหน้าที่จัดเก็บตัวแปรส่วนควบคุมการแสดงผลให้ผู้ใช้เห็น



รูปที่ 4-9 แผนภาพคลาสย่อยในเว็บ

3.4 แพ็กเกจ validator เป็นการตรวจสอบความถูกต้องของข้อมูลนำเข้า ซึ่งประกอบด้วย 2 คลาส ได้แก่ คลาส UploadValidator, คลาส UploadValidatorImpl ดังรูปที่ 4-10

1. คลาส UploadValidator มีหน้าที่จัดเก็บตัวแปรตรวจสอบการอัปโหลด
2. คลาส UploadValidatorImpl มีหน้าที่จัดเก็บตัวแปรตรวจสอบดำเนินการการอัปโหลด

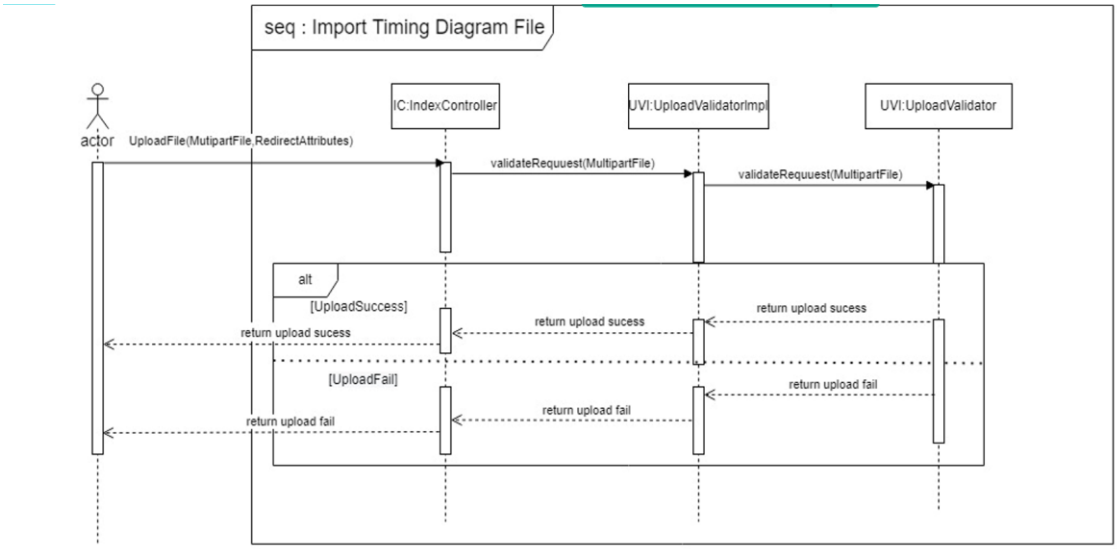


รูปที่ 4-10 แผนภาพคลาสย่อยใน validator

แบบจำลองเชิงพฤติกรรม (Behavior Model)

4) แผนภาพการทำงานแบบลำดับ (Sequence Diagram) แสดงลำดับการมีปฏิสัมพันธ์ (Interaction) ระหว่างวัตถุ โดยดูจากลำดับการส่งข้อความ (Message) จากวัตถุหนึ่งไปยังวัตถุหนึ่ง ตัวอย่างแผนภาพลำดับงานจะอ้างอิงจากคำอธิบายยูสเคสข้างต้น โดยหัวข้อที่ 4.1 และ 4.2 แสดงแผนภาพลำดับงานของกรณีผู้ใช้งาน Import Timing Diagram File กับแผนภาพลำดับงานของกรณีผู้ใช้งาน Transform Timing Diagram to Timed Automata

4.1 แผนภาพลำดับงานกรณีผู้ใช้งาน Import Timing Diagram File

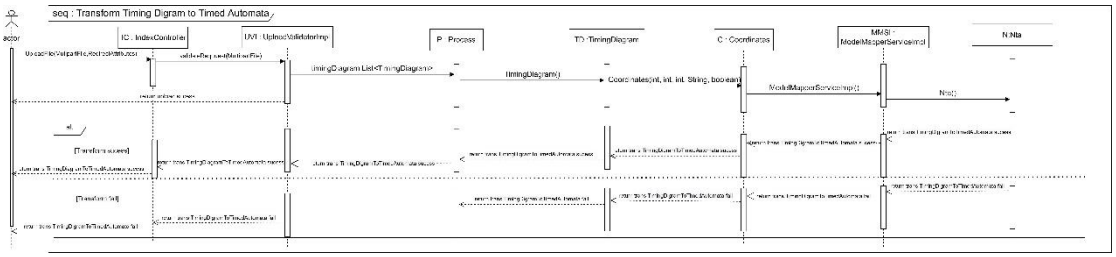


รูปที่ 4-11 แผนภาพลำดับงานของกรณีผู้ใช้งาน Import Timing Diagram File

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูป 4-11 ผู้ใช้จะทำการกดปุ่ม “Browse” เข้ามาในคลาส IndexController ผ่านหน้าเว็บ และเลือกอัปโหลดไฟล์แผนภาพเวลาในคลาส UploadValidatorImp และคลาส UploadValidator จะทำการตรวจสอบว่า เป็นตามไฟล์เอกซ์เอ็มแอลหรือไม่ ถ้าเป็นจะมีการบอกว่า อัปโหลดสำเร็จ ถ้าไม่สำเร็จจะมีการแจ้งเตือนข้อผิดพลาด

4.2 แผนภาพลำดับงานกรณีผู้ใช้งาน Transform Timing Diagram to Timed Automata

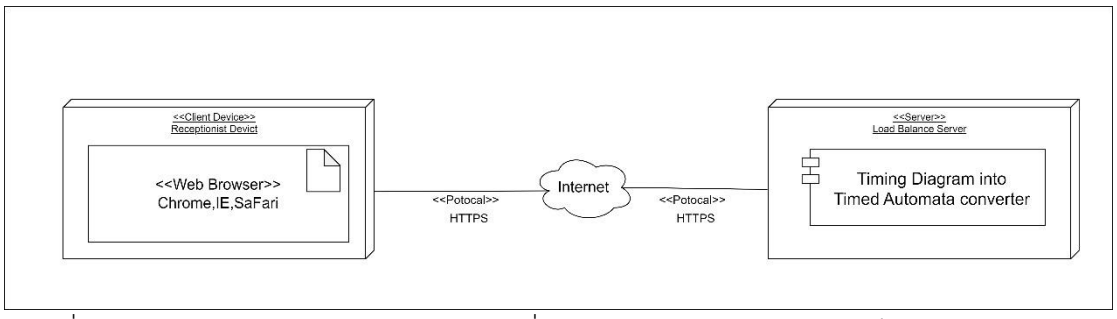


รูปที่ 4-12 แผนภาพลำดับงานของกรณีกรณีใช้งาน Transform Timing Diagram File to Timed Automata File

รูปที่ 4-12 ผู้ใช้จะทำการกดปุ่ม “Browse” เข้ามาในคลาส IndexController ผ่านหน้าเว็บ และเลือกอัปโหลดไฟล์แผนภาพเวลาในคลาส UploadValidatorImp เสร็จแล้วจะส่งข้อความว่าอัปโหลดสำเร็จ หลังจากนั้นผู้ใช้กดปุ่ม “Submit” คลาส process ของการนำเข้าแผนภาพเวลาต้นทาง จะทำการจับคู่กฎ 7 ข้อและตรวจสอบแท็กเอกซ์เอ็มแอลของไฟล์ต้นทางและปลายทางในคลาส ModelMapperServiceImp และนำออกเป็นไฟล์ไทมด์ออโตมาตาในคลาสหลัก คือ Nta ถ้าแปลงผ่านจะสามารถดาวน์โหลดไฟล์ เอกซ์เอ็มแอลปลายทางได้ หรือถ้าไม่ผ่านจะแสดงข้อผิดพลาดแจ้งเตือนไว้

แบบจำลองเชิงสถาปัตยกรรม (Architectural Model)

5) แผนภาพดีพอยเมนท์ (Deployment Diagram) แสดงสถาปัตยกรรมฮาร์ดแวร์และซอฟต์แวร์ของเฟรมเวิร์คของการแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ สามารถอธิบายเกี่ยวกับการออกแบบเชิงสถาปัตยกรรมของเครื่องมือฝั่งผู้ใช้อุปกรณ์ (Client Device) โดยผู้ใช้อุปกรณ์ต้องใช้โปรแกรม Timing Diagram into Timed Automata Converter ผ่านเว็บเบราว์เซอร์ แสดงรูปที่ 4-13



รูปที่ 4-13 แผนภาพดีพอยเมนท์การพัฒนาเครื่องมือการแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

4.1.2 ภาพรวมการทำงานของเครื่องมือ

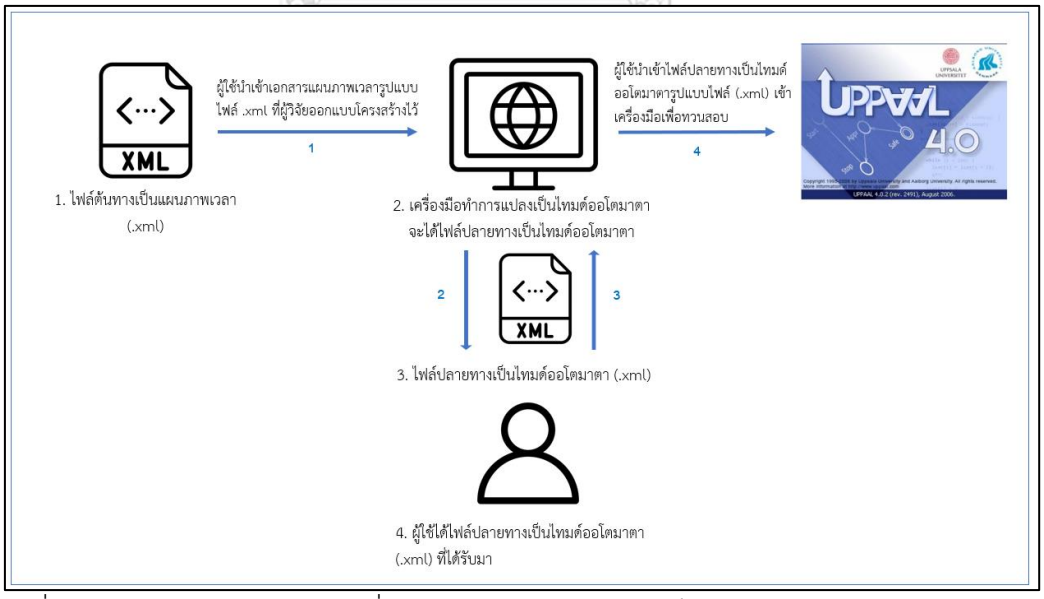
ในส่วนนี้ภาพรวมการทำงานของเครื่องมือรายละเอียดในภาพรวมการแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ จะถูกพัฒนาโดยใช้ภาษาจาวาและอยู่ในรูปแบบของเว็บ โดยเครื่องมือ “Timing Diagram into Timed Automata Converter” จะทำการแปลงส่วนประกอบของแผนภาพเวลา TDTP ไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟโดยใช้กฎการแปลงที่ผู้วิจัยออกแบบไว้ ภาพรวมการทำงานของเครื่องมือนี้จะแสดงดังรูป 4-13 มีรายละเอียดดังนี้

1) ผู้ใช้นำเข้าไฟล์สกุลเอกซ์เอ็มแอลแผนภาพเวลา TDTP แผนภาพเวลาจะต้องมีโครงสร้างเอกซ์เอ็มแอลที่ผู้ใช้ออกแบบไว้และตามโครงสร้างมาตรฐานเอกซ์เอ็มแอล

2) เมื่อนำเข้าไฟล์สกุลเอกซ์เอ็มแอลแผนภาพเวลา TDTP ไปยังเครื่องมือ “Timing Diagram into Timed Automata Converter” จากกการนำเข้าไฟล์สกุลเอกซ์เอ็มแอลแผนภาพเวลา TDTP แล้วใช้กฎการจับคู่ 7 ข้อที่ผู้วิจัยออกแบบไว้

3) ได้ผลลัพธ์ข้อมูลออกมาเป็นไฟล์สกุลเอกซ์เอ็มแอลไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

4) ผู้ใช้งานระบบนำเข้าไฟล์สกุลเอกซ์เอ็มแอลไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟในเครื่องมือทวนสอบ UPPAAL



รูปที่ 4-14 แผนภาพการทำงานเครื่องมือแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

4.1.3 ข้อกำหนดความต้องการเชิงฟังก์ชันของเครื่องมือ (Functional Requirements)

เครื่องมือ “Timing Diagram into Timed Automata Converter” สนับสนุนที่พัฒนาขึ้นต้องสามารถทำงานได้ดังนี้

1) ผู้ใช้สามารถนำเข้าข้อมูลแผนภาพเวลา TDTP ที่จัดเตรียมในรูปแบบไฟล์สกุลเอกซ์เอ็มแอลที่เครื่องมือกำหนดเท่านั้น

2) ผู้ใช้งานสามารถแปลงข้อมูลแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟในรูปแบบไฟล์สกุลเอกซ์เอ็มแอลโดยใช้กฎการแปลงที่ผู้วิจัยออกแบบไว้ได้

3) ผู้ใช้สามาถส่งออกแบบจำลองไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟในรูปแบบไฟล์สกุลเอกซ์เอ็มแอลที่สามารถนำไปเข้าเครื่องมือ UPPAAL ได้

4.1.4 ข้อจำกัดเครื่องมือ

เครื่องมือสนับสนุนการแปลงข้อมูลแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ เครื่องมือที่ชื่อว่า “Timing Diagram into Timed Automata Converter” มีข้อจำกัดดังนี้

1) เครื่องมือที่พัฒนาขึ้นรองรับระบบปฏิบัติการวินโดวส์เท่านั้น

2) ไฟล์ที่นำเข้าเครื่องมือต้องเป็นไฟล์สกุลเอกซ์เอ็มแอลที่ผู้วิจัยออกแบบไว้เท่านั้น และเป็นไปตามมาตรฐานเอกซ์เอ็มแอลซึ่งมีแท็กปิดและเปิด

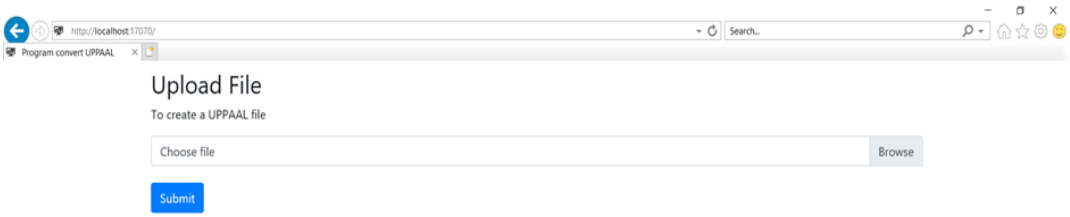
3) เวลาในแผนภาพเวลาที่ระบุไว้ในไฟล์สกุลเอกซ์เอ็มแอลต้องอยู่ในรูปแบบจำนวนเต็มบวกเท่านั้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.1.5 โครงสร้างของส่วนต่อประสานกับผู้ใช้ระบบ

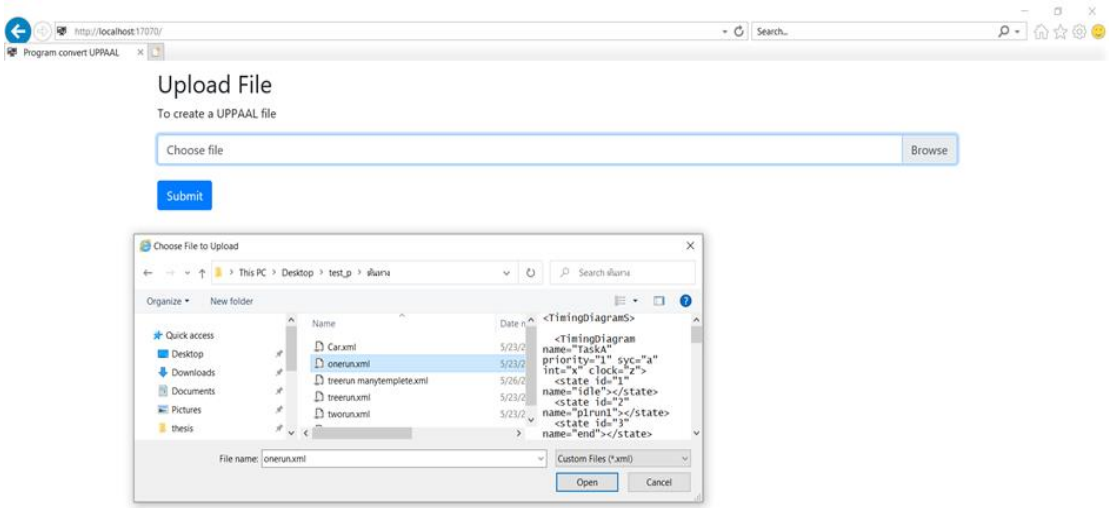
ผู้วิจัยพัฒนาเครื่องมือให้สำหรับผู้ใช้งานแปลงแผนภาพเวลาเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟเครื่องมือนี้จะขอใช้ชื่อว่า “Timing Diagram into Timed Automata Converter”โดยเครื่องมือนี้จะเริ่มทำงานตั้งแต่ในขั้นตอนการนำเข้าไฟล์แผนภาพเวลา TDTP สกุลเอกซ์เอ็มแอลไปจนถึงในขั้นตอนได้รับไฟล์ผลลัพธ์ที่เป็นไฟล์ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟในรูปแบบสกุลเอกซ์เอ็มแอล รายละเอียดของหน้าจอต่าง ๆ มีดังตอนไปนี้

1) หน้าจอหลักของเครื่องมือ “Timing Diagram into Timed Automata Converter” ในการรันโปรแกรมและเข้าเว็บบราวเซอร์ ต้องเข้าด้วยอินเทอร์เน็ตเท่านั้นถึงจะเข้าเครื่องมือ “Timing Diagram into Timed Automata Converter” ได้



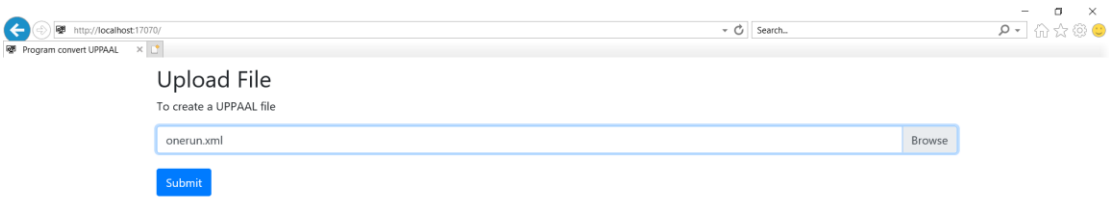
รูปที่ 4-15 หน้าจอหลักของเครื่องมือ “Timing Diagram into Timed Automata Converter”การแปลง

2) หน้าจอการเลือกนำเข้าไฟล์แผนภาพเวลา TDTP สกุลเอกซ์เอ็มแอล หน้าจอการเลือกนำเข้าไฟล์แผนภาพเวลา TDTP จะต้องเป็นสกุลเอกซ์เอ็มแอลเท่านั้นและเป็นโครงสร้างที่ผู้วิจัยได้ออกแบบเอกซ์เอ็มแอลไว้



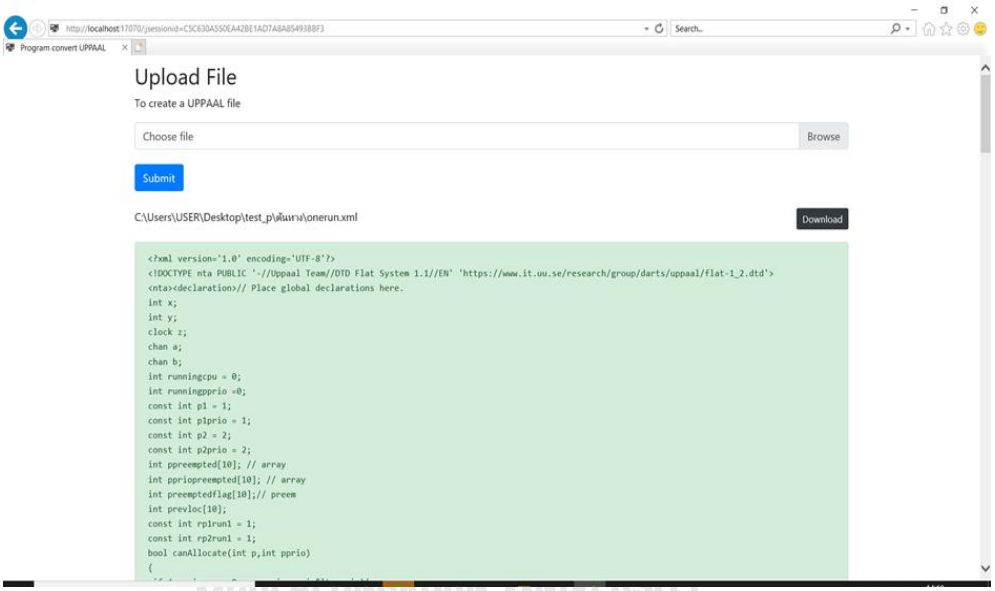
รูปที่ 4-16 หน้าจอการเลือกนำเข้าไฟล์แผนภาพเวลา TDTP สกุลเอกซ์เอ็มแอล

3) หน้าจอชื่อเอกสารนำเข้า เมื่อนำเข้าเอกสารไฟล์แผนภาพเวลา TDTP จะแสดงชื่อไฟล์ที่เลือกเพื่อดูความถูกต้องของชื่อไฟล์



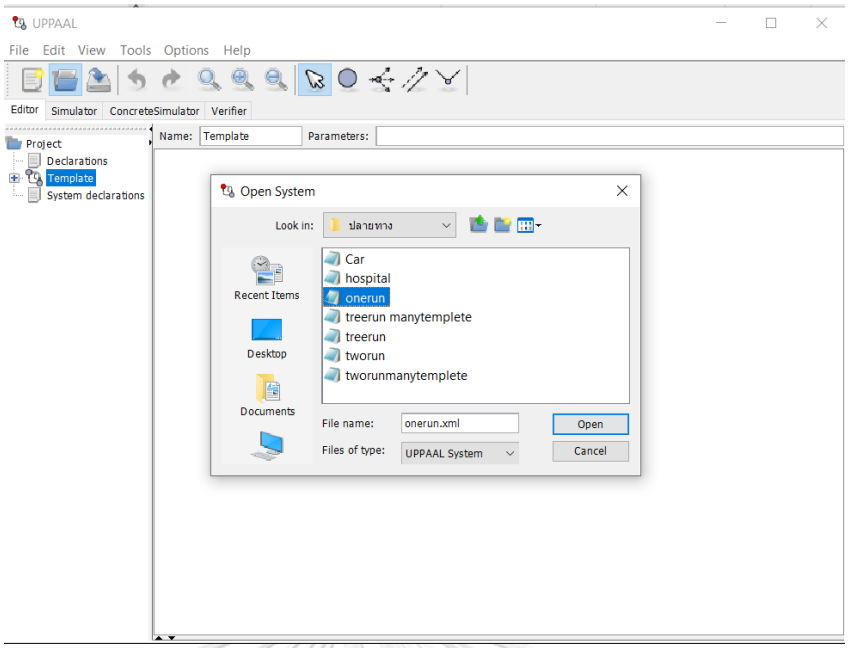
รูปที่ 4-17 หน้าชื่อจอเอกสารนำเข้า

4) หน้าจอการแปลงข้อมูลนำเข้า ผู้ใช้กดปุ่ม “Sumit” ถ้าสำเร็จจะขึ้นข้อความดังกล่าว ถ้าไม่สำเร็จจะแจ้งเตือนข้อผิดพลาด ถ้าสำเร็จก็ทำการกดปุ่ม “Download” ได้เลย



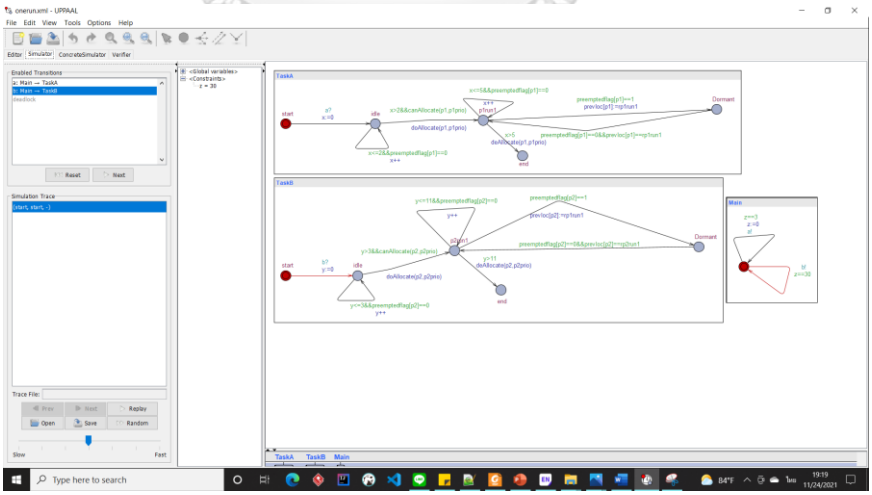
รูปที่ 4-18 หน้าจอการแปลงข้อมูลนำเข้า

5) หน้าจอนำไฟล์ที่แปลงเข้าเครื่องมือทวนสอบ UPPAAL ผู้ใช้เลือกไฟล์นำเข้าที่แปลงจากเครื่องมือ “Timing Diagram into Timed Automata Converter”แล้วหน้าจอการเรียกใช้เครื่องมือทวนสอบ UPPAAL



รูปที่ 4-19 หน้าจอนำไฟล์ที่แปลงเข้าเครื่องมือทวนสอบ UPPAAL

6) หน้าจอการแบบจำลองที่เข้าเครื่องมือ UPPAAL แสดงหน้าจอเพื่อเรียกใช้งานเครื่องมือสำหรับทวนสอบไทมด์ออโตมาตา UPPAAL หลังจากที่แปลงไฟล์แผนภาพเวลาสำเร็จแล้ว ซึ่งมีรายละเอียดของหน้าจอการใช้งาน ดังรูปที่ 4.20



รูปที่ 4-20 หน้าจอการแบบจำลองที่เข้าเครื่องมือ UPPAAL

บทที่ 5
การทดสอบและประเมินผล

การทดสอบเครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟนั้น จะเป็นการทดสอบผ่านเครื่องมือที่ผู้วิจัยได้ออกแบบไว้เพื่อตรวจสอบว่าเครื่องมือที่ผู้วิจัยได้พัฒนาขึ้นสามารถทำการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟได้อย่างถูกต้องหรือไม่ ผู้วิจัยได้ใช้ตัวอย่างกรณีศึกษามาทดสอบทั้งหมด 3 กรณีศึกษาเพื่อวิเคราะห์ผลทดสอบ สรุปผลการทดลองในการใช้เครื่องมือที่ผู้วิจัยออกแบบไว้นี้ รวมถึงเป็นแนวทางการปรับปรุงวิธีการแปลงและกฎที่ผู้วิจัยได้สร้างขึ้นมาอีกด้วย

5.1 สภาพแวดล้อมที่ใช้ในการทดสอบ

สภาพแวดล้อมที่ใช้ในการทดสอบ สำหรับการทดสอบเครื่องมือที่ผู้วิจัยออกแบบไว้ ผู้วิจัยเลือกใช้คอมพิวเตอร์และโปรแกรมประยุกต์ โดยมีรายละเอียดดังนี้

5.1.1 ฮาร์ดแวร์ (Hardware)

1) เครื่องคอมพิวเตอร์โน้ตบุ๊ก ชนิดระบบ 64 บิต หน่วยประมวลผลอินเทลคอร์ไอ 7 เจนเนอร์เรชั่น 8 2.00 กิกะเฮิร์ท (Intel Core i7 8th Generation 2.00 GHZ)

2) หน่วยความจำสำรอง ดีดีอาร์ 4 ขนาด 16.0 กิกะไบต์ (RAM DDR4 16.0 GB)

3) ฮาร์ดดิสก์ (Hard disk) ความจุ 250 กิกะไบต์ (250 GB) และ SSD 915 กิกะไบต์ (915 GB)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.1.2 ซอฟต์แวร์ (Software)

1) ระบบปฏิบัติการ Windows 10 Pro

2) โปรแกรม Notepad ++ เวอร์ชัน 8.1.9.1

3) เครื่องมือ UPPAAL เวอร์ชัน 4.1

4) โปรแกรม Visual Paradigm เวอร์ชัน 16.3

5) โปรแกรม IntelliJ IDEA เวอร์ชัน 2021.1.2 (Community Edition)

6) ภาษาที่ใช้ในการพัฒนาเครื่องมือเป็นภาษาจาวา

5.2 กรณีการทดสอบเครื่องมือการแปลง

การทดสอบเครื่องมือการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟนั้นมีการทดสอบดังต่อไปนี้

1) ผู้ใช้เริ่มต้นจากการสร้างไฟล์แผนภาพเวลาต้นทางให้อยู่ในรูปไฟล์นามสกุลเอกซ์เอ็มแอล (.xml)

2) ผู้ใช้ทำเครื่องมืออัตโนมัติทำการจับคู่กฎการแปลงและสกัดข้อมูลส่วนประกอบของกระบวนการ และแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟได้อย่างถูกต้อง นั่นคือผู้ใช้นำไฟล์ต้นทางจากข้อ 1 มาเข้าเครื่องมืออัตโนมัติที่ผู้วิจัยสร้างขึ้น และผู้วิจัยนำไฟล์ผลลัพธ์ไทมด์ออกโตมาตาในรูปไฟล์นามสกุลเอกซ์เอ็มแอล นำไฟล์ผลลัพธ์ไทมด์ออกโตมาตามาเข้าเครื่องมือ UPPAAL มาทวนสอบบนเครื่องมือ UPPAAL เพื่อดูพฤติกรรมของกรณีศึกษาทั้ง 3 กรณีศึกษามาทดสอบว่าเป็นไปตามผลลัพธ์หรือไม่

3) ทำการทวนสอบผลลัพธ์ที่ได้จากการแปลงด้วยเครื่องมือ UPPAAL

5.3 กรณีทดสอบ

ผู้วิจัยได้นิยามกฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟที่ได้ออกแบบไว้ในบทที่ 3 แต่ละกรณีสามารถแสดงข้อมูลรูปแบบการใช้กฎที่สามารถอ้างอิงไว้ในบทที่ 3 ผู้วิจัยจะแสดงรายละเอียดของกฎการแปลงของแต่ละกรณีการทดสอบ 1-3 กรณีการทดสอบ (CaseStudy01- CaseStudy03) กรณีทดสอบ 3 กรณี ได้แก่

- เมื่อมีรถฉุกเฉินมา (CaseStudy01)
- ณ โรงพยาบาลแห่งหนึ่งที่มีการจัดลำดับความสำคัญ (CaseStudy02)
- ลำดับความสำคัญของงานบำรุงรักษา (CaseStudy03)

ตารางที่ 5-1 รายการกรณีทดสอบและการใช้งานกฎของกรณีทดสอบ CASE01-CASE03

กรณีศึกษา	รายการการใช้กฎการแปลง						
	Rule1	Rule2	Rule3	Rule4	Rule5	Rule6	Rule7
CaseStudy01	●	●	●	●	●	●	●
CaseStudy02	●	●	●	●	●	●	●
CaseStudy03	●	●	●	●	●	●	●

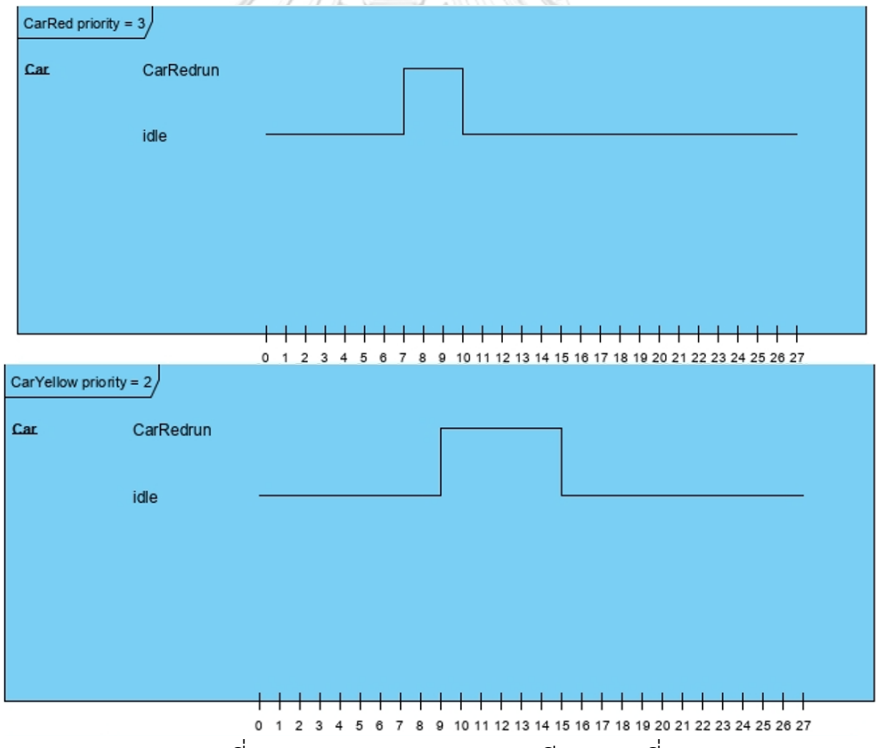
เพื่อให้สามารถประเมินผลของการออกแบบกฎการแปลงแผนภาพเวลาไปเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ ผ่านเครื่องมือที่ชื่อว่า “Timing Diagram into Timed Automata Converter” ที่ได้พัฒนาไว้ ผู้วิจัยจะแสดงกรณีทดสอบโดยมีรายละเอียดของแต่ละกรณีทดสอบดังต่อไปนี้

5.4 การทดสอบเครื่องมือการแปลงโดยใช้กรณีศึกษา

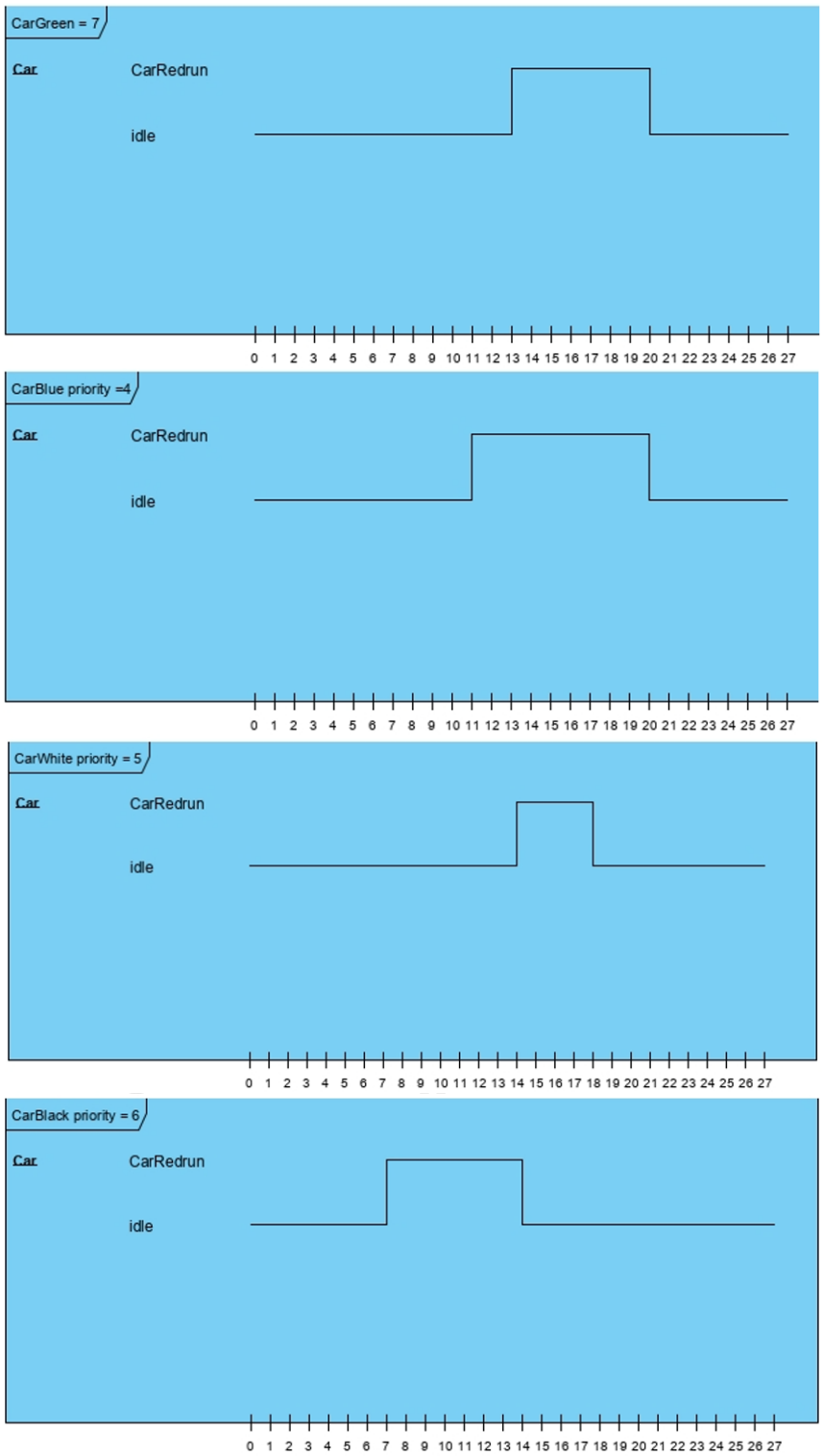
5.4.1 กรณีทดสอบที่ 1 เมื่อมีรถฉุกเฉินมา (CASE01)

ตัวอย่างกรณีทดสอบของแผนภาพเวลาในกรณีที่ 1 จะแบ่งขั้นตอนการทำงานที่ 1-7 จากรายละเอียดของกรณีทดสอบ CASE01 สามารถออกแบบกระบวนการแผนภาพเวลาได้ดังรูปที่ 5.1 และแปลงเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟโดยเครื่องมือการแปลงที่พัฒนาขึ้นมาได้ดังรูปที่ 5-2 ตัวอย่างกรณีทดสอบที่ 1 มีรถอยู่ 7 คัน รถสีแดง (Red Car), สีเหลือง (Yellow Car), สีเขียว (Green Car), สีน้ำเงิน (Blue Car), สีขาว (White Car), สีดำ (Black Car), รถพยาบาลฉุกเฉิน (Emergency Car) เมื่อรถสีต่าง ๆ เคลื่อนที่บนถนนตามปกติแต่มีรถพยาบาลฉุกเฉินมาแทรกเพื่อที่ไปส่งคนไข้ที่ได้รับบาดเจ็บสาหัสให้ทันเวลาทำให้รถสีต่าง ๆ ต้องให้ทาง ก่อนที่รถพยาบาลฉุกเฉินขับผ่านพ้นไปแล้วรถสีต่าง ๆ จะขับเคลื่อนไปอย่างปกติ

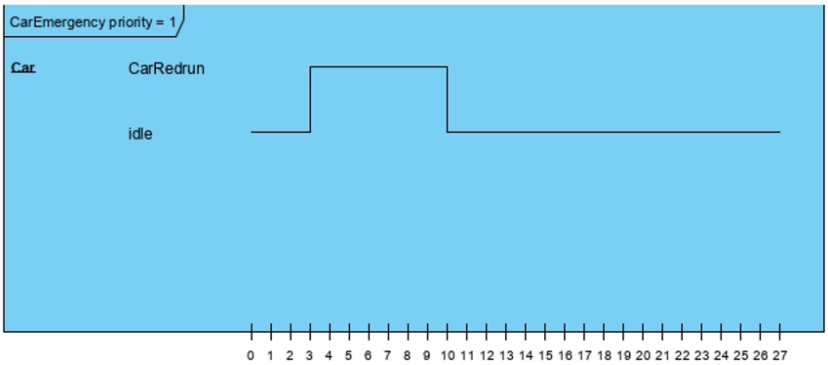
5.4.1.1 แผนภาพเวลากรณีทดสอบที่ 1



รูปที่ 5-1 แผนภาพเวลาของกรณีทดสอบที่ 1



รูปที่ 5-1 แผนภาพเวลาของกรณีทดสอบที่ 1 (ต่อ)



รูปที่ 5-1 แผนภาพเวลาของกรณีทดสอบที่ 1 (ต่อ)

5.4.1.2 ตารางแสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ

ตารางที่ 6-2 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ [13]

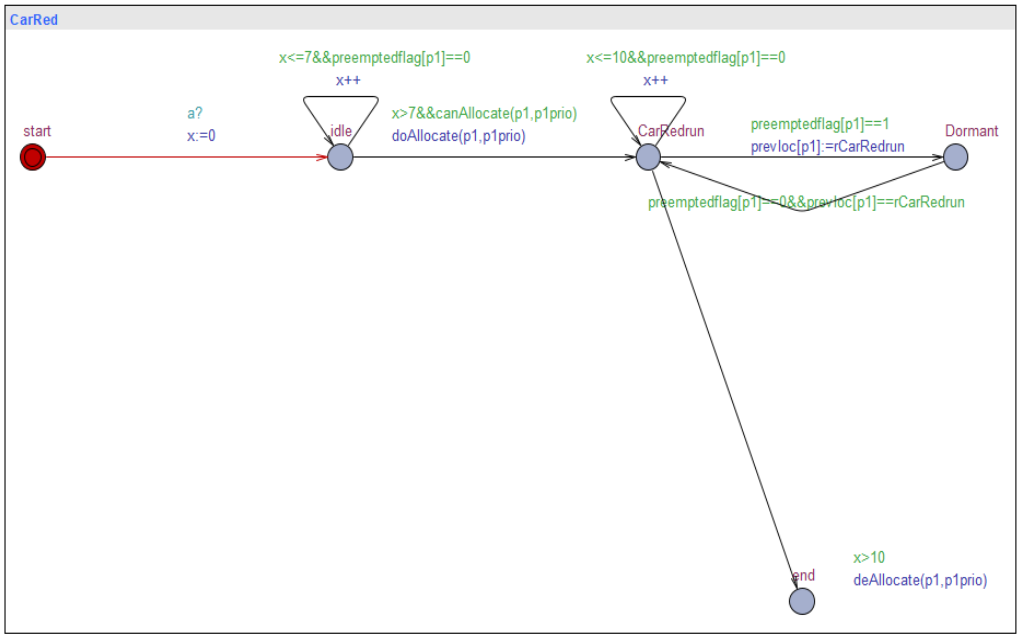
Process Id	Process Name	Priority	State Name	Begin	End
P1	Red Car	3	idle	0	7
			CarRedrun	7	10
			end	10	16
P2	Yellow Car	2	idle	0	9
			CarYellowrun	9	15
			end	15	20
P3	Green Car	7	idle	0	11
			CarGreenrun	11	18
			end	18	25
P4	Blue Car	4	idle	0	13
			CarBluerun	13	20
			end	20	30
P5	White Car	5	idle	0	16
			CarWhiterun	16	18
			end	18	25
P6	Black Car	6	idle	0	8
			CarBlackrun	8	14

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 5-2 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ (ต่อ)

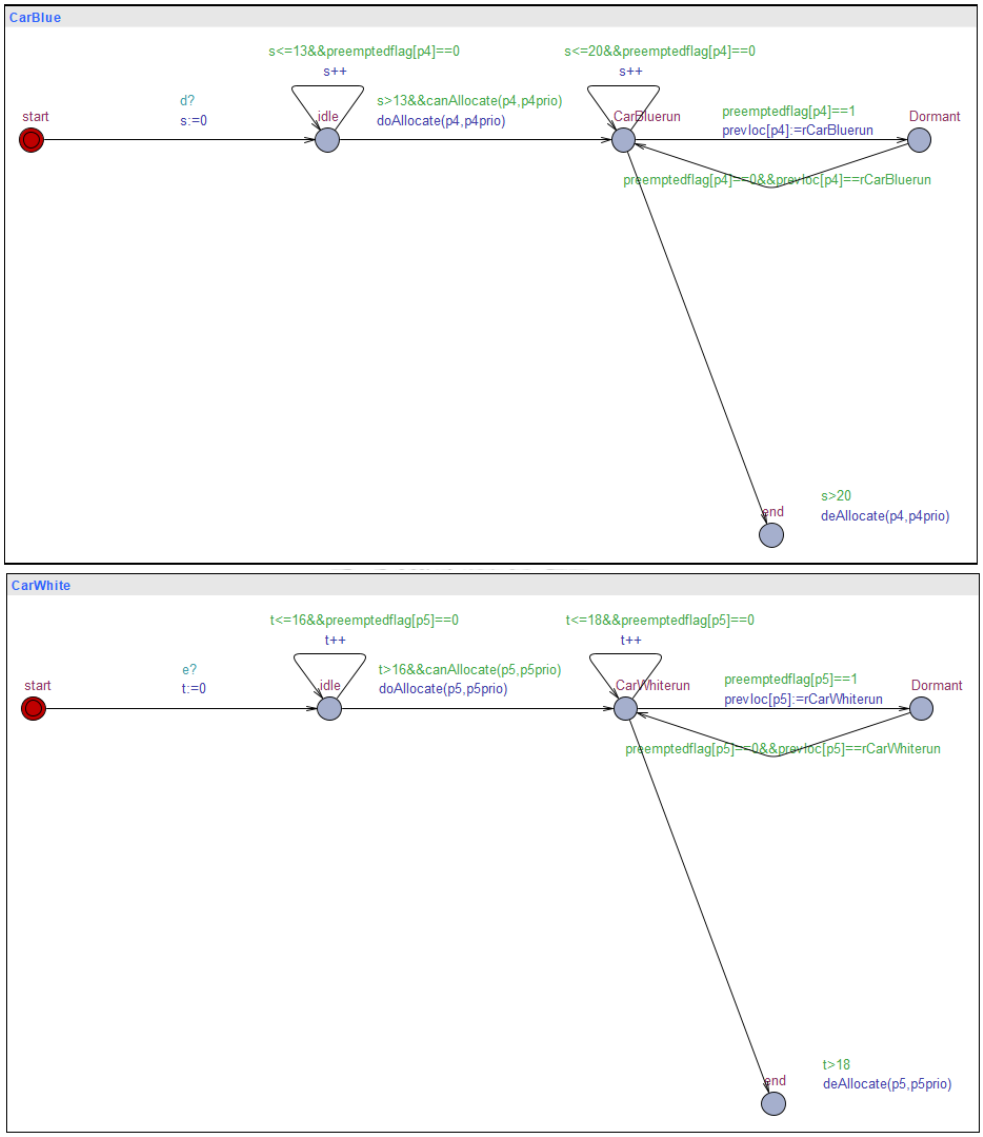
Process Id	Process Name	Priority	State Name	Begin	End
			end	14	20
P7	Emergency Car	1	idle	0	3
			Emergencycarrun	3	10
			end	10	15

ในตารางที่ 5-2 ประกอบด้วย 7 กระบวนการ ได้แก่ P1, P2, P3, P4, P5, P6 และ P7 ลำดับความสำคัญ โดยระบบจะกำหนดค่าที่น้อยที่สุดแสดงถึงลำดับความสำคัญมากที่สุดและให้ตัวเลขที่มีค่ามากที่สุดแสดงถึงลำดับความสำคัญน้อยที่สุด จะได้ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ

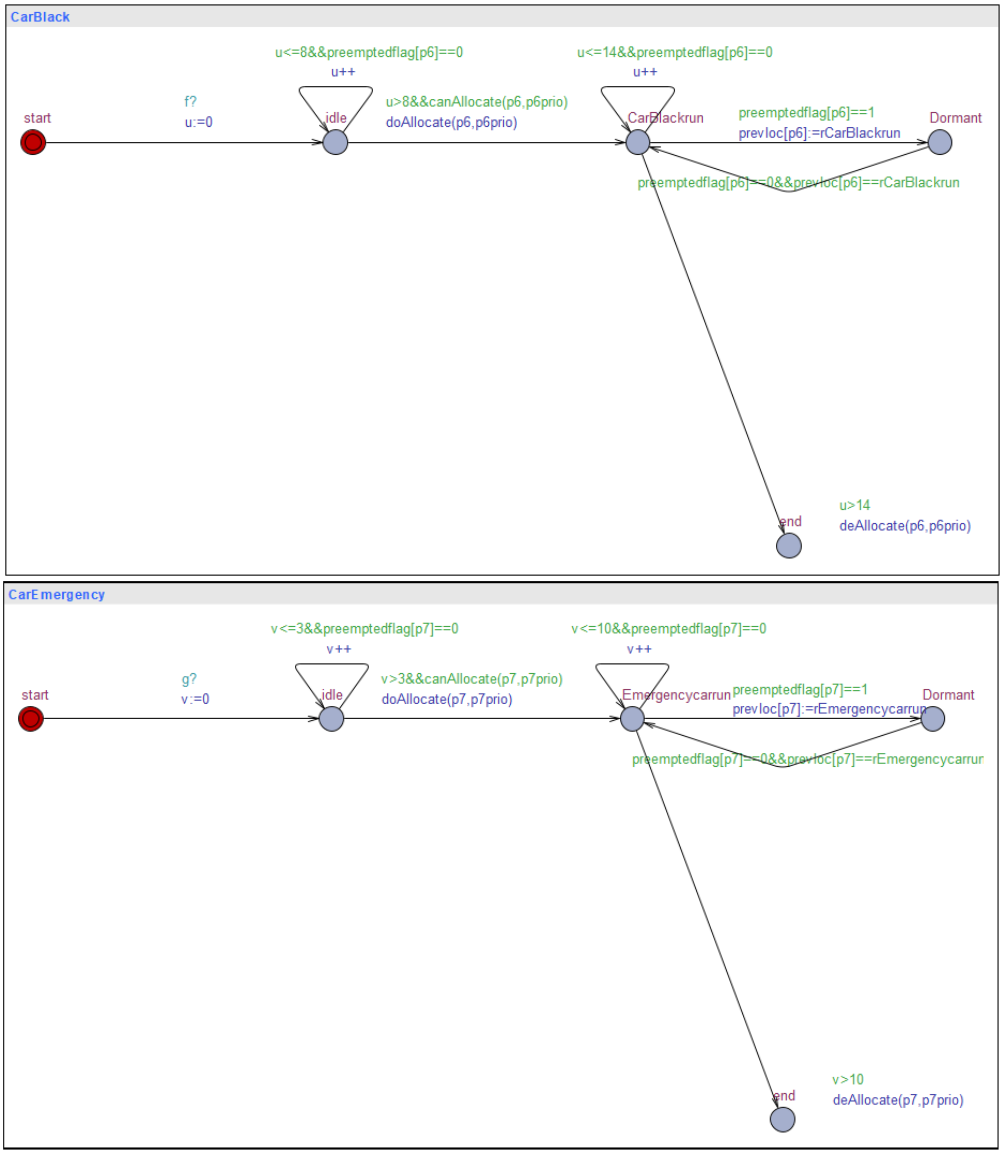


รูปที่ 5-2 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 1

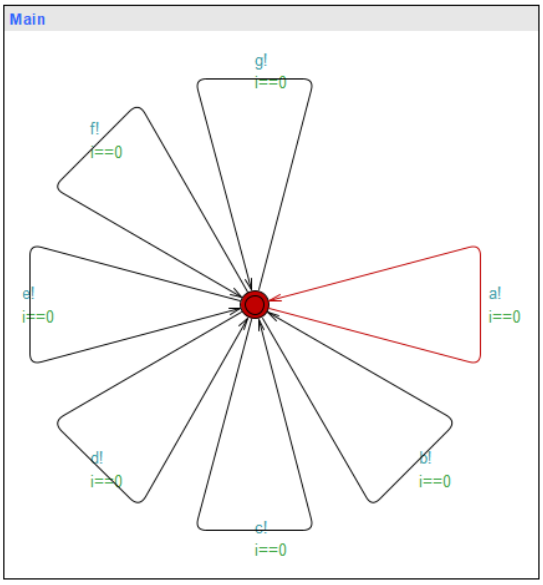
รูปที่ 5-2 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 1 (ต่อ)



รูปที่ 5-2 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 1 (ต่อ)



รูปที่ 5-2 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 1 (ต่อ)



รูปที่ 5-2 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 1 (ต่อ)

ตัวอย่างการทดสอบคุณสมบัติของแบบจำลองของกรณีทดสอบที่ 1 สามารถทวนสอบด้วยภาษา TCTL ของแต่ละข้อได้ดังรูป 5-3 เมื่อสัญลักษณ์ไฟสีแดงหมายความว่า คุณสมบัติไม่พึงพอใจ (Property is not satisfied) และสัญลักษณ์ไฟสีเขียวหมายความว่า คุณสมบัติพึงพอใจ (Property is satisfied) ตัวอย่างการทวนสอบภาษา TCTL ดังนี้

1) E<> deadlock หมายความว่า มีบางเส้นทางในที่สุดแล้วมีคุณสมบัติภาวะติดตาย จะได้ว่า Property is satisfied

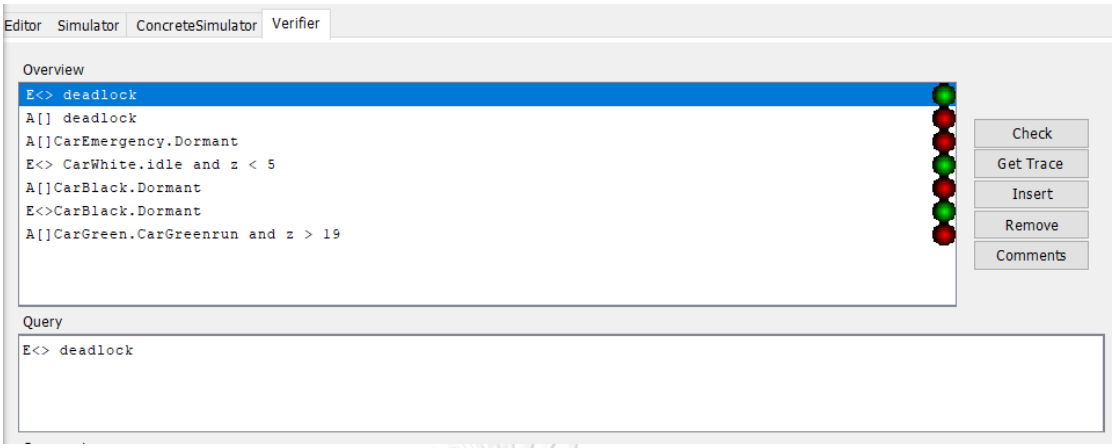
2) A[]CarEmergency.Dormant หมายความว่า มีทุกเส้นทางที่รถฉุกเฉินอยู่สถานะ “Dormant” เป็นจริงตลอด จะได้ว่า Property is not satisfied

3) A[]CarBlack.Dormant หมายความว่า มีทุกเส้นทางที่รถสีดำอยู่สถานะ “Dormant” เป็นจริงตลอด จะได้ว่า Property is not satisfied

4) A[]CarGreen.CarGreenrun and z > 19 หมายความว่า มีทุกเส้นทางที่รถสีเขียวอยู่สถานะ “CarGreenrun” และตัวแปรเวลา z มีค่ามากกว่า 19 เป็นจริงตลอด จะได้ว่า Property is not satisfied

5) A[]CarEmergency.Dormant หมายความว่า มีทุกเส้นทางที่รถฉุกเฉินอยู่สถานะ “Dormant” เป็นจริงตลอด จะได้ว่า Property is not satisfied

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



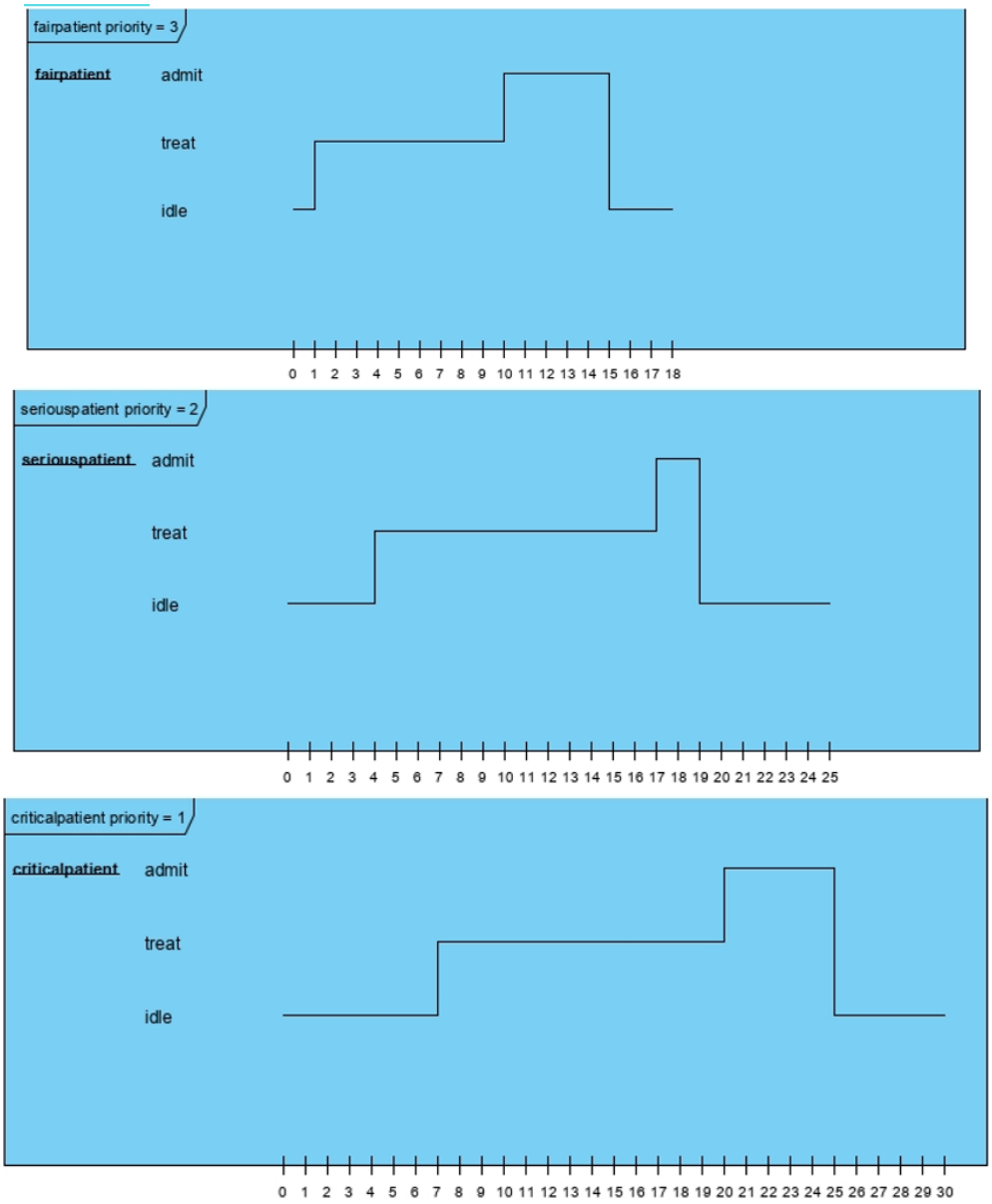
รูปที่ 5-3 ผลการทวนสอบจากเครื่องมือทวนสอบ UPPAAL ของกรณีศึกษาที่ 1

5.4.2 กรณีทดสอบที่ 2 ณ โรงพยาบาลแห่งหนึ่งที่มีการจัดลำดับความสำคัญ (CASE02)

ตัวอย่างกรณีทดสอบของแผนภาพเวลาในกรณีที่ 1 จะแบ่งขั้นตอนการทำงานที่ 1-3 จากรายละเอียดของกรณีทดสอบ CASE02 สามารถออกแบบกระบวนการแผนภาพเวลาได้ดังรูปที่ 5-4 และแปลงเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟโดยเครื่องมือการแปลงที่พัฒนาขึ้นมาได้ดังรูปที่ 5-5 ตัวอย่างกรณีทดสอบที่ 2 ณ โรงพยาบาลแห่งหนึ่ง มีผู้ป่วย 3 ประเภท ได้แก่ ผู้ป่วยวิกฤต (Critical Patient), ผู้ป่วยธรรมดา (Fair Patient) เช่น ไอ จาม, ผู้ป่วยหนัก (Serious Patient)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.4.2.1 แผนภาพเวลากรณีทดสอบที่ 2



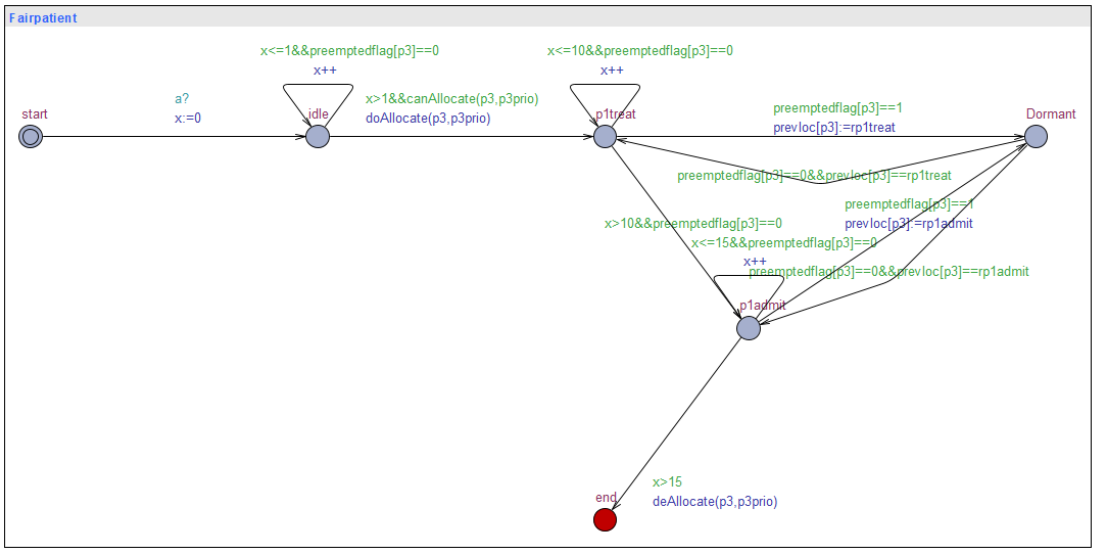
รูปที่ 5-4 แผนภาพเวลาของกรณีทดสอบที่ 2

5.4.2.2 ตารางแสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ

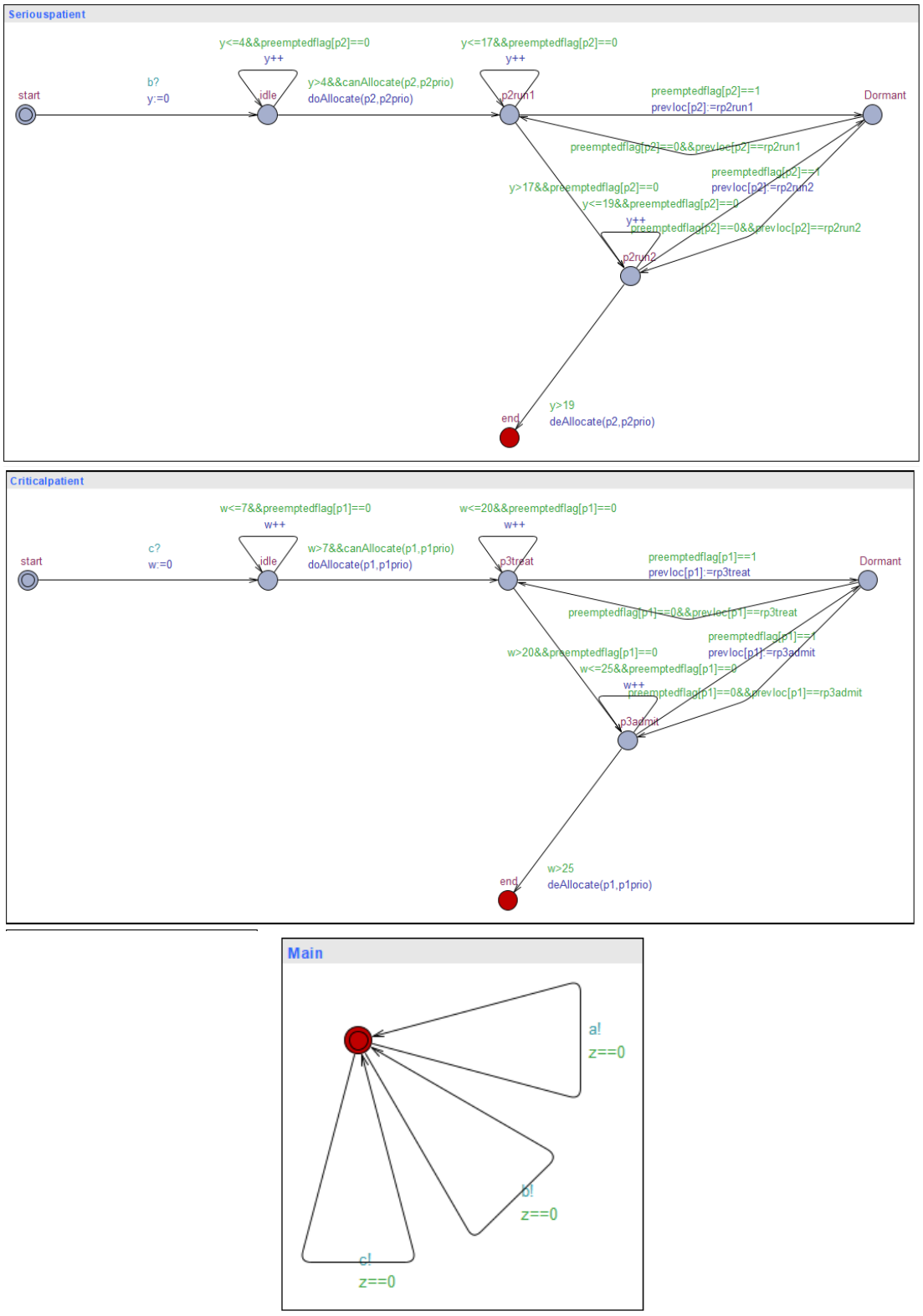
ตารางที่ 5-3 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ

Process Id	Process Name	Priority	State Name	Begin	End
P1	Critical Patient	1	idle	0	7
			treat	7	20
			admit	20	25
			end	25	30
P3	Fair Patient	3	idle	0	1
			treat	1	10
			admit	10	15
			end	15	18
P2	Serious Patient	2	idle	0	4
			treat	4	17
			admit	17	19
			end	19	25

ในตารางที่ 5-3 ประกอบด้วย 3 กระบวนการ ได้แก่ P1, P2, P3 ลำดับความสำคัญ จะได้ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟ



รูปที่ 5-5 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 2



รูปที่ 5-5 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 2 (ต่อ)

ตัวอย่างการทดสอบคุณสมบัติของแบบจำลองของกรณีทดสอบที่ 2 สามารถทวนสอบด้วยภาษา TCTL ของแต่ละข้อได้ดังรูป 5-6 เมื่อสัญลักษณ์ไฟสีแดงหมายความว่า คุณสมบัติไม่พึ่งพอใจ และสัญลักษณ์ไฟสีเขียวหมายความว่า คุณสมบัติพึ่งพอใจ ตัวอย่างการทวนสอบภาษา TCTL ดังนี้

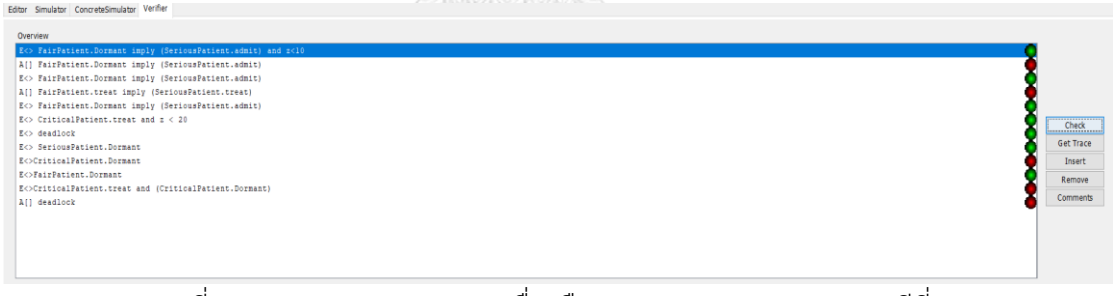
1) E<> FairPatient.Dormant imply (SeriousPatient.admit) and z<10 หมายความว่า ถ้ามีบางเส้นทางที่ผู้ป่วยประเภทธรรมดาในที่สุดแล้วอยู่สถานะ “Dormant” แล้วผู้ป่วยหนักในที่สุดแล้ว อยู่สถานะ “admit”และตัวแปรเวลา z มีค่าน้อยกว่า 10 จะได้ว่า Property is satisfied

2) E<> CriticalPatient.treat and z < 20 หมายความว่า มีบางเส้นทางที่ผู้ป่วยวิกฤตในที่สุดแล้วอยู่สถานะ “treat” และตัวแปรเวลา z มีค่าน้อยกว่า 20 จะได้ว่า Property is satisfied

3) A[] FairPatient.treat imply (SeriousPatient.treat) หมายความว่า ถ้ามีทุกเส้นทางผู้ป่วยประเภทธรรมดาอยู่สถานะ “treat” แล้วผู้ป่วยวิกฤตอยู่สถานะ “treat”แล้วเป็นจริงตลอดจะได้ว่า Property is not satisfied

4) E<>FairPatient.Dormant หมายความว่า มีบางเส้นทางที่ผู้ป่วยประเภทธรรมดาในที่สุดแล้วเกิดสถานะ “Dormant”จะได้ว่า Property is satisfied

5) A[] deadlock หมายความว่า ทุกเส้นทางมีคุณสมบัติภาวะติดตาย จะได้ว่า Property is not satisfied



รูปที่ 5-6 ผลการทวนสอบจากเครื่องมือทวนสอบ UPPAAL ของกรณีที่ 2

5.4.3 กรณีทดสอบที่ 3 ลำดับความสำคัญของงานบำรุงรักษา (CASE03)

ตัวอย่างกรณีทดสอบของแผนภาพเวลาในกรณีที่ 3 จะแบ่งขั้นตอนการทำงานที่ 1-5 จากรายละเอียดของกรณีทดสอบ CASE03 สามารถออกแบบกระบวนการแผนภาพเวลาได้ดังรูปที่ 5-7 และแปลงเป็นไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟโดยเครื่องมือการแปลงที่พัฒนาขึ้นมา ตัวอย่างกรณีทดสอบที่ 3 ลำดับความสำคัญของงานบำรุงรักษา (Priorities of Maintenance Work)

• งานฉุกเฉิน (Emergency Work) ควรเริ่มทำงานทันที ลักษณะงานที่มีผลทันทีต่อความปลอดภัย สิ่งแวดล้อม คุณภาพ

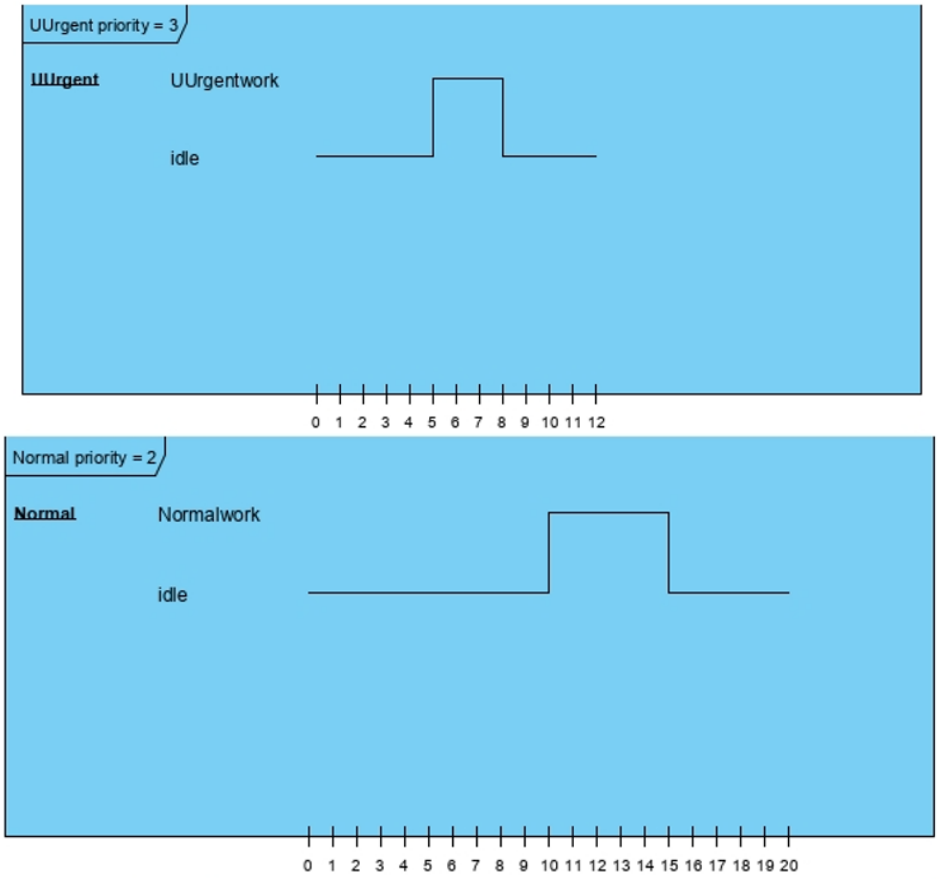
• งานด่วน (Urgent Work) ทำงานภายใน 24 ชั่วโมง ลักษณะงานที่มีแนวโน้มที่จะส่งผลกระทบต่อความปลอดภัย สิ่งแวดล้อม คุณภาพ

• ทำงานปกติ (Normal Work) ทำงานควรเริ่ม 48 ชั่วโมง ลักษณะงานที่มีแนวโน้มที่จะส่งผลกระทบต่อการผลิตภายใน 1 สัปดาห์

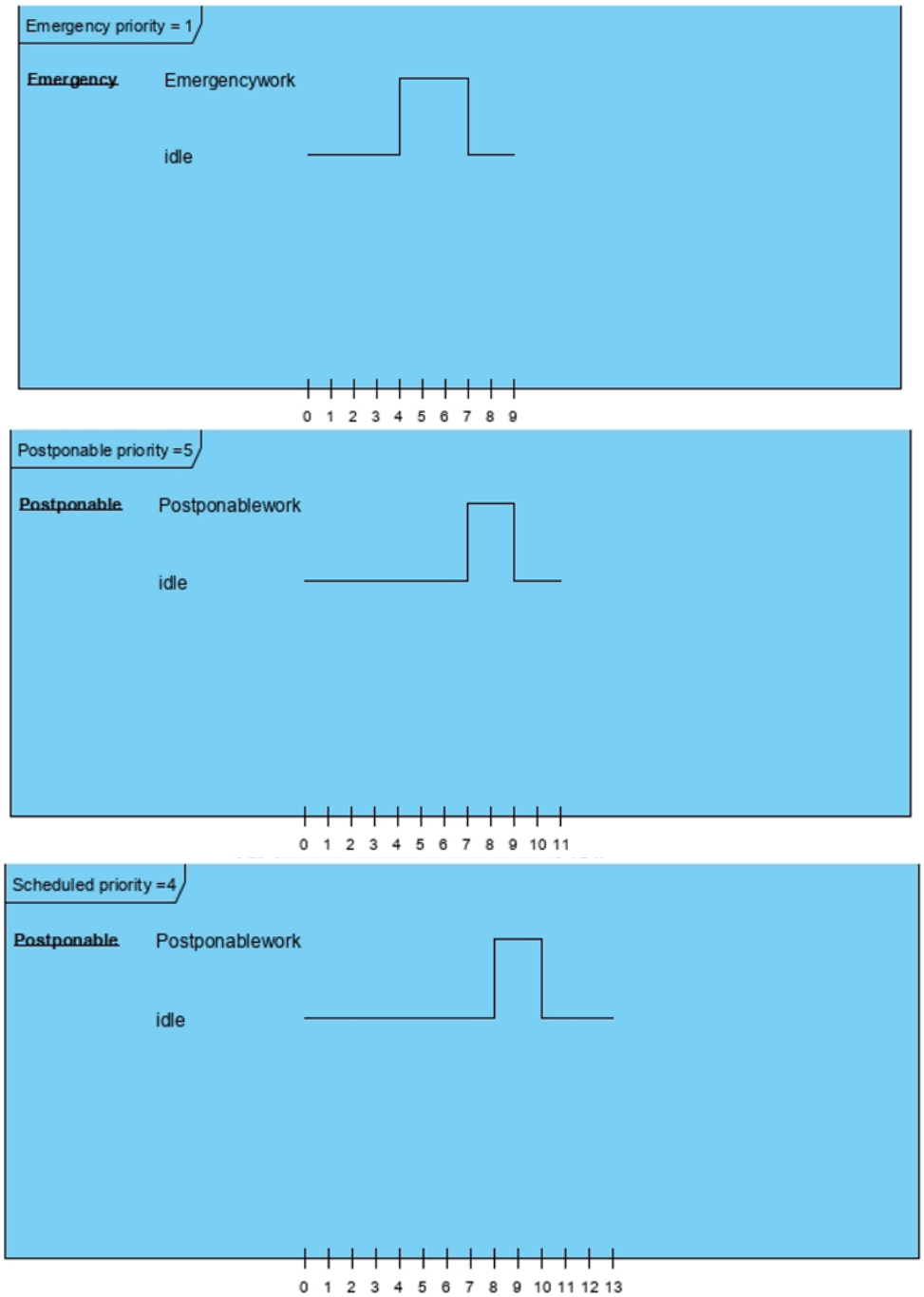
• กำหนดการ (Scheduled Work) ทำงานตามตารางเวลา ลักษณะงานการบำรุงรักษาและกิจวัตรประจำวันแบบป้องกัน งานทั้งหมดที่จัดตารางไว้

• เลื่อนได้ (Postponable Work) ลักษณะงานที่ไม่มีผลกระทบในทันทีต่อความปลอดภัย สุขภาพ สิ่งแวดล้อม หรือการดำเนินการผลิต

5.4.3.1 แผนภาพเวลากรณีทดสอบที่ 3



รูปที่ 5-7 แผนภาพเวลาของกรณีทดสอบที่ 3



รูปที่ 5-7 แผนภาพเวลาของกรณีทดสอบที่ 3 (ต่อ)

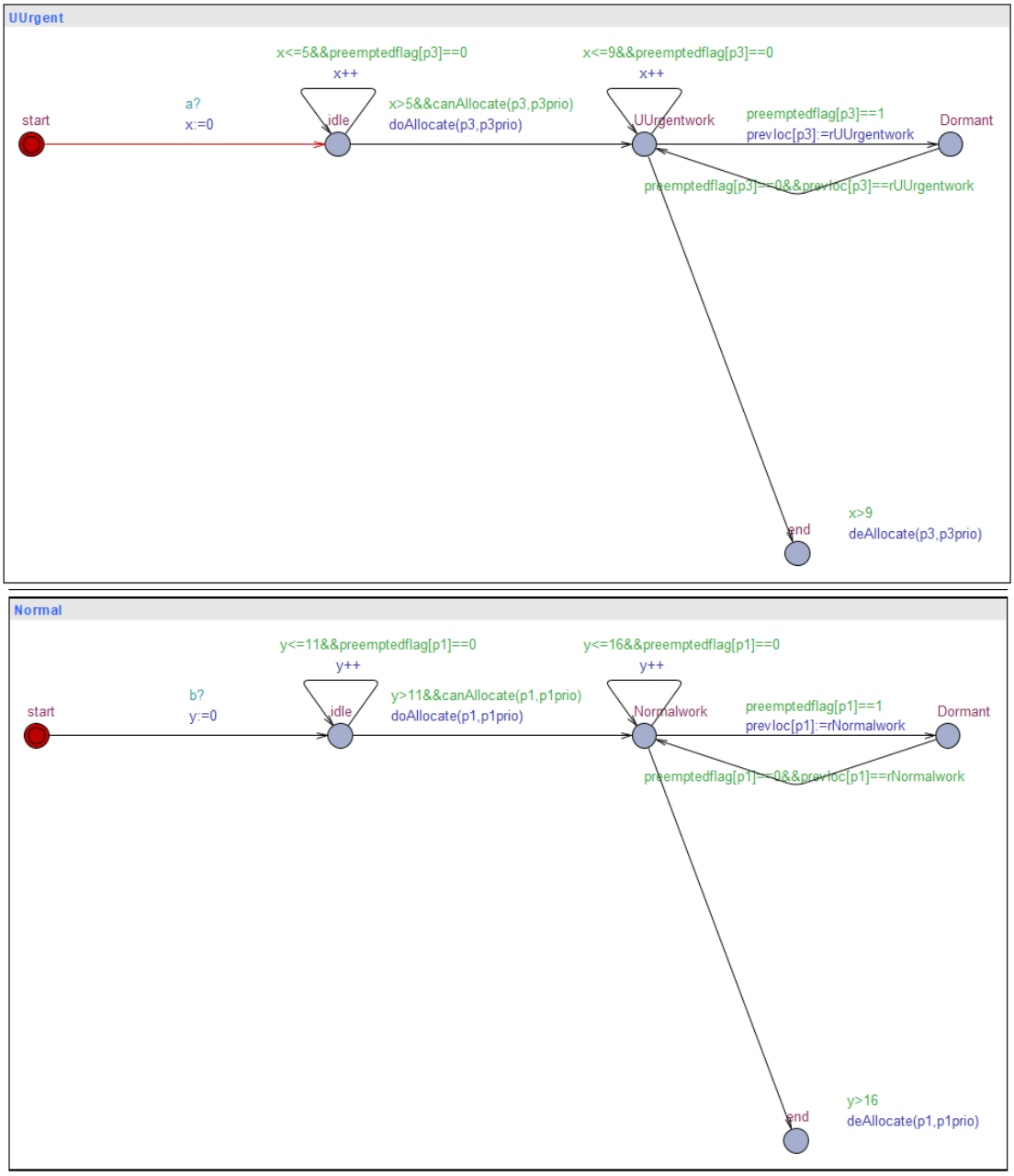
5.4.3.2 ตารางแสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ

ตารางที่ 5-4 แสดงลำดับของกระบวนการทำงานของซีพียูแบบพรีเอ็มทีฟ

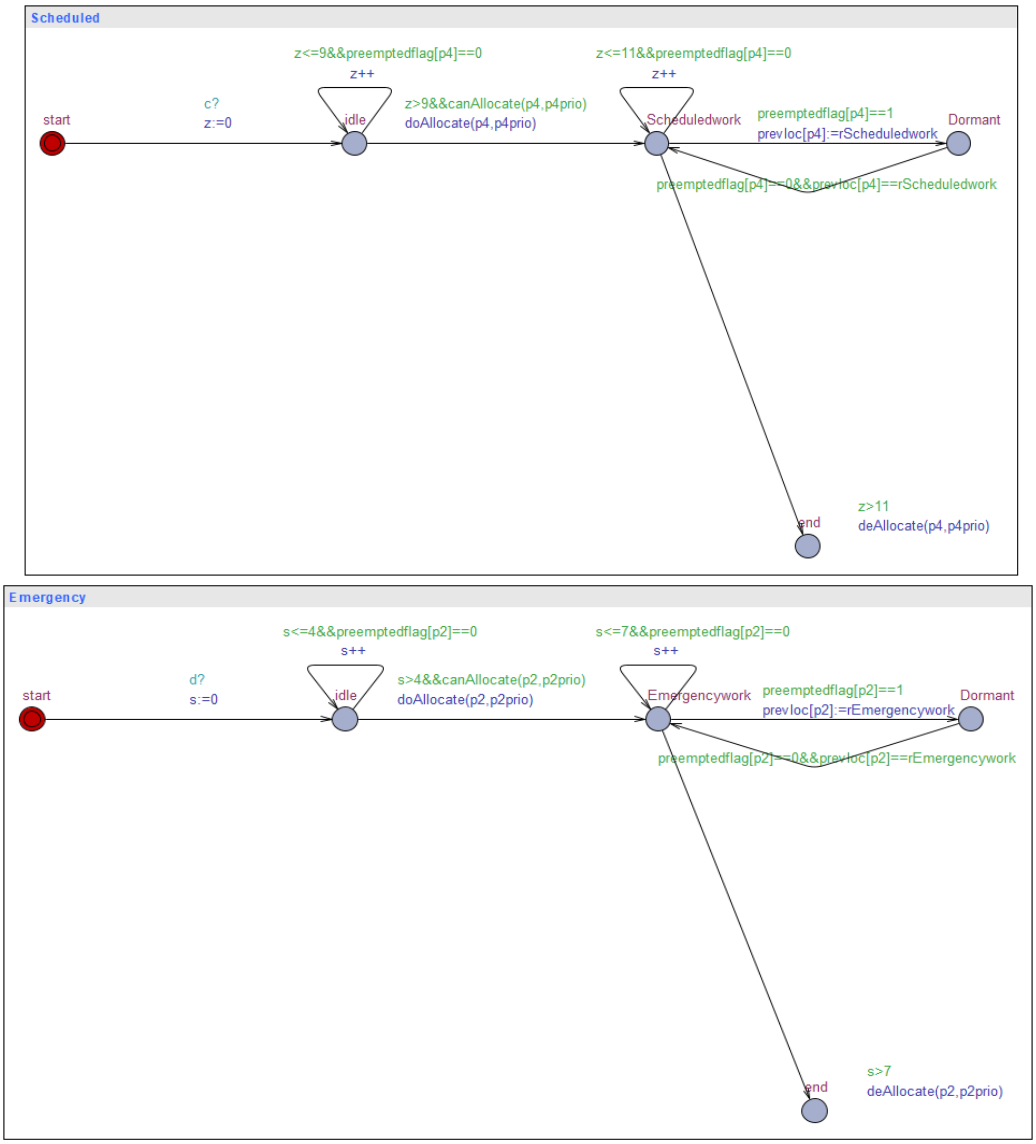
Process Id	Process	Priority	State Name	Begin	End
P2	Emergency	2	idle	0	4
			Emergency Work	4	7
			end	7	9
P3	UUrgent	3	idle	0	5
			Urgent	5	9
			end	9	12
P1	Normal	1	idle	0	11
			Normal Work	11	16
			end	16	20
P4	Scheduled	4	idle	0	9
			Scheduled Work	9	11
			end	11	13
P5	Postponable	5	idle	0	7
			Postponable Work	7	9
			end	9	11

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

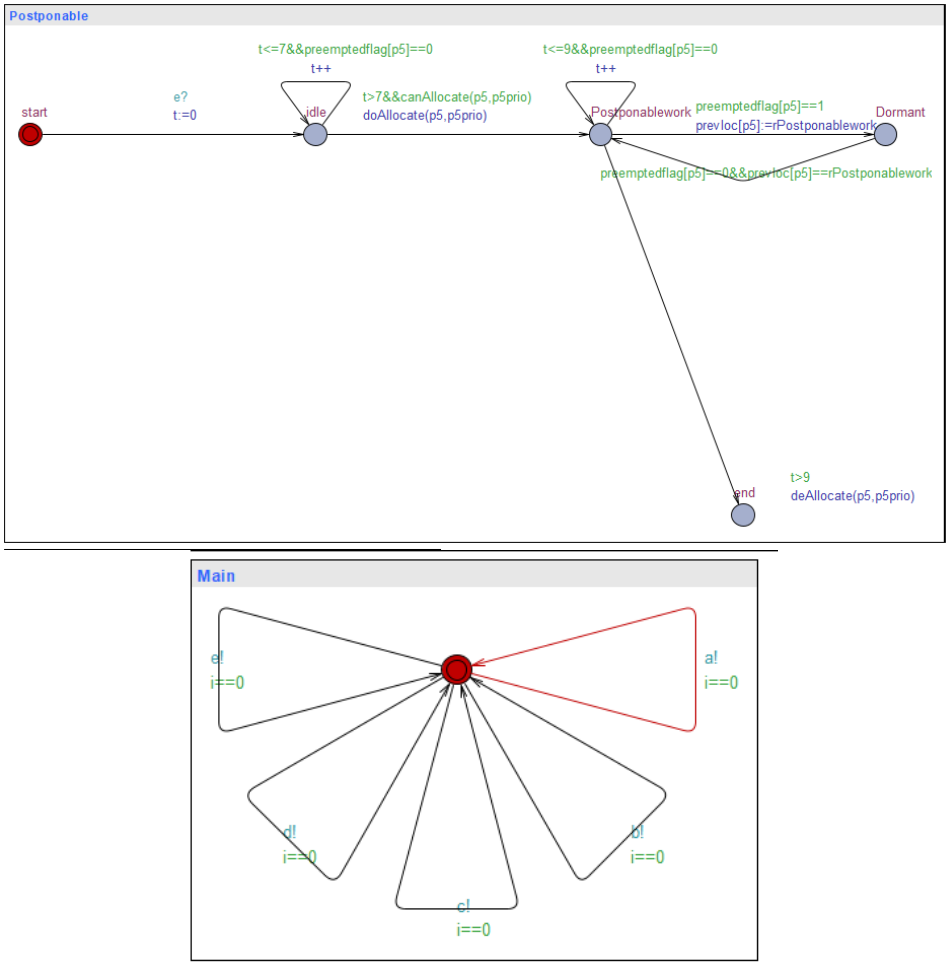
ในตารางที่ 5-4 ประกอบด้วย 5 กระบวนการ ได้แก่ P1, P2, P3, P4, P5 ลำดับความสำคัญโดยระบบจะกำหนดค่าที่น้อยที่สุดแสดงถึงลำดับความสำคัญมากที่สุดและให้ตัวเลขที่มีค่ามากที่สุดแสดงถึงลำดับความสำคัญน้อยที่สุด



รูปที่ 5-8 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 3



รูปที่ 5-8 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 3 (ต่อ)



รูปที่ 5-8 ไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟกรณีศึกษาที่ 3 (ต่อ)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตัวอย่างการทดสอบคุณสมบัติของแบบจำลองของกรณีทดสอบที่ 3 สามารถทวนสอบด้วยภาษา TCTL ของแต่ละข้อได้ดังรูป 5-9 เมื่อสัญลักษณ์ไฟสีแดงหมายความว่า คุณสมบัติไม่พึงพอใจ และสัญลักษณ์ไฟสีเขียวหมายความว่า คุณสมบัติพึงพอใจ ตัวอย่างการทวนสอบภาษา TCTL ดังนี้

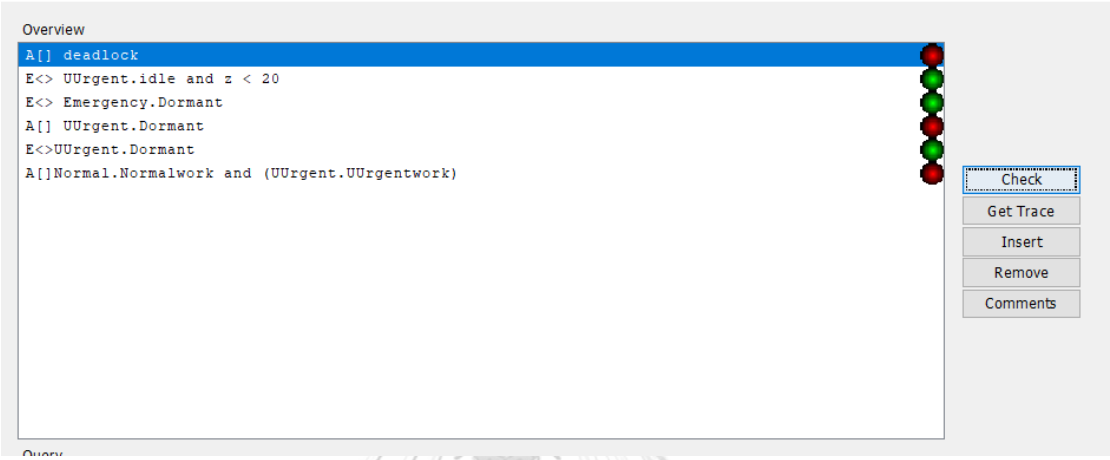
1) A[] deadlock หมายความว่า มีทุกเส้นทางที่สามารถมีคุณสมบัติภาวะติดตาย

2) E<> UUrgent.idle and z < 20 หมายความว่า มีบางเส้นทางที่เป็นงานด่วนในที่สุดแล้วอยู่สถานะ “idle” และตัวแปรเวลา z มีค่าน้อยกว่า 20 จะได้ว่า Property is satisfied

3) E<> Emergency.Dormant หมายความว่า มีบางเส้นทางที่เป็นงานฉุกเฉินในที่สุดแล้วอยู่สถานะ “Dormant”จะได้ว่า Property is satisfied

4) A[] UUrgent.Dormant หมายความว่า มีทุกเส้นทางที่เป็นงานฉุกเฉินอยู่สถานะ “Dormant” เป็นจริงตลอดทุกเส้น จะได้ว่า Property is not satisfied

5) E<>UUrgent.Dormant หมายความว่า มีบางเส้นทางที่เป็นงานฉุกเฉินในที่สุดแล้วอยู่สถานะ “Dormant”จะได้ว่า Property is satisfied



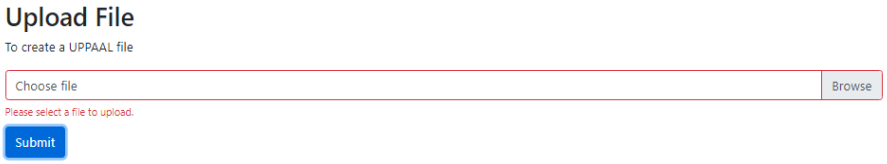
รูปที่ 5-9 ผลการทวนสอบจากเครื่องมือทวนสอบ UPPAAL ของกรณีที่ 3

5.5 กรณีทดสอบหน้าเว็บเครื่องมือการแปลงอัตโนมัติที่ผู้วิจัยสร้างขึ้น “Timing Diagram into Timed Automata Converter”

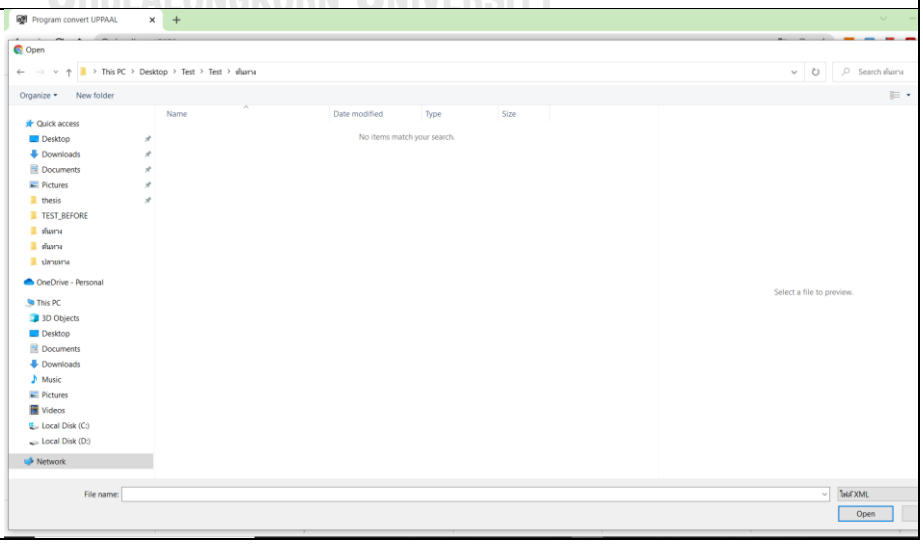
ผู้วิจัยได้ทำการดักข้อผิดพลาดที่เกิดขึ้นที่เป็นไปได้ กรณีทดสอบหน้าเว็บเครื่องมือการแปลงอัตโนมัติบนหน้าเว็บ ตัวอย่างเช่น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

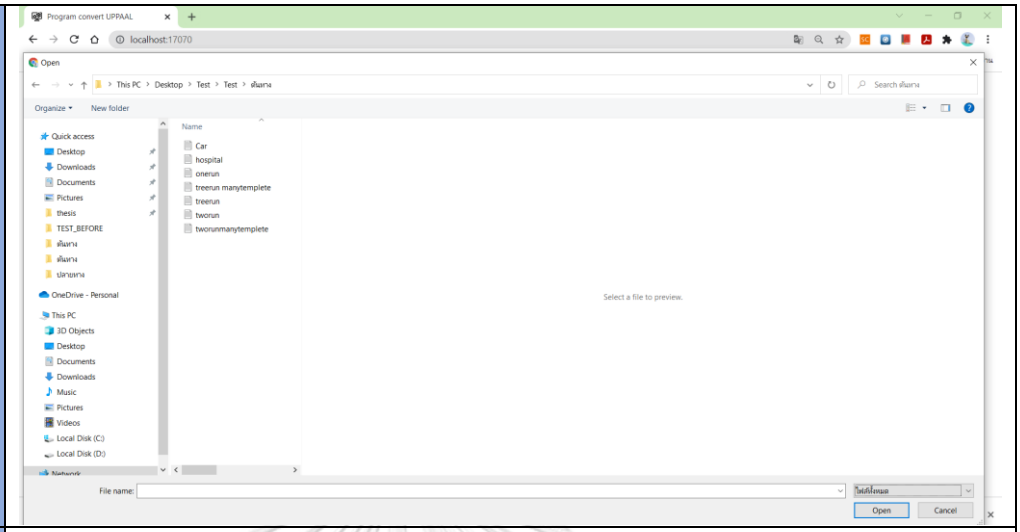
ตารางที่ 5-4 ทดสอบกรณีไม่ใส่ไฟล์

Test Case ID	TestCase01
Precondition: (screen)	
Inputs	ทดสอบโดยไม่ใส่ไฟล์
Postcondition: (screen)	
หมายเหตุ	จะแจ้งเตือนว่า “Please select a file to upload.”

ตารางที่ 5-5 ทดสอบกรณีไม่ใส่ไฟล์ที่มีนามสกุลเอกซ์เอ็มแอลต้นทาง

Test Case ID	TestCase03
Precondition: (screen)	
Inputs	เมื่อผู้ใช้คลิก Browse
Postcondition: (screen)	

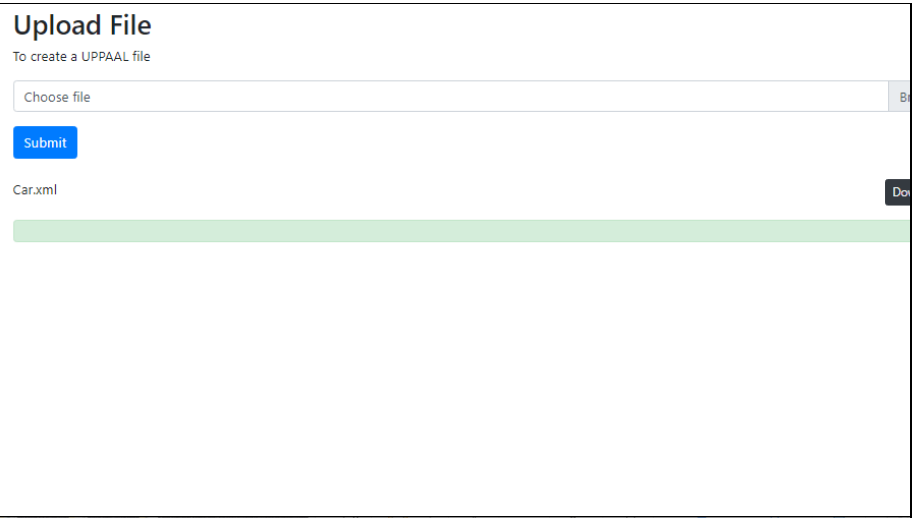
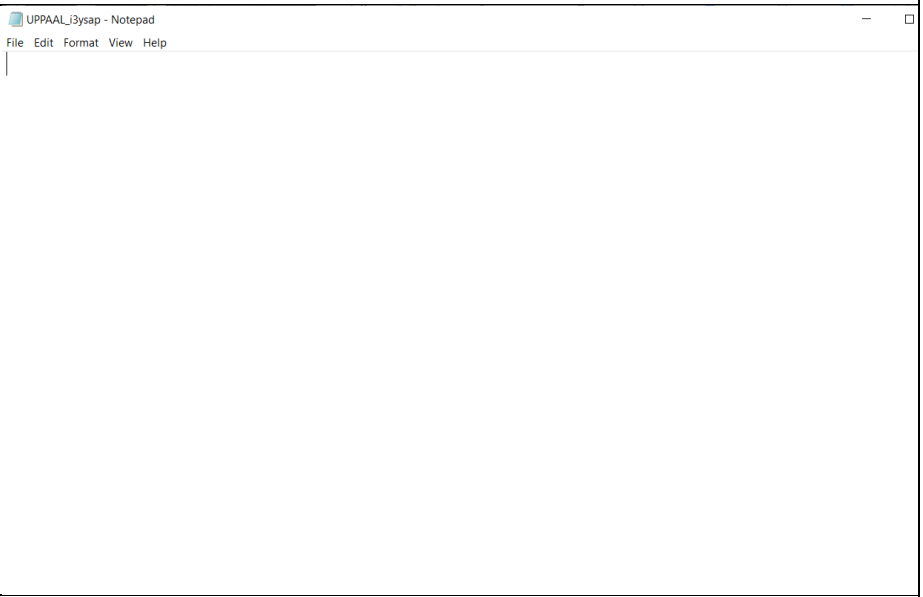
ตารางที่ 5-5 ทดสอบกรณีไม่ใส่ไฟล์ที่มีนามสกุลเอกซ์เอ็มแอลต้นทาง (ต่อ)

	
หมายเหตุ	ถ้าผู้ใช้เขียนนามสกุลไม่ใช่ไฟล์สกุลเอกซ์เอ็มแอลจะไม่สามารถเห็นไฟล์ได้ ต้องเลือกไฟล์ทั้งหมดถึงจะเห็นไฟล์นั้น

ตารางที่ 5-6 ทดสอบกรณีใส่ไฟล์ที่มีนามสกุลต้นทางเอกซ์เอ็มแอลแต่ไม่มีโครงสร้างที่ผู้วิจัยได้ออกแบบไว้ข้างต้น

Test Case ID	TestCase03
Precondition: (screen)	
Inputs	อัปโหลดไฟล์ที่ไม่ตรงตามโครงสร้างเอกซ์เอ็มแอลที่ผู้วิจัยระบุไว้

ตารางที่ 5-6 ทดสอบกรณีใส่ไฟล์ที่มีนามสกุลต้นทางเอกซ์เอ็มแอลแต่ไม่มีโครงสร้างที่ผู้วิจัยได้ออกแบบไว้ข้างต้น (ต่อ)

Postcondition: (screen)	
	
หมายเหตุ	จะดาวน์โหลดไฟล์เอกซ์เอ็มแอลได้แต่จะไม่มีข้อมูลปลายทางเลยทำให้ไม่สามารถนำเข้าเครื่องมือ UPPAAL ได้

บทที่ 6
บทสรุปและข้อเสนอแนะวิทยานิพนธ์

6.1 สรุปผลวิทยานิพนธ์

วิทยานิพนธ์นี้ได้มีการนิยามและออกแบบกฎการแปลงไทมด์ออโตมาตาสำหรับการจัดกำหนดการเชิงพรีเอ็มทีฟโดยเครื่องมือการแปลงที่พัฒนาขึ้น จากนั้นผู้วิจัยได้ทำเครื่องมือการแปลงอัตโนมัติที่ชื่อว่า “Timing Diagram into Timed Automata Converter” ซึ่งไฟล์นำเข้าต้นทางเป็นแผนภาพเวลาที่ระบุค่าลำดับความสำคัญและการทำงานเป็นอิสระต่อกัน นามสกุลไฟล์ต้นทางต้องเป็นไฟล์เอกซ์เอ็มแอลเท่านั้นไม่สามาถนำเข้าไฟล์นามสกุลอื่นได้ และเป็นไปตามที่ผู้วิจัยได้ออกแบบโครงสร้างเอกซ์เอ็มแอลไว้เมื่อนำไฟล์ต้นทางเข้าสู่เครื่องมือที่ผู้วิจัยออกแบบไว้จะได้ไฟล์ปลายทางออกมานั่นคือไฟล์ไทมด์ออโตมาตาที่มีการจัดกำหนดการเชิงพรีเอ็มทีฟ ซึ่งไฟล์ปลายทางจะมีรูปแบบเป็นเอกซ์เอ็มแอลในการทำเครื่องมือ “Timing Diagram into Timed Automata Converter” นั้นว่าแปลงได้ถูกต้องหรือไม่ โดยผู้วิจัยได้ทำการประยุกต์กรณีศึกษามา 3 กรณี อีกทั้งยังดักจับข้อผิดพลาดในหน้าเว็บอีกด้วยและนำไฟล์ปลายทางที่ได้เข้าเครื่องมือที่ชื่อว่า UPPAAL ที่สามารถใช้ในการทวนสอบคุณสมบัติต่อไปได้ ไม่ว่าจะเป็นคุณสมบัติการคงอยู่ คุณสมบัติความปลอดภัย และคุณสมบัติสภาวะการติดตาย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บรรณานุกรม

1. Laplante, P.A., *Real-time systems design and analysis*. 3 ed, ed. 3th. 2004: A JOHN WILEY & SONS, INC.,. 1-13.
2. วัฒนาวุฒิ, ว., *Formal Verification* การทวนสอบเชิงรูปนัย. 2560, พญาไท กรุงเทพฯ 10330: คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย. 239.
3. Cui, K., et al. *Unifying modeling and simulation based on uml timing diagram and uppaal*. in 2010 *Second International Conference on Computer Modeling and Simulation*. 2010. IEEE.
4. Sukvanich, P., A. Thongtak, and W. Vatanawood. *Formalizing Real-Time Embedded System into Promela*. in *MATEC Web of Conferences*. 2015. EDP Sciences.
5. Chandratarat, P. and W. Vatanawood. *Transforming WS-BPEL into Timed Automata*. in *Proceedings of the* 2019 *3rd International Conference on Software and e-Business*. 2019.
6. Alur, R. and D.L. Dill, *A theory of timed automata.* Theoretical computer science, 1994. 126(2): p. 183-235.
7. Group, O.M., *OMG Unified Modeling Language (OMG UML), superstructure, version* 2.5. 2011, Needham, MA, USA: Object Management Group. p. 601-604.
8. Silberschatz, A., P.B. Galvin, and G. Gagne, *Operating system concepts* 2009: Wiley Publishing.
9. Cottet, F., et al., *Scheduling in real-time systems*. 2002: Wiley Online Library.
10. Jaiswal, S., *Priority Scheduling.* 2011.
11. ทองจริง, ป., ระบบปฏิบัติการ *Operating Systems* Vol. 116-140. 2556, ต.ท่าช้าง อ.เมือง จ.จันทบุรี: สำนักวิทยบริการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏรำไพพรรณี.
12. Fakhroutdinov, K. *Timing Diagrams*. 2014; Available from: https://www.uml-diagrams.org/timing-diagrams.html.
13. จงสถิตย์วัฒนา, จ., ออโตมาตาจำกัด *Finite Automata*. 2559. 3-4.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียน

ชื่อ-สกุล	อมรัตน์ พิมโคตร
วัน เดือน ปี เกิด	วันอาทิตย์ที่ 27 สิงหาคม พ.ศ. 2538
สถานที่เกิด	สมุทรปราการ
วุฒิการศึกษา	วิทยาศาสตรบัณฑิต สาขาคณิตศาสตร์ประยุกต์ คณะวิทยาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ที่อยู่ปัจจุบัน	บ้านเลขที่ 223/305 หมู่บ้านทิพมาศเสรี-บางนา ซอย 10 หมู่ 2 ถนนปาณวิถี ตำบลบางเพรียง อำเภอบางบ่อ จังหวัดสมุทรปราการ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY