# รายการอ้างอิง


1. Brad Appleton. Patters and software: Essential concepts and terminology. Object Magazine online V.3, 5 (May 1997): 20-25.
2. S. Fraser, et al. Patterns: Cult to culture?. Proceedings of the 10th annual conference on Object-oriented programming systems languages and applications (OOPSLA) Vol. 30 Issue10, pp. 231-234. October, 1995.
3. Christopher Alexander. A Pattern Language: Towns, Buildings, Construction. Oxford University Press, 1977.
4. E. Gramma, R. Helm, R. Johnson and J. Wlissides. Design Patterns Elements of Reusable Object-Oriented Software. 7th Indian Reprint Edition. Pearson Education, 2002.
5. Patricia L. Ferdinandi. A Requirements Pattern: Succeesing in the Internet Economy. Addison-Wesley, June 2001.
6. S.Konrad and Betty H.C. Cheng. Requirements patterns for embedded systems. Proceedings of IEEE Joint International Conference on Requirements Engineering (RE'02), pp. 127-136. 2002.
7. Suzanne Robertson. Requirements patterns via events/use cases. The Atlantic Systems Guild. 1996.
8. Toshihiko Tsumaki. Requirements engineering pattern structure. Proceeding of IEEE The 11th Asia-Pacific software Engineering Conference (APSEC'2004), pp. 502-509. 2004.
9. L. Hagge, K. Lappe. Patterns for the RE process. Proceedings of IEEE The 12th International Requirements Engineering Conference (RE'04), pp. 90-99. 2004.
10. Nancy G. Leveson. Software: system safety and computers. Addison-Wesley, 1995.
11. David Alberico, et al. Software System Safety Handbook: A Technical & Managerial Team Approach. Joint Sevices Computer Resources Management Group, U.S. Navy, U.S Army and the U.S. Air Force, December 1999.
12. Robyn Lutz. Software engineering for safety: A roadmap. Proceedings of the conference on The Future of Software Engineering, International Conference on Software Engineering, pp. 213-226. 2000.
13. Gerard Meszaros and Jim Doble. A pattern language for pattern writing. Pattern languages of program design 3, pp. 529-574. 1997.

14. K. Beck and W. Cunningham. Using pattern languages for object-oriented programs. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'87), December, 1987.

15. Martin Fowler. Analysis Patterns: Reusable Object Models. Addison-Wesley, 1996.

16. James O. Coplien. Advanced C++ Programming Styles and Idioms. Addison-Wesley, 1991.

17. Frank Buschmann, et al. Pattern-Oriented Software Architecture, Volume 1: A System of Patterns. John Wiley & Sons, 1996.

18. Mark Grand. Patterns in Java, Volume 1: A Catalog of Reusable Design Patterns Illustrated with UML. John Wiley & Sons, 1998.

19. G. Kotonya and I. Sommerville. Requirements Engineering: processes and techniques. John Wiley & Sons, 1997.

20. D. Leffingwell and D. Widrig. Managing Software Requirements: A Use Case Approach. Second Edition. Pearson Education, 2003.

21. IEC 16508, Functional safety of electrical/electronic/programmable electronic safety-related systems. International Electrotechnical Commission (IEC), December 1997.

22. Edmund M. Clarke and Jeannette M. Wing. Formal methods: State of the art and future directions. ACM Computing Surveys, V.28, 4 (December 1996): 624-643.

23. F. Modugno, Nancy G. Leveson, et al. Integrated safety analysis of requirements specifications. Proceedings of the Third IEEE International Symposium on Requirements Engineering, pp. 148-159. January, 1997.

24. Donald Firesmith. A taxonomy of safety-related requirements. 3rd International Workshop on Requirements for High Assurance System (RHAS 2004), 2004.

25. I. Sommerville. Software engineering. 7 Edition. Addison Wesley, May 2004.

26. MIL-STD-882D, Standard practice for system safety. Department of Defense (DoD), February 10, 2000.

27. M. Mahemoff, A. Hussey and L. Johnston. Pattern-based reuse of successful designs: Usability of safety-critical systems. Proceedings of Software Engineering Conference (ASWEC'01), pp.31-39. August 2001.

28. I. Sommerville and P. Sawyer. Requirements Engineering: A good practice guide. John Wiley & Sons, 1997.

29. Magnus Penker, Penker and Hans-Erik Eriksson, Business Modeling With UML: Business Patterns at Work. John Wiley & Sons, 2000.

30. IEEE Dtd 1474.1-1999, IEEE Standard for Communications-Based Train Control(CBTC) Performance and Functional Requirements, September 16, 1999.

# ภาคผนวก

# ภาคผนวก ก
# บทความวิชาการ

บทความวิชาการของผู้วิจัยที่ได้รับการคัดเลือกและตีพิมพ์ในการประชุมวิชาการระดับชาติ คือ

บทความวิชาการเรื่อง "การนำความต้องการซอฟต์แวร์กลับมาใช้ใหม่ด้วยแบบรูปความต้องการ (Software Requirements Reuse by Requirements Patterns)" ซึ่งได้นำเสนอและตีพิมพ์ในงาน "การประชุมทางวิชาการทางวิทยาศาสตร์และวิศวกรรมคอมพิวเตอร์ระดับชาติ ครั้งที่ 9 (The 9th National Computer Science and Engineering Conference: NCSEC 2005)" ระหว่างวันที่ 27-28 ตุลาคม 2548 ณ มหาวิทยาลัยหอการค้าไทย กรุงเทพฯ

# การนำความต้องการซอฟต์แวร์กลับมาใช้ใหม่ด้วยแบบรูปความต้องการ
# Software Requirements Reuse by Requirements Patterns

เอกชัย แซ่ตั้ง และ นครทิพย์ พร้อมพูล
ห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ ศูนย์เชี่ยวชาญเฉพาะทางด้านวิศวกรรมซอฟต์แวร์
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
เขตปทุมวัน กรุงเทพฯ
อีเมล: Eakachai.C@gmail.com, Nakornthip.S@chula.ac.th

## Abstract

*Pattern is one form of software reuse which significantly beneficial to developer develops quality software. It is always used in system design and implement phase. If pattern is used in the early phase such as requirement engineering, it will effectively improve the quality of software development. This research presents two types of patterns used for requirements engineering: requirements domain patterns and requirements process patterns. These patterns a proposed based on the problem domain and software requirements engineering process.*

**Key Words:** Software Engineering, Software Requirements Engineering, Pattern, Requirements Pattern, Requirements Reuse

## บทคัดย่อ

*แบบรูป (Pattern) เป็นลักษณะหนึ่งของการนำกลับมาใช้ใหม่ ซึ่งนำไปใช้เพิ่มคุณภาพในการพัฒนาซอฟต์แวร์ แบบรูปมักถูกนำมาใช้ในขั้นตอน การออกแบบ และการพัฒนาโปรแกรม ถ้าสามารถนำแบบรูปมาใช้กับการพัฒนาในเฟสต้นๆอย่างวิศวกรรมความต้องการซอฟต์แวร์ได้ ก็จะสามารถเพิ่มประสิทธิภาพ ของการพัฒนาซอฟต์แวร์ได้อีกมาก งานวิจัยนี้จึงได้นำเสนอแบบรูป 2 ประเภทสำหรับวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งคือ แบบรูปโดเมนความต้องการ และแบบรูปกระบวนการความต้องการ ซึ่งนำเสนอมาจากปัญหาในโดเมนและกระบวนการในวิศวกรรมความต้องการซอฟต์แวร์*

**คำสำคัญ:** วิศวกรรมซอฟต์แวร์ วิศวกรรมความต้องการซอฟต์แวร์ แบบรูป แบบรูปความต้องการ การนำความต้องการกลับมาใช้ใหม่

## 1. บทนำ

การพัฒนาซอฟต์แวร์ในยุคปัจจุบัน วิศวกรรมความต้องการซอฟต์แวร์มีความสำคัญมากขึ้นเรื่อยๆ เนื่องจากพบว่าบ่อยครั้ง ที่การพัฒนาซอฟต์แวร์เกิดความล้มเหลวอันเนื่องมาจาก การไม่ให้ความสำคัญกับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ [1] ทำให้ได้ความต้องการที่ไม่สมบูรณ์ ไม่ได้รับความต้องการที่แท้จริงของผู้ใช้ ไม่สามารถรองรับเปลี่ยนแปลงความต้องการได้

แบบรูปเป็นที่รู้จักกันอย่างแพร่หลาย จากการนำไปใช้กับการออกแบบ [2] แต่โดยต้นกำเนิดแล้วแบบรูปมีขึ้นเพื่อนำไปใช้กับงานด้านสถาปัตยกรรม [3] แล้วจึงนำมาใช้ในการออกแบบส่วนต่อประสานผู้ใช้ [4] เป็นครั้งแรก ปัจจุบันมีผู้นำแบบรูปมาใช้ในขั้นตอนต่างๆของการพัฒนาซอฟต์แวร์เป็นจำนวนมาก [5, 6]

การนำแบบรูปมาใช้กับวิศวกรรมความต้องการ ซึ่งเรียกว่าแบบรูปความต้องการ (Requirements Pattern) มี

เป้าหมายในการแก้ปัญหาที่เกิดขึ้นในลักษณะเดียวกันบ่อยๆ มุ่งหวังผลลัพธ์ที่เป็นข้อกำหนดความต้องการของระบบ เพื่อนำไปใช้ในการพัฒนาซอฟต์แวร์

อนึ่งการใช้แบบรูปในปัจจุบัน มีความครอบคลุมในทุกส่วนของกระบวนการพัฒนาซอฟต์แวร์ ดังนั้นการทำแบบรูปให้มีความสมบูรณ์ และสามารถเชื่อมต่อไปถึงแบบรูปส่วนอื่นๆ ย่อมเพิ่มความสามารถในการนำกลับมาใช้ใหม่ และเป็นผลดีโดยรวม ในวงจรการพัฒนาซอฟต์แวร์ (Software Development Life Cycle: SDLC) ฟ

งานวิจัยนี้ทำการวิเคราะห์ลักษณะปัญหา ที่พบอยู่เสมอในงานวิศวกรรมความต้องการซอฟต์แวร์ โดยวิเคราะห์ตามมุมมองที่ขึ้นกับโดเมนของระบบ ประกอบกับแนวทางของกระบวนการทางวิศวกรรมความต้องการซอฟต์แวร์ เพื่อให้ได้ลักษณะแบบรูปความต้องการที่เหมาะสม สามารถนำมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์

โดยงานวิจัยนี้ได้แบ่งหัวข้อออกเป็น 6 หัวข้อด้วยกัน โดยหัวข้อที่ 2 ที่เป็นหัวข้อถัดไปจะกล่าวถึงงานวิจัยที่เกี่ยวข้อง ในหัวข้อที่ 3 จะกล่าวถึงหลักการและการวิเคราะห์การนำกลับมาใช้ใหม่สำหรับวิศวกรรมความต้องการซอฟต์แวร์ ก่อนจะสรุปออกมาเป็นโครงสร้างแบบรูปทั้ง 2 ประเภทในหัวข้อที่ 4 ในหัวข้อที่ 5 จะยกตัวอย่างกรณีศึกษาของการใช้แบบรูปกับระบบห้องสมุด และสรุปงานวิจัยในหัวข้อที่ 6

## 2. งานวิจัยที่เกี่ยวข้อง

แบบรูปเริ่มนำมาใช้กับ วิศวกรรมความต้องการซอฟต์แวร์ครั้งแรก โดย S. Robertson [7] ซึ่งเป็นแบบรูปที่ได้จากการวิเคราะห์ความต้องการจากเหตุการณ์ และแผนภาพยูสเคส (Use Case)

Sascha Konrad และ Betty H.C. Cheng ได้ใช้แบบรูปความต้องการกับระบบฝังตัว (Embedded Systems) [8] ทางด้าน Romi Satria Wohomo และ Jingde Cheng ได้นำแบบรูปความต้องการไปใช้กับ การพัฒนาโปรแกรมประยุกต์บนเว็บ [9] แบบรูปทั้งสองมีลักษณะการใช้ที่ขึ้นอยู่กับโดเมนของระบบเหมือนกัน แต่โครงสร้างของแบบรูปกลับมีลักษณะที่ต่างกัน ซึ่งทั้งสองแบบประยุกต์โครงสร้างแบบรูปที่ใช้จากแบบรูปการออกแบบ (Design Pattern) คนละตัวกัน ทั้งนี้โครงสร้างส่วนที่แต่ละงานวิจัยเพิ่มเข้ามาขึ้นอยู่กับลักษณะโดเมนตามการนำไปใช้

งานวิจัยสองชิ้นข้างต้นมีมุมมอง การใช้แบบรูปความต้องการที่ขึ้นอยู่กับโดเมนของระบบ แต่งานของ Marta Pantoquiho และคณะ [10] ให้มุมมองที่ต่างออกไป โดยพิจารณาแบบรูปการวิเคราะห์ (Analysis Pattern) เทียบกับแบบรูปความต้องการ ซึ่งให้ข้อสังเกตว่างานของ Konrad ที่กล่าวถึงในข้างต้นก็ไม่แตกต่างไปจากแบบรูปการวิเคราะห์ และเสนอลักษณะแบบรูปที่รวมลักษณะแบบรูปความต้องการ และแบบรูปการวิเคราะห์เข้าด้วยกัน

จะเห็นว่าแบบรูปความต้องการส่วนใหญ่จะเน้นไปที่ตัวความต้องการ แต่ Tsumaki [11] ได้นำเสนอแบบรูปวิศวกรรมความต้องการ (Requirements Engineering Pattern) ซึ่งเป็นการนำแบบรูปไปใช้กับกระบวนการทางวิศวกรรมความต้องการซอฟต์แวร์ โดยพิจารณาไปยังสถานการณ์ของปัญหา และแสดงผลลัพธ์ของแบบรูปเป็นวิธีการที่จะนำมาใช้แก้ปัญหาแทน

ในการนำแบบรูปไปใช้สำหรับวิศวกรรมความต้องการซอฟต์แวร์ จะเห็นได้ว่าด้านหนึ่งพิจารณาตามลักษณะของโดเมนของระบบ อีกด้านหนึ่งพิจารณาตามลักษณะกระบวนการ ซึ่งทั้ง 2 ลักษณะมีข้อดีแตกต่างกัน งานวิจัยนี้จึงได้นำเสนอแบบรูปความต้องการ ที่มีข้อดีของแบบรูปทั้ง 2 ลักษณะ กำหนดเป็นโครงสร้างที่เหมาะสมเพื่อให้เกิดประสิทธิภาพในการนำไปใช้ และครอบคลุมงานวิศวกรรมความต้องการซอฟต์แวร์

## 3. การวิเคราะห์วิศวกรรมความต้องการซอฟต์แวร์เพื่อการนำกลับมาใช้ใหม่

แบบรูป คือ ปัญหา (Problem) และผลเฉลย (Solution) ซึ่งเกิดขึ้นซ้ำๆกัน [12] การวิเคราะห์หาโครงสร้างแบบรูปจึงต้องทำการพิจารณาไปยังปัญหา และวิธีการเพื่อจะได้ผลเฉลย และอาศัยมุมมองของการนำกลับมาใช้ใหม่ตามที่วิเคราะห์ ดังนี้

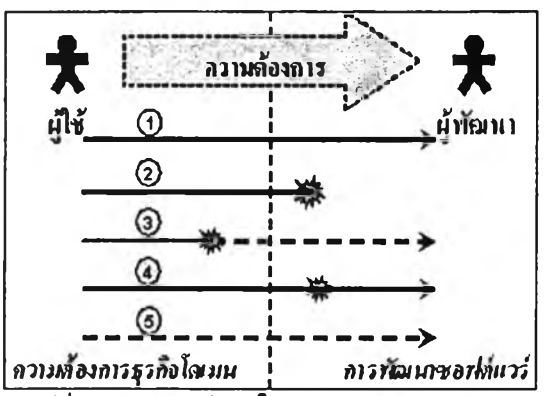
### 3.1 ปัญหาในวิศวกรรมความต้องการซอฟต์แวร์

ลักษณะปัญหาที่พิจารณา ในวิศวกรรมความต้องการซอฟต์แวร์ แบ่งออกเป็น 2 กลุ่ม คือ กลุ่มปัญหาที่สัมพันธ์กับโดเมนของระบบ และ กลุ่มปัญหาที่เกี่ยวข้องกับเทคนิคและกระบวนการ

1) <u>ปัญหาด้านโดเมน</u>

มีสาเหตุที่สำคัญมาจากองค์ประกอบของความต้องการมีความคลาดเคลื่อนซึ่งเกิดขึ้น 3 ลักษณะ ดังนี้

- ผู้ใช้ (User) เช่น ผู้ใช้ให้ความต้องการไม่ชัดเจน
- ผู้พัฒนา (Developer) เช่น ผู้พัฒนาทราบความต้องการไม่ครบ
- การสื่อสาร (Communicate) เช่น ผู้ใช้และผู้พัฒนามีความเข้าใจไม่ตรงกัน




**รูปที่ 1 แบบจำลองปัญหาในวิศวกรรมความต้องการซอฟต์แวร์**

รูปที่ 1 แสดงแบบจำลองลักษณะของปัญหากับวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งเห็นได้ว่าปัญหาสามารถเกิดขึ้นได้ทั้งในมุมมองความต้องการธุรกิจโดเมน (Business Domain) และมุมองการพัฒนาซอฟต์แวร์ โดยสามารถจำแนกลักษณะความต้องการได้เป็น 5 ประเภท ตามรูป ซึ่งมีความหมาย ดังนี้

*<u>ประเภทที่ 1</u>*

ความต้องการที่ส่งไปยังผู้พัฒนาโดยไม่เกิดปัญหา

*<u>ประเภทที่ 2</u>*

ความต้องการที่ไม่สามารถเกิดขึ้นได้ เนื่องจากไม่สามารถหลีกเลี่ยงปัญหาที่พบ เช่น ความต้องการซับซ้อนมาก ความต้องการไม่สามารถพัฒนาให้ใช้ได้จริง เป็นต้น

*<u>ประเภทที่ 3</u>*

ความต้องการที่เบี่ยงเบนหรือผิดพลาดเนื่องจากปัญหาระหว่างกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

*<u>ประเภทที่ 4</u>*

ความต้องการที่นำไปใช้พัฒนาได้ตามปกติ แม้มีปัญหาเกิดขึ้น

*<u>ประเภทที่ 5</u>*

ความต้องการที่ผิดพลาดมาแต่ต้น เป็นผลให้การนำไปใช้เกิดความผิดพลาดตามมา โดยไม่รู้ตัว

2) <u>ปัญหาด้านกระบวนการ</u>

ขึ้นอยู่กับกลุ่มกระบวนการหลักทางวิศวกรรมความต้องการซอฟต์แวร์ที่นำมาใช้ เกิดปัญหาและความผิดพลาดขึ้น ซึ่งกลุ่มกระบวนการทั้งหมด[13, 14] ประกอบด้วย

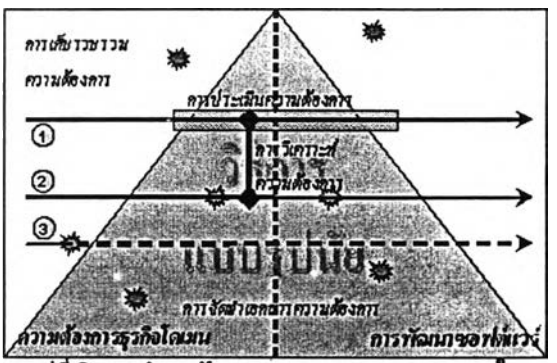
- การเก็บรวบรวมความต้องการ
- การวิเคราะห์ความต้องการ
- การจัดทำเอกสารความต้องการ
- การประเมินความต้องการ
- การบริหารความต้องการ

ความสัมพันธ์ของกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ สามารถพิจารณาได้ตามรูปที่ 2 โดยเป็นกระบวนการที่สัมพันธ์กับความต้องการ จากมุมมองด้านธุรกิจโดเมน และมุมมองมองด้านการพัฒนาซอฟต์แวร์ ซึ่งกระบวนการทางวิศวกรรมความต้องการซอฟต์แวร์ อาจอาศัยวิธีการที่เป็นรูปนัย (Formal) กึ่งรูปนัย (Semi-formal) หรือไม่เป็นรูปนัย (Informal) แต่โดยมากแล้วกระบวนการที่เกิดขึ้นมักจะอาศัยวิธีการที่เป็นรูปนัยหรือกึ่งรูปนัยทั้ง กระบวนการประเมิน กระบวนการตรวจสอบ และกระบวนการจัดทำเอกสาร

เมื่อเปรียบเทียบกับโมเดลของปัญหาสามารถสรุปได้ว่าการเก็บความต้องการจะเป็นการเริ่มต้นส่งต่อความต้องการ

ในธุรกิจโดเมนไปสู่ผู้พัฒนาซอฟต์แวร์ การจัดทำเอกสารจะเป็นการจัดระเบียบความต้องการ ให้เกิดความชัดเจนมากขึ้น และสามารถนำไปประเมินและวิเคราะห์ความต้องการต่อไป โดยการประเมินจะทำกับความต้องการโดยอิสระเพื่อพิจารณากลับไปว่า ความต้องการเป็นไปตามความต้องการเริ่มต้นของผู้ใช้หรือไม่ ส่วนการพิจารณาระหว่างความต้องการว่าครงหรือขัดแย้งหรือก่อเกิดปัญหาตามมา จะเป็นหน้าที่ของกระบวนการวิเคราะห์ความต้องการ

สำหรับกระบวนการบริหารความต้องการจะเป็นกระบวนการที่พิจารณาในระดับบนควบคู่กับการบริหารโครงการ ซึ่งจะมาจัดการกับความเปลี่ยนแปลงความต้องการ การกำหนดขั้นตอนและกระบวนการ




**รูปที่ 2 แบบจำลองปัญหา มุมมอง และกระบวนการ ในวิศวกรรมความต้องการซอฟต์แวร์**

การใช้เทคนิคและวิธีการที่เหมาะสมที่ใช้กันเสมอ เรียกได้ว่าเป็นวิธีการปฏิบัติ (Practice) [1] หากพิจารณาตามลักษณะกระบวนการ วิธีการปฏิบัติเป็นขั้นตอนที่ช่วยลดปัญหาที่เกิดขึ้นโดยไม่ขึ้นกับลักษณะโดเมน หากพิจารณาในรูปที่ 2 วิธีการปฏิบัติก็คือความต้องการที่ 1 และ 2 ที่ส่งไปพัฒนาซอฟต์แวร์ได้อย่างสมบูรณ์ ซึ่งวิธีการปฏิบัติก็จะมีส่วนช่วยลด และตรวจพบปัญหาได้ ดังเช่น ความต้องการที่ 2 ในรูป อย่างไรก็ดีวิธีการปฏิบัตินั้นเจาะจงไปที่กระบวนการ ดังนั้นวิธีการปฏิบัติจึงเป็นแนวทางที่เป็นกลางไม่ขึ้นกับโดเมนใดเมนหนึ่ง ซึ่งถือเป็นข้อด้วยหากต้องการที่จะมีการเจาะจงการทำงานที่เป็นไปในแนวทางของโดเมนใดเป็นพิเศษ

### 3.2 การนำกลับมาใช้ใหม่สำหรับวิศวกรรมความต้องการซอฟต์แวร์

จากลักษณะปัญหาที่พิจารณาจะเห็นได้ว่าปัญหาทั้ง 2 กลุ่ม มีที่มาและพฤติกรรมที่ต่างกัน ปัญหาจากโดเมนของระบบ เป็นปัญหาที่พบอีกครั้งเมื่อมีการพัฒนาระบบใหม่ที่มีความคล้ายคลึงกับระบบเดิม ซึ่งเป็นปัญหามักจะขึ้นอยู่กับลักษณะของโดเมน แต่ปัญหาของกระบวนการเป็นปัญหาที่เกิดจากการทำงาน ซึ่งในหลายๆครั้งเป็นปัญหาในการเลือกเทคนิคหรือวิธีการ ให้เหมาะสมกับโดเมนที่กำลังพัฒนาอยู่ และมักจะไม่ยึดติดกับประเภทของโดเมน

ดังที่แสดงในรูปที่ 2 การที่ทราบปัญหาที่เกิดขึ้นไม่ว่าจากมุมทองผู้ใช้ในธุรกิจโดเมน และมุมมองผู้พัฒนาในส่วนการพัฒนาซอฟต์แวร์ เราจะสามารถหลีกเลี่ยงปัญหาหรือหาวิธีทางในการจัดการกับปัญหาได้ดีกว่า ซึ่งถือได้ว่าเป็นการนำองค์ความรู้กลับมาใช้ใหม่ และการทราบกระบวนการที่เหมาะสมตลอดจนวิธีการปฏิบัติที่ทราบแล้วว่าเป็นเทคนิคและวิธีการที่ประสบความสำเร็จก็จะช่วยให้ทราบและเข้าใจปัญหาในการใช้เทคนิควิธีการนั้น ช่วยให้สามารถตัดสินใจใช้เทคนิคและวิธีการได้อย่างถูกต้อง ซึ่งเป็นการนำเทคนิควิธีการกลับมาใช้ใหม่

ดังนั้นในการนำความต้องการกลับมาใช้ใหม่ จะทำการหาองค์ประกอบที่จะนำมาใช้แก้ปัญหา หรือองค์ประกอบที่อธิบายหรือช่วยในการทำงาน โดยองค์ประกอบความรู้ที่จะนำมาใช้ในการนำกลับมาใช้ใหม่ เกิดจากการแยกพิจารณามุมมองของของปัญหา โดยขึ้นกับโดเมนและกระบวนการตามที่กล่าวถึงข้างต้น ซึ่งทำให้องค์ประกอบความรู้ที่จะนำมาใช้ในการนำกลับมาใช้ใหม่ แบ่งออกเป็น 2 กลุ่มด้วยเช่นกัน คือ กลุ่มที่แสดงด้านโดเมนซึ่งทำให้เห็นปัญหาและเข้าใจในลักษณะของโดเมน และกลุ่มที่แสดงด้านกระบวนการ ซึ่งช่วยเพิ่มความเข้าใจและลดปัญหาในการใช้เทคนิคหรือวิธีการ โดยสามารถสรุปองค์ประกอบความรู้ได้ดังนี้

- *องค์ประกอบความรู้ด้านโดเมน ประกอบด้วย*
  - *1) เป้าหมาย*
  - *2) ความหมายและนิยาม*

*3) ลักษณะของโดเมน*

*4) ความต้องการ*

*5) ปัญหาของโดเมน*

*6) ข้อกำหนดและกฎเกณฑ์*

*7) ปัญหาของการพัฒนา*

*8) สภาพแวดล้อมของระบบ*

- *องค์ประกอบความรู้ด้านกระบวนการ ประกอบด้วย*

*1) สถานการณ์*

*2) จุดประสงค์*

*3) กลุ่มประเภทของระบบ*

*4) ประสบการณ์*

*5) ขั้นตอนวิธี*

*6) เครื่องมือ*

*7) ค่าใช้จ่าย*

องค์ประกอบความรู้ด้านโดเมน จะช่วยให้เข้าใจและทราบปัญหาในมุมมองต่างๆ ทำให้สามารถแก้ไขหรือหลีกเลี่ยงปัญหาตามลักษณะโดเมน ส่วนองค์ประกอบความรู้ด้านกระบวนการจะเป็นเครื่องมือในการตัดสินใจใช้เทคนิคหรือวิการที่มีประสิทธิผลตามวิธีปฏิบัติ ซึ่งทั้งหมดเป็นองค์ความรู้ที่เพียงพอในการนำความต้องการกลับมาใช้ใหม่ในลักษณะการนำกลับมาใช้ใหม่ทางองค์ความรู้ และการนำกลับมาใช้ใหม่ทางวิธีการ

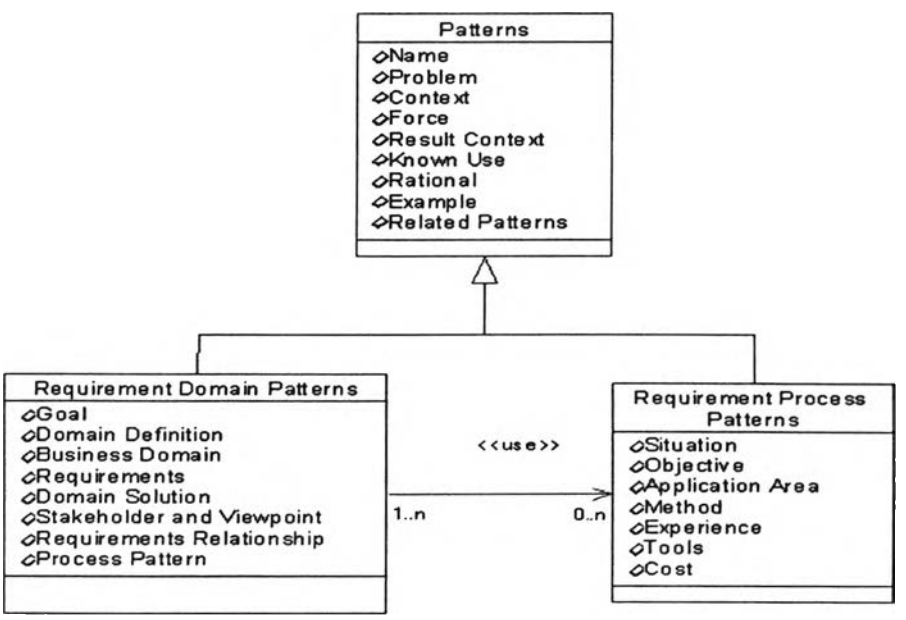
## 4. การออกแบบโครงสร้างแบบรูปความต้องการซอฟต์แวร์

โครงสร้างของแบบรูปความต้องการ เกิดจากการนำเอาองค์ประกอบความรู้ ที่ได้จากการวิเคราะห์ปัญหาและการนำกลับมาใช้ใหม่ ทางวิศวกรรมความต้องการซอฟต์แวร์มาสรุปเป็นลักษณะของแบบรูปที่เหมาะสม

การกำหนดโครงสร้างแบบรูปให้เหมาะสม ไม่มีลักษณะตายตัว แต่เมื่อพิจารณาตามลักษณะขององค์ความรู้ในการนำความต้องซอฟต์แวร์กลับมาใช้ใหม่ ร่วมกับการใช้แบบรูปในการสร้างแบบรูป [15] ทำให้สามารถวิเคราะห์และสรุปแบบรูปได้ ดังนี้

### 4.1 ลักษณะและประเภทของแบบรูป

แบบรูปสำหรับวิศวกรรมความต้องการซอฟต์แวร์ เมื่อพิจารณาจากปัญหาซ้ำเพื่อหาผลเฉลยที่เหมาะสม จะพบว่ามีปัญหาใน 2 กลุ่ม ตามที่เสนอในเบื้องต้น คือ ปัญหาของโดเมน และปัญหาของกระบวนการ และตามงานที่มีผู้




**รูปที่ 3 องค์ประกอบและความสัมพันธ์ของแบบรูปความต้องการซอฟต์แวร์**

นำเสนอไปก่อนหน้า มีทั้งการนำเสนอแบบรูปที่แก้ปัญหาเพียงแค่ด้านโดเมนหรือกระบวนการเพียงด้านเดียว

ดังนั้นแบบรูปที่วิเคราะห์ตามงานวิจัยนี้จึงควรประกอบไปด้วยทั้ง 2 ส่วน แต่ก็จะมีลักษณะองค์ประกอบในเบื้องต้นของแบบรูปที่เหมือนกันอยู่ ได้แก่ ชื่อ (Name) ปัญหา (Problem) สภาพแวดล้อม (Context) ฟอร์ซ (Force) แบบรูปที่เกี่ยวข้อง (Related Patterns) และตัวอย่าง (Example) โดยแบบรูปทั้ง 2 ประเภท กำหนดชื่อเรียก ความหมาย และรายละเอียดไว้ ดังนี้

1) แบบรูปโดเมนความต้องการ

คือ แบบรูปที่นำมาใช้แก้ปัญหา ที่เกิดขึ้นในการเก็บความต้องการสำหรับโดเมนที่กล่าวถึง

2) แบบรูปกระบวนการความต้องการ

คือ แบบรูปที่นำมาใช้แก้ปัญหา ที่เกิดขึ้นกับเทคนิคหรือกระบวนการ ที่นำมาใช้กับความต้องการที่กล่าวถึง

ตามรูปที่ 3 แสดงโครงสร้างและองค์ประกอบของแบบรูปความต้องการ โดยอาศัยโครงสร้างแบบรู้ที่เป็นพื้นฐานและใช้กันทั่วไป [2] มาทำการเพิ่มเติมองค์ประกอบตามลักษณะของ แบบรูปทั้ง 2 ประเภท

สำหรับองค์ประกอบของแบบรูปทั้ง 2 ที่ใช้ร่วมกัน อธิบายได้ ดังนี้

*1) Pattern Name*

ชื่อแบบรูปเพื่อใช้เรียก และบอกประเภทของแบบรูป

*2) Problem*

ปัญหาที่เกิดขึ้น โดยจะนำไปใช้ตัดสินใจเลือกใช้แบบรูป ปัญหาที่เกิดขึ้นจะแสดงลักษณะปัญหาที่สังเกตได้และเกิดขึ้นซ้ำๆกัน

*3) Context*

สภาวะแวดล้อมที่เคยเกิดปัญหาขึ้น ซึ่งใช้ประกอบกับปัญหาที่เกิดขึ้น เป็นอีกส่วนที่ช่วยในการพิจารณาใช้แบบรูป แม้ว่าปัญหาที่เกิดขึ้นเหมือนกัน แต่ลักษณะสภาวะแวดล้อมต่างกัน อาจใช้ผลเฉลยแบบเดียวกันไม่ได้

*4) Force*

ปัจจัยบ่งชี้ลักษณะที่เจาะจงลงไป เพื่อช่วยตัดสินใจเลือกแบบรูปมาใช้แก้ปัญหา

*5) Result Context*

แสดงผลลัพธ์ที่ได้ หรือผลลัพธ์ที่ตามมาหลังจากใช้แบบรูปนี้ แสดงสิ่งที่น่าจะเกิดขึ้นจากการใช้แบบรูป ทั้งทางตรงและทางอ้อม

*6) Known Use*

แสดงให้เห็นว่านำไปใช้แก้ปัญหาได้จริง และลักษณะการนำเอาไปใช้มีลักษณะอย่างไร การใช้จะนำไปใช้ในกรณีไหนบ้าง

*7) Rational*

ให้เหตุผลในการสนับสนุนว่า ผลลัพธ์ของแบบรูปนั้นใช้ได้จริง แสดงความสำเร็จหรือผลที่จะได้ซึ่งเกิดขึ้นจริง

*8) Example*

แสดงตัวอย่างของการนำไปใช้

*9) Related Patterns*

แสดงความเกี่ยวเนื่องกับแบบรูปอื่นๆ

### 4.2 แบบรูปโดเมนความต้องการ

**รูปที่ 4 โครงสร้างแบบรูปโดเมนความต้องการ**

แบบรูปโดเมนความต้องการ มีองค์ประกอบที่สัมพันธ์กันซึ่งพิจารณาได้จากรูปที่ 4 โดยมีองค์ประกอบที่เพิ่มเติม เข้ามาจากลักษณะแบบรูปโครงสร้างปกติ ดังนี้

*1) Goal*

แสดงเป้าหมาย ซึ่งเป็นตัวบ่งบอกความต้องการเพื่อให้เข้าใจมูลเหตุของปัญหาในความต้องการนั้น

*2) Domain Definition*

เป็นตัวช่วยในการให้ความหมายของระบบ ซึ่งมีความจำเป็นต้องเข้าใจในทางเดียวกัน

*3) Business Domain*

แสดงลักษณะและการทำงานตามธุรกิจโดมนที่เป็นอยู่ เพื่อจะได้เห็นภาพและเข้าใจเหตุผลทางธุรกิจ

*4) Requirements*

ความต้องการซึ่งเป็นส่วนสำคัญและกำหนดระบบในการพัฒนา

*5) Domain Solution*

คือแนวทางและผลลัพธ์ สำหรับวิศวกรรมความต้องการตามโดเมนที่เป็นปัญหาอยู่ คำถามหรือประเด็นที่ต้องหาคำตอบที่นำไปใช้กับความต้องการ

*6) Stakeholders and Viewpoint*

วิธีการและมุมมองที่ช่วยให้เข้าใจผลเฉลยมากขึ้น ซึ่งเป็นลักษณะผลเฉลยสำหรับมุมมองหรือบุคคล ที่มีน่าที่ต่างกัน

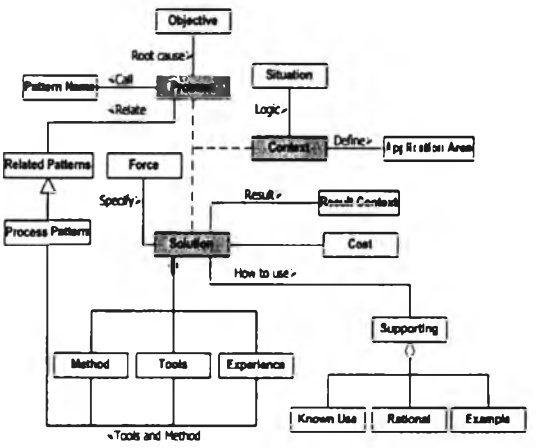
*7) Requirements Relationship and Constant*

อธิบายแสดงความเกี่ยวข้อง การขัดแย้งหรือซ้ำซ้อนในตัวความต้องการ การให้ระดับความสำคัญ ความต้องการที่ขึ้นแก่กัน

*8) Process Patterns*

ส่วนท้ายอิงไปยังแบบรูปกระบวนการความต้องการ เพื่อที่จะนำวิธีการหรือเครื่องมือที่เหมาะสมมาใช้ได้อย่างมีประสิทธิภาพ

### 4.3 แบบรูปกระบวนการความต้องการ




**รูปที่ 5 โครงสร้างแบบรูปกระบวนการความต้องการ**

แบบรูปกระบวนการความต้องการ มีโครงสร้างและความสัมพันธ์ระหว่างองค์ประกอบดังรูปที่ 5 โดยมีองค์ประกอบที่เป็นลักษณะของแบบรูปกระบวนการความต้องการ ดังนี้

*1) Situation*

เป็นสถานการณ์ ที่ควรนำแบบรูปกระบวนการความต้องการไปใช้

*2) Objective*

คือจุดประสงค์หลักของกระบวนการ ว่าจะสามารถนำไปใช้แก้ปัญหาในลักษณะใดอย่างไร

*3) Application Area*

อธิบายว่าแบบรูปมีความเหมาะสม ที่จะนำไปใช้กับโดเมนในกลุ่มหรือลักษณะใด

*4) Method*

วิธีการของกระบวนการ วิธีปฏิบัติที่ทำให้ประสบผลตามจุดประสงค์ที่มี

*5) Experience*

ลักษณะประสบการณ์ ที่สามารถนำมาใช้ช่วยให้เกิดประสิทธิภาพในกระบวนการที่นำมาปฏิบัติ

*6) Tools*

เครื่องมือที่มีความเมาะสม และสามารถนำมาช่วยเพิ่มประสิทธิภาพในการใช้กระบวนการได้

*7) Cost*

คือค่าใช้จ่ายที่ต้องเสียไป ไม่ว่าจะเป็นเวลาหรือทรัพยากร ซึ่งสามารถนำมาร่วมกับการตัดสินใจในการเลือกกระบวนการได้ หากเห็นว่ามีค่าใช้จ่ายที่ไม่เหมาะสม

### 4.4 ความสัมพันธ์และการนำแบบรูปไปใช้

แบบรูปสำหรับวิศวกรรมความต้องการซอฟต์แวร์ทั้ง 2 ประเภท มีลักษณะแตกต่างกัน แม้ว่าทั้ง 2 จะเป็นอิสระ แต่ยังมีความสมบูรณ์ในการนำไปใช้ในตัวเอง และเนื่องจากแบบรูปทั้ง 2 มีความสัมพันธ์กันกันในแง่ของการใช้ ซึ่งแบบรูปกระบวนการเป็นเครื่องมือและวิธีการ ที่สนับสนุนแบบรูปโดเมนให้เกิดประสิทธิภาพ และขยายลักษณะองค์ประกอบของผลเฉลยออกไปได้ เมื่อนำแบบรูปทั้ง 2 ไปใช้งานร่วมกันจึงทำให้เกิดประสิทธิภาพมากกว่าการใช้ลักษณะแบบรูปเพียงอย่างใดอย่างหนึ่ง

## 5. แบบรูปความต้องการ

ในส่วนนี้จะนำเสนอตัวอย่างแบบรูปความต้องการ ทั้ง 2 ประเภท ซึ่งเป็นความต้องการซอฟต์แวร์ของระบบห้องสมุด

### 5.1 แบบรูปโดเมนความต้องการ: *ยืม-คืน หนังสือ*

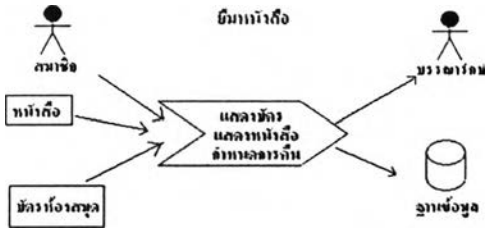
**Pattern Name:** ยืม-คืน หนังสือ

**Problem:** ระบบบันทึกและบริหารการยืมคืนหนังสือ ซึ่งการยืมคืนหนังสือจะต้องมีการเก็บข้อมูลผู้ยืม และแก้ไขสถานะของหนังสือในระบบ

**Context:** ระบบหอสมุดแบบเดี่ยว (ไม่มีการยืมหรือคืนหนังสือข้ามห้องสมุด)

**Goal:** สามารถให้บริการยืมหรือคืนหนังสือได้ พร้อมทั้งบันทึกข้อมูลละเปลี่ยนสถานการณ์ยืม

**Business Domain:**



**Domain Definition:**

*บรรณารักษ์* – เจ้าหน้าที่ดูแลห้องสมุดซึ่งเป็นผู้ที่ให้บริการยืม-คืนหนังสือด้วย

สมาชิกห้องสมุด – ผู้มีสิทธิ์ยืมหนังสือ

*รหัสหนังสือ* – รหัสที่ใช้อ้างอิงหมวดหมู่หรือการจัดประเภทของหนังสือ

*หมายเลขหนังสือ* – หมายที่ใช้อ้างอิงในฐานข้อมูลหนึ่งหมายเลขจะใช้กับหนังสือหนึ่งเล่มเท่านั้น ซึ่งอาจใช้ร่วมกันกับรหัสหนังสือ

*บัตรห้องสมุด* – บัตรที่ใช้แสดงความเป็นสมาชิกห้องสมุดและใช้ในการยืม-คืนหนังสือ

**Requirements:**

1) สมาชิกห้องสมุดทำการยืมหนังสือได้
2) บรรณารักษ์ทำการตรวจสอบสถานะของผู้ยืมหนังสือ
3) ระบบทำการแก้ไขข้อมูลในฐานข้อมูลเมื่อมีการยืมหนังสือ
4) สมาชิกห้องสมุดทำการคืนหนังสือได้
5) ระบบตรวจสอบและคำนวณค่าปรับการยืมหนังสือเกินเวลาเมื่อมีการคืนหนังสือ
6) ระบบทำการแก้ไขฐานข้อมูลหนังสือเมื่อทำการคืนหนังสือ
7) สมาชิกห้องสมุดทำการต่อเวลาการยืมหนังสือได้

**Requirements Relationship and Constant:**

1) จำนวนหนังสือที่จะยืมได้จะต้องไม่เกินจำนวนและประเภทที่ถูกกำหนด
2) หนังสือที่ให้ยืมต้องไม่ถูกจองหรือเป็นหนังสือที่ห้ามยืม
3) การยืมต่อจะทำได้ถ้าหนังสือไม่ถูกจอง
4) การยืมต่อจะทำได้ถ้าหนังสือยังไม่เกินเวลายืมปกติ
5) เมื่อหนังสือถูกยืม การสืบค้นต้องแสดงว่ามีแค่ถูกยืมอยู่

**Stakeholder and Viewpoint:**

บรรณารักษ์ – ต้องการเห็นสถานะของหนังสือ และผู้เป็นสมาชิกทั้งหมดบนหน้าจอแสดงผล การที่สมาชิกเข้ามายืมหนังสือ จะต้องยื่นหนังสือและบัตรสมาชิกห้องสมุด เพื่อใส่ค่าลงระบบ การคืนหนังสือก็มีลักษณะคล้ายๆกัน บรรณารักษ์ไม่จำเป็นต้องดูแลอะไร และมีข้อความเตือนหรือแจ้งทุกครั้งเกี่ยวกับ การยืมลักษณะที่ยืมไม่ได้ การคืนที่จะต้องเสียค่าปรับ หรือสมาชิกที่มายืมมีปัญหายังไม่คืนหนังสือก็ควรแจ้งด้วย

สมาชิกห้องสมุด – อยากให้งานยืมและคืนทำได้เร็วที่สุด รายละเอียดและกฎในการยืมเป็นสิ่งที่รู้อยู่แล้ว ไม่จำเป็นต้องมีระบบส่วนนี้สนับสนุนก็ได้ แต่อยากได้รายละเอียดที่แสดงว่ายืมเมื่อใดและจะต้องคืนเมื่อไร และไม่ต้องการกรอกแบบฟอร์ม หรือรายการใดๆเพื่อทำการยืม-คืนหนังสือ

**Domain Solution:**

มีอุปกรณ์รับค่าเข้าสู่ระบบเช่นตัวอ่านบาร์โค๊ด แถบแม่เหล็ก และทำการเก็บข้อมูลทั้งหนังสือและสมาชิกลงฐานข้อมูลกลางที่ใช้ร่วมกันกับระบบส่วนอื่น มีการออกใบเสร็จค่าปรับและกำหนดเวลาคืนหนังสือซึ่งเป็นไปได้ที่

จะเป็นระบบอัตโนมัติ จุดยืมคืนหนังสือเป็นจุดที่จะได้ติดต่อกับผู้ถือบัตร ดังนั้นจุดนี้อาจใช้แจ้งข้อมูลกับผู้ถือบัตรรายบุคคล เช่น มีหนังสือที่ยังไม่คืน

การทำรายการความต้องการจะช่วยตรวจสอบว่าความต้องการมีความครบถ้วนแค่ไหน การวิเคราะห์กลุ่มความต้องการจะทำให้เห็นส่วนที่จะเกี่ยวข้องกับความต้องการส่วนอื่น การทำต้นแบบก็จะช่วยทดสอบลักษณะหน้างานของเจ้าหน้าที่บรรณารักษ์เพื่อเก็บความต้องการด้านการใช้งานได้

**Result Context:**

การยืม-คืนหนังสือแบบอัตโนมัติหรือกึ่งอัตโนมัติ ที่มีการนำเครื่องมืออ่านค่าจากบัตรและหนังสือมาใช้ ซึ่งทำให้ระยะเวลาในการยืม-คืนหนังสือสั้นลง และทราบข้อมูลหนังสือและสมาชิกห้องสมุดในขณะยืม-คืนทันที

**Known Use:**

นำไปใช้กับห้องสมุดที่มีขนาดใหญ่ จะได้ผลดีมาก สำหรับห้องสมุดขนาดเล็กอาจไม่มีความจำเป็น หรือได้ประโยชน์ไม่มากนัก

**Related Patterns:**

การสืบค้นหนังสือ

การจองหนังสือ

การทำหนังสือหาย

**Process Patterns:**

รายการตรวจสอบความต้องการ

ต้นแบบ (Prototype)

การวิเคราะห์กลุ่มความต้องการ

### 5.2 แบบรูปกระบวนการความต้องการ: รายการตรวจสอบความต้องการ (Requirements Checklist)

**Pattern Name:** รายการตรวจสอบความต้องการ

**Problem:** จะทำอย่างไรเพื่อที่จะมั่นใจได้ว่าสามารถแสดงความต้องการที่มีทั้งหมดได้อย่างครบถ้วน

**Objective:** ต้องการรายการที่สามารถนำมาใช้ตรวจสอบความต้องการ ที่มีทั้งหมดได้โดยครบถ้วย

**Context:** มีความต้องการที่ครอบคลุมในหลายๆส่วน ต้องการนำความต้องการ มาทำเป็นรายการ เพื่อตรวจสอบ

**Situation:**

1) ต้องการแสดงความต้องการทั้งหมดที่มี
2) ต้องการนำความต้องการไปตรวจสอบ
3) จะจัดลำดับและให้ความสำคัญกับความต้องการ
4) การมีความต้องการในลักษณะกว้าว มีลักษณะแยกย่อยหลายด้าน

**Application Area:**

1) ในระบบที่ต้องการตรวจสอบให้แน่ใจว่าความต้องการครอบคลุมหรือครบหรือไม่
2) ในระบบที่ต้องการกลับมาตรวจสอบว่า สิ่งที่พัฒนาเป็นไปตามความต้องการที่มีทั้งหมด
3) ระบบที่เน้นกับความต้องการที่เป็นหน้าที่การทำงาน

**Method:**

1) ทำการจดบันทึกความต้องการทั้งหมดที่มีอยู่
2) จัดกลุ่มหรือหมวดหมู่ของความต้องการ
3) บันทึกรายละเอียดในการลำดับรายการ
4) เพิ่มเติม แก้ไข หรือตรวจสอบรายการทุกครั้ง ที่มีการเปลี่ยนแปลงความต้องการ

**Experience:**

การจัดหมวดหมู่ของความต้องการ จะช่วยให้ลำดับรายการได้ไม่สับสน การลำดับอาจเขียนหรือจดขึ้นมาเป็นส่วนๆ และอาจได้มาจากขั้นตอนการเก็บความต้องการที่มีการบันทึกไว้ การเขียนความต้องการควรเขียนให้ชัดเจน คล้ายกับการเขียนในข้อกำหนดความต้องการ

ควรนำรายการตรวจสอบความต้องการ ไปจัดลำดับความสำคัญ อาจมีการกำหนดรายการการตรวจสอบในลักษณะแบบฟอร์มที่มีช่องตรวจสอบ ช่องใส่คำอธิบายเพิ่มเติม และช่องใส่ตัวอย่างความต้องการนั้น

**Cost:**

ค่าใช้จ่ายและเวลา ในการทำรายการตรวจสอบความต้องการอยู่ในขั้นต่ำ การทำรายการตรวจสอบความต้องการทำได้ง่าย และคุ้มค่า

**Result Context:**

มีรายการตรวจสอบความต้องการที่สามารถนำมาใช้เป็นตัวตรวจสอบการพัฒนาระบบ เห็นภาพความต้องการโดยรวมของระบบมากขึ้น และง่ายต่อการจัดลำดับความสำคัญความต้องการ

**Related Patterns:**

การจัดลำดับความสำคัญของความต้องการ

การตรวจสอบความต้องการ

## 6. บทสรุป

เราสามารถนำความต้องการกลับมาใช้ใหม่ ในลักษณะของแบบรูปได้ ซึ่งเป็นวิธีนำกลับมาใช้ใหม่ทั้งองค์ความรู้และองค์ประกอบ ซึ่งการนำแบบรูปมาใช้ ช่วยทำให้การนำความต้องการกลับมาใช้ใหม่เป็นแบบแผนและชัดเจนมากขึ้น ง่ายและมีประสิทธิภาพ สามารถนำไปช่วยในกระบวนการพัฒนาซอฟต์แวร์ได้

แบบรูปความต้องการทั้ง 2 ประเภท คือ แบบรูปโดเมนความต้องการ และแบบรูปกระบวนการความต้องการ สามารถนำมาใช้คู่กัน เอื้อประโยชน์ในการปรับใช้กับความต้องการซอฟต์แวร์ของระบบต่างๆ และเสริมการใช้งานของกันและกันได้เป็นอย่างดี

อนึ่งปัจจุบัน มีแบบรูปที่นำไปใช้ในขั้นตอนต่างๆของกระบวนการพัฒนาซอฟต์แวร์ ดังนั้นการใช้แบบรูปการออกแบบร่วมกับแบบรูปอื่นจะทำให้ได้แนวทางในการพัฒนาซอฟต์แวร์ที่สามารถอาศัยความรู้เดิมที่มีอยู่ ทำให้เกิดประสิทธิภาพและพัฒนาซอฟต์แวร์ได้เร็วยิ่งขึ้น

## 7. รายการอ้างอิง


[1] Ralph R. Young, Effective Requirements Practices: Addison-Wesley, 2001.

[2] E. Gramma, R. Helm, R. Johnson and J. Wlissides, Design Patterns Elements of Reusable Object-Oriented Software, 7th ed: Pearson Education, 2002.

[3] C. Alexander, A Pattern Language: Towns, Buildings, Construction: Oxford University Press, 1977.

[4] K. Beck and W. Cunningham, "Using Pattern Languages for Object-Oriented Programs," presented at Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'87), 1987.

[5] M. Fowler, Analysis Pattern: Reuseable Object Models: Addison-Wesley, 1997.

[6] James O. Coplien, Advanced C++ Programming Styles and Idioms: Addison-Wesley, 1991.

[7] S. Robertson, "Requirements Patterns Via Events/Use Case," 1996.

[8] S. Konrad and Betty H.C. Cheng, "Requirements Patterns for Embedded Systems," presented at International Conference on Requirements Engineering, 2002.

[9] R.S. Wahomo and J. Cheng, "Extensible Requirements Patterns of Web Application for Efficient Web Application Development," presented at First International Symposium on Cyber Worlds, 2002.

[10] M. Pantoquilho, R. Raminhos and J. Araujo, "Analysis Patterns Specification: Filling the Gaps," presented at Viking Pattern Language of Program (Viking Plop), 2003.

[11] T. Tsumaki, "Requirements Engineering Patterns Structure," presented at Asia-Pacific Software Engineering Conference, 2004.

[12] S. Fraser, G. Booch, F. Buschmann, J. Coplien, N. Kerth, I. Jacobson and M.B. Rosson, "Patterns: cult to culture?," presented at the 10th annual conference on Object-oriented programming systems languages and applications(OOPSLA). 1995.

[13] IEEE, "Guide to the Software Engineering Body of Knowledge (SWEBOK)," 2004.

[14] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques: John Wiley & Sons, 1997.

[15] G. Meszaros and J. Doble, "A Pattern Language for Pattern Writing," in Pattern languages of program design 3, 1997.

# ภาคผนวก ข

# เอกสารแม่แบบและเอกสารตัวอย่างของแบบรูปความต้องการ

## ข.1 เอกสารเอ็กซ์เอ็มแอลสกีมาซึ่งเป็นเอกสารแม่แบบแบบรูปความต้องการ

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://se.cp.eng.chula.ac.th/repattern"
    targetNamespace="http://se.cp.eng.chula.ac.th/repattern"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
      version="2.0">
  <xs:annotation>
    <xs:documentation>
        <![CDATA[

                This XML Schema used for "Requirements Pattern" data
                in my thesis - Design of Requirement Pattern Structure
for
                Safety-Critical Systems.

           Name: Requirements Pattern Schema
           version: 2.0
           Create Date: 21 March 2007

        ]]>
      </xs:documentation>
  </xs:annotation>
  <xs:complexType name="tCategory">
    <xs:sequence>
      <xs:element name="cat" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tKeywords">
    <xs:sequence>
      <xs:element name="key" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tAgreements">
    <xs:sequence>
      <xs:element name="agree" type="xs:integer"/>
      <xs:element name="disagree" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="tPatternState">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="pattern"/>
      <xs:enumeration value="protopattern"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="tForce">
```

```
    <xs:sequence>
      <xs:element name="F" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="seq" type="xs:integer" use="required"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tRelatedPatterns">
    <xs:sequence>
      <xs:element name="problem" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="pattern" type="xs:IDREF"/>
            <xs:element name="detail" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="seq" type="xs:integer" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tGoal">
    <xs:sequence>
      <xs:element name="G" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="detail" type="xs:string"/>
            <xs:element name="eval" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="seq" type="xs:integer" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tDomainProblem">
    <xs:sequence>
      <xs:element name="DMP" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="problem" type="xs:string"/>
            <xs:element name="type">
              <xs:simpleType>
                <xs:restriction base="xs:NCName">
                  <xs:enumeration value="solve"/>
                  <xs:enumeration value="improve"/>
                  <xs:enumeration value="change"/>
                  <xs:enumeration value="initiate"/>
                  <xs:enumeration value="define"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="detail" type="xs:string"/>
            <xs:element name="from" minOccurs="1" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
```

```
                    <xs:element name="who" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element name="when" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element name="where" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
                    <xs:element name="event" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="rootcause" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tDevelopmentProblem">
      <xs:sequence>
        <xs:element name="DEVP" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="problem" type="xs:string"/>
              <xs:element name="type">
                <xs:simpleType>
                  <xs:restriction base="xs:NCName">
                    <xs:enumeration value="technology"/>
                    <xs:enumeration value="performance"/>
                    <xs:enumeration value="capacity"/>
                    <xs:enumeration value="cost"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="detail" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tBusinessModel">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="reference">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ref" type="xs:string"/>
              <xs:element name="relate" type="xs:string"/>
              <xs:element name="detail" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="word_meaning">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="word" type="xs:string"/>
              <xs:element name="meaning" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
```

```
      </xs:element>
      <xs:element name="diagram">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="description" type="xs:string"/>
            <xs:element name="source" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="seq" type="xs:integer" use="required"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="listDetail">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="topic" type="xs:string"/>
            <xs:element name="description" type="xs:string"/>
            <xs:element name="List">
              <xs:complexType>
                <xs:sequence minOccurs="1" maxOccurs="unbounded">
                  <xs:element name="lst">
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="xs:string">
                          <xs:attribute name="seq" type="xs:integer"
use="required"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="seq" type="xs:integer" use="required"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="topicDetail">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="topic" type="xs:string"/>
            <xs:element name="description" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="seq" type="xs:integer" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="tEnvironments">
    <xs:sequence>
      <xs:element name="ent" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="type">
              <xs:simpleType>
                <xs:restriction base="xs:NCName">
                  <xs:enumeration value="system"/>
                  <xs:enumeration value="organize"/>
                  <xs:enumeration value="peple"/>
                </xs:restriction>
```

```
            </xs:simpleType>
          </xs:element>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tSystemConstrain">
  <xs:sequence>
    <xs:element name="const" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tRequirements">
  <xs:choice minOccurs="1" maxOccurs="unbounded">
    <xs:element name="functionReq">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="from" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="Req_ID" type="xs:ID" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="non-functionReq">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="from" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="type">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="product_req"/>
                <xs:enumeration value="process_req"/>
                <xs:enumeration value="performance_req"/>
                <xs:enumeration value="interface_req"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="Req_ID" type="xs:ID" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
<xs:complexType name="tDomainSolution">
  <xs:sequence>
    <xs:element name="sol" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
```

```
            <xs:element name="topic" type="xs:string"/>
            <xs:element name="detail" type="xs:string"/>
            <xs:element name="deal" type="xs:string"/>
            <xs:element name="conclude" type="xs:string"/>
            <xs:element name="reqs" type="xs:IDREFS"/>
          </xs:sequence>
          <xs:attribute name="seq" type="xs:integer" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tReqRelationshipConstrain">
    <xs:sequence>
      <xs:element name="relationship" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="topic" type="xs:string"/>
            <xs:element name="req">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="req_rel" maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="re_ref" type="xs:IDREF"/>
                        <xs:element name="piority" type="xs:integer"/>
                      </xs:sequence>
                      <xs:attribute name="seq" type="xs:integer"/>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="detail" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="seq" type="xs:integer" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tStakeholderViewpoint">
    <xs:sequence>
      <xs:element name="stakeholder" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="role" type="xs:string"/>
            <xs:element name="views">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="req" minOccurs="1"
maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="re_ref" type="xs:IDREF"/>
                        <xs:element name="view" type="xs:string"/>
                      </xs:sequence>
                      <xs:attribute name="seq" type="xs:integer"/>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
```

```
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tDomainExample">
  <xs:choice>
    <xs:element name="diagram" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="description" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="topic" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="reference" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ref" type="xs:string"/>
          <xs:element name="relate" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="document" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="doc" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
<xs:complexType name="tSafetyAcceptance">
  <xs:sequence>
    <xs:element name="acceptance" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="cataloglevel">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="Low"/>
                <xs:enumeration value="Medium"/>
```

```
                <xs:enumeration value="Hight"/>
                <xs:enumeration value="Serious"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="operatelevel" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tSafetyRequirements">
  <xs:sequence>
    <xs:element name="sre" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="type">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="protect"/>
                <xs:enumeration value="detect"/>
                <xs:enumeration value="react"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="SR_ID" type="xs:ID" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tSafetySignificant">
  <xs:sequence>
    <xs:element name="ssig" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="sreq" type="xs:string"/>
          <xs:element name="probability">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="frequent"/>
                <xs:enumeration value="probable"/>
                <xs:enumeration value="occasional"/>
                <xs:enumeration value="remote"/>
                <xs:enumeration value="improbable"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="severity">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="catastrophic"/>
                <xs:enumeration value="critical"/>
                <xs:enumeration value="marginal"/>
                <xs:enumeration value="negligible"/>
              </xs:restriction>
            </xs:simpleType>
```

```
          </xs:element>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tAccidentIncident">
  <xs:sequence>
    <xs:element name="ac" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="type">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="hardware"/>
                <xs:enumeration value="software"/>
                <xs:enumeration value="human"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tRiskHazard">
  <xs:sequence>
    <xs:element name="rh" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="effect" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tSafetyRelatedExample">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="diagram">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="description">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="topic" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
        </xs:sequence>
```

```
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="reference">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ref" type="xs:string"/>
          <xs:element name="relate" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="document">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="doc" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
<xs:complexType name="tObjective">
  <xs:sequence>
    <xs:element name="obj" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tProcessProblem">
  <xs:sequence>
    <xs:element name="PCP" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="problem" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="rootcause" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tSituation">
  <xs:sequence>
    <xs:element name="sit" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
```

```
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tMethod">
  <xs:sequence>
    <xs:element name="process" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tTools">
  <xs:sequence>
    <xs:element name="tool" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="tool_ref" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tCost">
  <xs:sequence minOccurs="1">
    <xs:element name="money" type="xs:string" minOccurs="0"/>
    <xs:element name="peple" type="xs:string" minOccurs="0"/>
    <xs:element name="skill" type="xs:string" minOccurs="0"/>
    <xs:element name="time" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tProcessExample">
  <xs:choice>
    <xs:element name="diagram" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="description" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="topic" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
```

```
      <xs:element name="reference" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ref" type="xs:string"/>
            <xs:element name="relate" type="xs:string"/>
            <xs:element name="detail" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="document" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="doc" type="xs:string"/>
            <xs:element name="source" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="tBasicPt">
    <xs:sequence>
      <xs:element name="CreateDate" type="xs:date"/>
      <xs:element name="ModifyDate" type="xs:date"/>
      <xs:element name="Author" type="xs:string"/>
      <xs:element name="Category" type="tCategory"/>
      <xs:element name="Keywords" type="tKeywords"/>
      <xs:element name="Comment" type="xs:string"/>
      <xs:element name="Agreements" type="tAgreements"/>
      <xs:element name="PatternState" type="tPatternState"/>
      <xs:element name="PatternName" type="xs:string" nillable="false"/>
      <xs:element name="PatternIntroduction" type="xs:string"
nillable="false"/>
      <xs:element name="Force" type="tForce"/>
      <xs:element name="ResultContext" type="xs:string"/>
      <xs:element name="KnownUse" type="xs:anyType"/>
      <xs:element name="RelatedPatterns" type="tRelatedPatterns"/>
    </xs:sequence>
    <xs:attribute name="Pattern_ID" type="xs:ID" use="required"/>
  </xs:complexType>
  <xs:complexType name="tDomainPt">
    <xs:complexContent>
      <xs:extension base="tBasicPt">
        <xs:sequence>
          <xs:element name="Goal" type="tGoal"/>
          <xs:element name="DomainProblem" type="tDomainProblem"/>
          <xs:element name="DevelopmentProblem"
type="tDevelopmentProblem"/>
          <xs:element name="BusinessModel" type="tBusinessModel"/>
          <xs:element name="Environments" type="tEnvironments"/>
          <xs:element name="SystemConstrain" type="tSystemConstrain"/>
          <xs:element name="Requirements" type="tRequirements"/>
          <xs:element name="DomainSolution" type="tDomainSolution"/>
          <xs:element name="ReqRelationshipConstrain"
type="tReqRelationshipConstrain"/>
          <xs:element name="StakeholderViewpoint"
type="tStakeholderViewpoint"/>
          <xs:element name="ProcessPatterns" type="tRelatedPatterns"/>
          <xs:element name="DomainExample" type="tDomainExample"/>
        </xs:sequence>
```

```
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="tProcessPt">
    <xs:complexContent>
      <xs:extension base="tBasicPt">
        <xs:sequence>
          <xs:element name="Objective" type="tObjective"/>
          <xs:element name="ProcessProblem" type="tProcessProblem"/>
          <xs:element name="Situation" type="tSituation"/>
          <xs:element name="ApplicationArea" type="xs:string"
nillable="false"/>
          <xs:element name="Method" type="tMethod"/>
          <xs:element name="Tools" type="tTools"/>
          <xs:element name="Cost" type="tCost"/>
          <xs:element name="Experiece" type="xs:string"/>
          <xs:element name="ProcessExample" type="tProcessExample"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="tSafetyPt">
    <xs:complexContent>
      <xs:extension base="tDomainPt">
        <xs:sequence>
          <xs:element name="SafetyAcceptance" type="tSafetyAcceptance"/>
          <xs:element name="SafetyRequirements"
type="tSafetyRequirements"/>
          <xs:element name="SafetySignificant" type="tSafetySignificant"/>
          <xs:element name="AccidentIncident" type="tAccidentIncident"/>
          <xs:element name="RiskHazard" type="tRiskHazard"/>
          <xs:element name="SafetyRelatedExample"
type="tSafetyRelatedExample"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="RequirementsPatternSystem">
    <xs:complexType>
      <xs:choice minOccurs="1" maxOccurs="unbounded">
        <xs:element name="DomainPattern" type="tDomainPt"/>
        <xs:element name="ProcessPattern" type="tProcessPt"/>
        <xs:element name="SafetyRelatedPattern" type="tSafetyPt"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## ข.2 ตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่เป็นข้อมูลแบบรูปความต้องการ

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="PatternDocTranslate.xsl"?>
<RequirementsPatternSystem xmlns="http://se.cp.eng.chula.ac.th/repattern"
xsi:schemaLocation="http://se.cp.eng.chula.ac.th/repattern
file:///C:/Documents%20and%20Settings/Media%20Center%20Edition/Desktop/The
sis/pattern%20data/PatternSchema.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DomainPattern Pattern_ID="DPt001">
```

<CreateDate>2006-03-21</CreateDate>
<ModifyDate>2006-03-21</ModifyDate>
<Author>เอกชัย แซ่ตั้ง</Author>
<Category>
<cat>ระบบจำหน่ายตั๋ว</cat>
</Category>
<Keywords>
<key>จำหน่ายตั๋ว</key>
<key>สถานีโดยสาร</key>
</Keywords>
<Comment>สร้างขึ้นเพื่อใช้ในการทดลอง</Comment>
<Agreements>
<agree>3</agree>
<disagree>0</disagree>
</Agreements>
<PatternState>pattern</PatternState>
<PatternName>งานจำหน่ายตั๋วโดยสารที่สถานีโดยสาร</PatternName>
<PatternIntroduction>เป็นระบบจำหน่ายตั๋วโดยสาร โดยเน้นไปที่การจำหน่ายตั๋วแก่ผู้ที่จะเดินทางในขณะนั้น และให้ความสำคัญกับการจำหน่ายตั๋วแบบล่วงหน้าหรือขายแบบชุดรองลงไป การจำหน่ายตั๋วโดยสารจะให้ความสำคัญในแง่ของประสิทธิภาพเพื่อให้ทันต่อความต้องการในการเดินทางในช่วงเวลาหนึ่งๆ ทั้งนี้ข้อมูลการจำหน่ายตั๋วโดยสารยังต้องทำการบันทึกไว้ในระบบเนื่องจากมีความสำคัญเพื่อใช้ในการตรวจสอบยอดจำหน่ายตั๋วโดยสาร และยังสามารถนำไปใช้เพื่อการวิเคราะห์ข้อมูลในด้านต่างๆได้อีกด้วย</PatternIntroduction>
<Force>
<F seq="1">ตั๋วโดยสารทั่วไปมีราคาตามระยะทางการเดินทาง</F>
<F seq="2">รูปแบบของตั๋วมีหลายลักษณะ</F>
</Force>
<ResultContext>ระบบจำหน่ายตั๋วที่รองรับรูปแบบของตั๋วได้หลายรูปแบบ โดยจำกัดเป็นกลุ่มประเภทของตั๋วที่จะมีใช้ในระบบ ระบบจำหน่ายตั๋วที่หน้าสถานีโดยสารโดยใช้เครื่องจำหน่ายตั๋วแบบอัตโนมัติร่วมด้วย ฐานข้อมูลจากการจำหน่ายตั๋วที่นำไปใช้สรุปรายงาน</ResultContext>
<KnownUse>เอกชัย</KnownUse>
<RelatedPatterns>
<problem seq="1">
<pattern>p1</pattern>
<detail>p1-detail</detail>
</problem>
</RelatedPatterns>
<Goal>
<G seq="1">
<detail>การจำหน่ายตั๋วในแต่ละวันจำเป็นต้องมีการบันทึกข้อมูลเพื่อสามารถใช้อ้างอิงได้ว่ามียอดในการจำหน่ายตั๋วโดยสารเท่าไหร่ มีการจำหน่ายไปในแบบใดบ้าง ตลอดจนต้องบันทึกเจ้าหน้าที่ที่รับผิดชอบในการจำหน่ายตั๋วแต่ละส่วนเอาไว้ โดยจะต้องสามารถนำข้อมูลมาสรุปเป็นรายงานยอดจำหน่วยตั๋วได้</detail>
<eval>มีการบันทึกการขายเก็บไว้ และสรุปเป็นรายงานรายวัน รายเดือนได้</eval>
</G>
<G seq="2">

<detail>การจำหน่ายตั๋วโดยเฉพาะในชั่วโมงเร่งด่วนจะต้องสามารถจำหน่ายได้อย่างรวดเร็ว และไม่ควรจะต้องใช้เวลานาน การจำหน่ายตั๋วไม่ว่าจะเป็นประเภทใดจะต้องไม่ยุ่งยากในการใช้ การใช้งานระบบก็ควรจะช่วยให้ทำงานงานจำหน่ายตั๋วโดยสารมีความผิดพลาดลดลงได้</detail>

<eval>จำหน่ายตั๋วได้อย่างมีประสิทธิภาพ ถูกต้อง และรวดเร็ว</eval>

</G>

</Goal>

<DomainProblem>

<DMP seq="1">

<problem>จำหน่ายตั๋วในลักษณะที่แตกต่างกันหลายรูปแบบ</problem>

<type>improve</type>

<detail>ต้องการให้ตั๋วที่จำหน่ายมีหลากหลายรูปแบบตามพฤติกรรมการใช้ของผู้โดยสาร การขายต้องสนับสนุนรูปแบบทั้งหมดและรูปแบบที่อาจมีเพิ่มเติมในอนาคต</detail>

<from>

<who>ฝ่ายวางแผนกลยุทธ์</who>

<event>ต้องพัฒนาระบบให้สนับสนุนรูปแบบการจำหน่ายตั๋วโดยสารแบบใหม่ๆ</event>

</from>

<rootcause>รูปแบบของตั๋วโดยสารมีไม่จำกัด</rootcause>

</DMP>

<DMP seq="2">

<problem>การจำหน่ายตั๋วโดยสารให้รวดเร็วทันความต้องการ</problem>

<type>improve</type>

<detail>ในชั่วโมงเร่งด่วนจำนวนผู้โดยสารมีจำนวนมากส่วนที่จำหน่ายตั๋วนั้นมีอยู่อย่างจำกัด ทำให้การจำหน่ายตั๋วทำได้ไม่ทันกับความต้องการ</detail>

<from>

<who>ผู้โดยสาร</who>

<where>สถานีโดยสารที่เป็นจุดเชื่อมต่อกับระบบขนส่งขนาดใหญ่อื่นๆ จะทำให้จำนวนผู้โดยสารที่มาต่อรถไฟมีมากเป็นช่วงๆ</where>

</from>

<rootcause>จำนวนผู้โดยสารมีมากและไม่สม่ำเสมอ</rootcause>

</DMP>

</DomainProblem>

<DevelopmentProblem>

<DEVP seq="1">

<problem>จะทำการใช้เครื่องจำหน่ายตั๋วอัตโนมัติ</problem>

<type>capacity</type>

<detail>การนำเครื่องจำหน่ายตั๋วแบบอัตโนมัติมาใช้จะช่วยให้ผู้โดยสารสามารถซื้อตั๋วได้ด้วยตัวเอง แต่เครื่องจำหน่ายตั๋วอัตโนมัติที่นำมาใช้อาจไม่เหมาะสมกับผู้โดยสารทุกกลุ่ม ตั๋วบางประเภทยุ่งยากและไม่เหมาะสมในการขายด้วยเครื่องอัตโนมัติ การจำหน่ายตั๋วด้วยเครื่องอัตโนมัติจึงจะต้องพัฒนาซอฟต์แวร์ขึ้นมาเป็นพิเศษให้รองรับลักษณะการทำงานที่เฉพาะเจาะจงได้ ซึ่งจัดได้ว่าเป็นระบบที่คุ้นเคยแค่เฉพาะผู้ที่เคยพัฒนาซอฟต์แวร์สำหรับระบบฝังตัวลักษณะนี้มาก่อนเท่านั้น ดังนั้นทีมพัฒนาจำเป็นต้องศึกษาลักษณะการทำงานของเครื่องจำหน่ายตั๋วอัตโนมัติ ตลอดจนต้องศึกษาคำสั่งต่างๆที่ใช้ในการพัฒนาซอฟต์แวร์สำหรับควบคุมเครื่องซึ่งรวมไปถึงลักษณะการทำงานที่เป็นส่วนต่อประสานผู้ใช้ด้วย</detail>

```
      </DEVP>
      <DEVP seq="2">
        <problem>ไม่สามารถกำหนดประเภทของตั๋วโดยสารที่จะมีในอนาคต</problem>
        <type>cost</type>
        <detail>รูปแบบของตั๋วโดยสารเป็นปัจจัยที่ทำให้ระบบในการจำหน่ายตั๋วมีความซับซ้อนและเกิดการ
ทำงานที่ยุ่งยาก ซึ่งจากลักษณะของระบบทำให้ไม่สามารถกำหนดประเภทของตั๋วและจำกัดลักษณะได้อย่างชัดเจน การ
จะพัฒนาระบบให้สามารถสนับสนุนการจำหน่ายตั๋วได้ทุกประเภทจึงทำได้ยาก อาจมีค่าใช้จ่ายในการพัฒนาสูงกว่าปกติ
</detail>
      </DEVP>
    </DevelopmentProblem>
    <BusinessModel>
      <listDetail seq="1">
        <topic>รูปแบบของตั๋วโดยสารที่มีการจำหน่าย</topic>
        <description>รูปแบบตั๋วโดยสารสำหรับใช้ในการโดยสารในระบบรถไฟมีรูปแบบที่สามารถจำแนก
ดังที่แสดง แต่สำหรับการนำมาใช้จริงอาจไม่มีความจำเป็นต้องจำหน่ายตั๋วโดยสารในทุกรูปแบบ</description>
        <List>
          <lst seq="1">ตั๋วเที่ยวเดียว (ไปอย่างเดียว)</lst>
          <lst seq="2">ตั๋วไปกลับ (ไปกลับตามสถานีโดยสารที่กำหนด)</lst>
          <lst seq="3">ตั๋วรายวันไม่จำกัด (ใช้ไม่จำกัดใน 1 วัน)</lst>
          <lst seq="4">ตั๋วรายวันจำกัด (ใช้ได้หลายครั้งใน 1 วัน แต่ใช้ได้ไม่เกินเที่ยวที่
กำหนด)</lst>
          <lst seq="5">ตั๋วรายสัปดาห์ รายเดือน ฯลฯ (ตามแต่จะกำหนดช่วงอายุของตั๋ว)</lst>
          <lst seq="6">ตั๋วตามจำนวนเที่ยว</lst>
          <lst seq="7">ตั๋วแบบเติมเงินหรือเติมเที่ยวได้</lst>
        </List>
      </listDetail>
      <topicDetail>
        <topic>ลักษณะการซื้อตั๋วจากเครื่องจำหน่ายอัตโนมัติ</topic>
        <description>การซื้อจากเครื่องจำหน่ายตั๋วอัตโนมัติจะเริ่มจากการเลือกสถานีโดยสาร แล้วจำนวน
ค่าโดยสารจะแจ้งให้ทราบ ในบางกรณีจะมีให้ระบุจำนวนตั๋วที่ซื้อด้วยว่ากี่ใบ รูปแบบการชำระค่าโดยสารของเครื่อง
จำหน่ายตั๋วจะขึ้นอยู่กับคุณสมบัติของเครื่อง ซึ่งอาจรวมไปถึงการชำระค่าโดยสารผ่านบัตรเครดิตด้วย
</description>
      </topicDetail>
      <topicDetail>
        <topic>ลักษณะการซื้อตั๋วจากพนักงาน</topic>
        <description>การซื้อตั๋วโดยสารจากพนักงานจำหน่ายตั๋ว จะขึ้นอยู่กับผู้โดยสารว่าจะแจ้งซื้อตั๋วใน
ลักษณะใด หรือเป็นหน้าที่ของพนักงานจำหน่ายตั๋วที่จะช่วยให้คำแนะนำในการซื้อตั๋ว อย่างไรก็ดีตั๋วโดยสารบาง
ประเภทจะไม่มีการเตรียมตั๋วโดยสารไว้ก่อน เช่นตั๋วแบบเติมเงิน ตั๋วแบบเที่ยวเดียว ซึ่งจะเป็นการใช้ระบบออกตั๋วแบบ
ทันทีในขณะที่ทำการซื้อตั๋วผ่านระบบคอมพิวเตอร์</description>
      </topicDetail>
    </BusinessModel>
    <Environments>
      <ent seq="1">
```

```
        <type>organize</type>
        <detail>หน่วยงานที่มีหน้าที่ดูแลระบบขนส่งมวลชนจะเป็นผู้กำหนดกรอบของราคา และส่งผลถึง
ประเภทของบัตรที่จะมีการนำมาใช้ ซึ่งอาจกำหนดเป็นบัตรที่ใช้ร่วมกับระบบขนส่งอื่นๆด้วย</detail>
      </ent>
      <ent seq="2">
        <type>peple</type>
        <detail>บุคคลบางกลุ่มจะมีสิทธิพิเศษในการซื้อตั๋วโดยสาร เช่น กลุ่มนักเรียนนักศึกษา กลุ่มผู้สูงอายุ
เป็นต้น</detail>
      </ent>
    </Environments>
    <SystemConstrain />
    <Requirements>
      <functionReq Req_ID="F1">
        <from>ผู้จัดการฝ่ายขาย</from>
        <detail>จะต้องสามารถใช้ในการจำหน่ายตั๋วได้</detail>
      </functionReq>
      <functionReq Req_ID="F2">
        <from>ผู้จัดการฝ่ายขาย</from>
        <detail>จะต้องสนับสนุนการจำหน่ายตั๋วหลายประเภท</detail>
      </functionReq>
      <functionReq Req_ID="F3">
        <from>ผู้จัดการฝ่ายการตลาด</from>
        <detail>จะต้องมีการบันทึกข้อมูลการขายเก็บไว้เสมอ</detail>
      </functionReq>
      <functionReq Req_ID="F4">
        <from>ผู้จัดการฝ่ายขาย</from>
        <detail>จะต้องใช้ในการสรุปยอดจำหน่ายตั๋วได้</detail>
      </functionReq>
      <non-functionReq Req_ID="NF1">
        <from>ผู้จัดการฝ่ายขาย</from>
        <detail>ข้อมูลจากการจำหน่ายตั๋วต้องมีความถูกต้อง</detail>
        <type>product_req</type>
      </non-functionReq>
      <non-functionReq Req_ID="NF2">
        <from>ผู้จัดการฝ่ายการตลาด</from>
        <detail>ข้อมูลการจำหน่ายตั๋วต้องสามารถเชื่อมโยงกับฐานข้อมูลเพื่อใช้ในการวิเคราะห์ได้
</detail>
        <type>interface_req</type>
      </non-functionReq>
      <non-functionReq Req_ID="NF3">
        <from>ผู้จัดการฝ่ายขาย</from>
        <detail>การจำหน่ายตั๋วต้องทำได้ทันกับความต้องการซื้อตั๋ว</detail>
        <type>performance_req</type>
      </non-functionReq>
      <non-functionReq Req_ID="NF4">
        <from>ผู้จัดการฝ่ายขาย</from>
```

```
        <detail>จะต้องรองรับจำนวนผู้โดยสารที่มาใช้งานได้อย่างเพียงพอ</detail>
        <type>product_req</type>
      </non-functionReq>
    </Requirements>
    <DomainSolution>
      <sol seq="1">
        <topic>จำนวนตั๋วมีหลายรูปแบบ</topic>
        <detail>จากปัญหาที่ตั๋วโดยสารมีด้วยกันหลายประเภท และอาจมีการปรับเปลี่ยนหรือเพิ่มรูปแบบตั๋ว
โดยสารสำหรับอนาคตได้ ทำให้เกิดปัญหาตามมาในหลายลักษณะ</detail>
        <deal>จำกัดรูปแบบที่เป็นไปได้ โดยจัดกลุ่มตามลักษณะคุณสมบัติของตั๋วโดยสาร ซึ่งหากบัตรประเภท
ใดมีลักษณะการใช้งานคล้ายกันก็จะจัดให้อยู่ในกลุ่มเดียวกัน โดยชนิดของตั๋วโดยสารจะยอมให้มีได้ไม่จำกัดแต่จะต้อง
อยู่ภายใต้กลุ่มของรูปแบบตั๋วโดยสารที่กำหนดไว้แต่แรกเท่านั้น</deal>
        <conclude>จำกัดกลุ่มของตั๋วโดยสารซึ่งแยกประเภทตามคุณสมบัติของตั๋วโดยสารที่คล้ายกัน
</conclude>
        <reqs>F2</reqs>
      </sol>
      <sol seq="2">
        <topic>ผู้โดยสารมีจำนวนมาก งานจำหน่ายตั๋วอาจไม่เพียงพอและทันต่อความต้องการ</topic>
        <detail>ตามที่มีบางช่วงเวลาและบางเหตุการณ์ทำให้ผู้ที่มาซื้อตั๋วมีจำนวนมาก และเกรงว่างานจำหน่าย
ตั๋วโดยสารอาจไม่เพียงพอต่อความต้องการ คงไม่สามารถหาวิธีการที่มีประสิทธิผลร้อยเปอร์เซ็นต์มาแก้ปัญหาใน
ลักษณะนี้ แต่ก็น่าที่จะประมาณการ หรือมีแผนหรือระบบรองรับลักษณะที่จะเกิดขึ้น</detail>
        <deal>เป็นไปได้ที่จะใช้ simulation มาช่วยในการประเมิน รวมทั้งควรเอาสถิติผู้เข้ามาซื้อตั๋วแต่
แต่ละลักษณะมาพิจารณา ซึ่งการการจะพัฒนาระบบให้สามารถรองรับการทำงานเต็มร้อยเปอร์เซ็นต์คงไม่สามารถ
เป็นไปได้ แต่ระดับที่น่าจะยอมรับได้ซึ่งควรใช้กำหนดเป็นคุณสมบัติของงานจำหน่ายตั๋วคือ 60-70% ของผู้มาซื้อตั๋ว
โดยสารจะต้องใช้เวลาไม่เกิน 5 นาที และอย่างช้าที่สุดที่ผู้โดยสารจะใช้เวลาในการซื้อตั๋วคือ 20 นาที โดยเพิ่ม
ปริมาณเครื่องจำหน่ายตั๋วและอัตโนมัติและพนักงานจำหน่ายตั๋วเป็นการแก้ปัญหา และอาจอาศัยลักษณะตั๋วที่มีด้วยกัน
หลายประเภทมาใช้ลดเวลาที่จะต้องซื้อตั๋วได้ เช่น ตั๋วแบบรายวัน รายเดือน</deal>
        <conclude>นำการ simulation มาช่วย เพื่อหาวิธีให้สามารถจัดจำหน่ายตั๋วโดยสารได้อย่าง
เพียงพอ ด้วยการประกันผลที่ได้</conclude>
        <reqs>NF4</reqs>
      </sol>
      <sol seq="3">
        <topic>การบันทึกข้อมูลการจำหน่ายตั๋ว</topic>
        <detail>ข้อมูลที่ได้จากการจำหน่ายตั๋วจำเป็นต้องมีการบันทึกเก็บเอาไว้ แต่เนื่องจากรูปแบบของการ
จำหน่ายตั๋วมีหลายลักษณะไม่ว่าจะเป็นการขายโดยปกติ การขายด้วยเครื่องจำหน่ายตั๋วอัตโนมัติ รวมทั้งตั๋วที่มีจัด
จำหน่ายนอกเหนือจากในสถานีโดยสาร ข้อมูลที่จะต้องบันทึกนอกจากจำนวนเงินในการจำหน่ายตั๋วจำนวนตั๋วที่ขาย
อาจจะต้องทำการบันทึกข้อมูลอื่นๆเพิ่มเติมอีก เช่น พนักงานจำหน่ายตั๋ว ช่วงเวลาที่มีการจำหน่ายตั๋ว</detail>
        <deal>การบันทึกข้อมูลควรบันทึกเท่าที่จำเป็นเท่านั้น ไม่เช่นนั้นข้อมูลที่จะจัดเก็บจะมีจำนวนมากเกินไป
เช่นการเก็บบันทึกเวลาที่ทำการจำหน่ายตั๋วซึ่งไม่มีความจำเป็นเพียงพอ ซึ่งหนทางที่ดีและน่าจะยอมรับได้ คือการเก็บ
```

ข้อมูลโดยแจกแจงตามประเภทของรูปแบบการจำหน่าย โดยบันทึกผลลัพท์กิ่งผลสรุป เช่น จำนวนตั๋วโดยสารที่จำหน่ายผ่านเครื่องจำหน่ายตั๋วอัตโนมัติในช่วงเวลา 8.00 - 8.30น. เป็นต้น</deal>

<conclude>ทำการบันทึกข้อมูลเพียงบางส่วน จะไม่บันทึกข้อมูลทั้งหมดโดยละเอียด แต่ก็ยังคงแจกแจงข้อมูลที่มีที่มาต่างกันถือเป็นข้อมูลต่างชุดกัน</conclude>

```
        <reqs>F3</reqs>
      </sol>
    </DomainSolution>
    <ReqRelationshipConstrain>
      <relationship seq="1">
        <topic>การรองรับผู้โดยสารที่มีความหลากหลายเป็นจำนวนมาก</topic>
        <req>
          <req_rel>
            <re_ref>NF4</re_ref>
            <piority>1</piority>
          </req_rel>
          <req_rel>
            <re_ref>F2</re_ref>
            <piority>2</piority>
          </req_rel>
        </req>
        <detail>การจำหน่ายตั๋วจำเป็นต้องทันต่อความต้องการ แม้ว่าตั๋วอาจมีหลายลักษณะซึ่งการขายตั๋วแต่ละรูปแบบอาจมีความแตกต่างกัน ระบบจำหน่ายตั๋วก็จะต้องยึดถือและสนับสนุนการจำหน่ายตั๋วที่มีนัยของจำนวนตั๋วที่ขายเป็นสำคัญ</detail>
      </relationship>
      <relationship seq="2">
        <topic>การบันทึกข้อมูลและสรุปรายงาน</topic>
        <req>
          <req_rel>
            <re_ref>F3</re_ref>
            <piority>1</piority>
          </req_rel>
          <req_rel>
            <re_ref>F4</re_ref>
            <piority>1</piority>
          </req_rel>
        </req>
        <detail>ข้อมูลที่ทำการบันทึกและรายงานสรุปผลต้องมีความสอดคล้องกัน ซึ่งความต้องการเหล่านี้ต้องพิจารณาไปคู่กัน เพื่อสามารถเก็บข้อมูลได้สอดคล้องกับข้อมูลที่จะนำมาเป็นรายงาน และเมื่อพิจารณาจากรายงานก็จะช่วยให้เห็นข้อมูลที่จะต้องจัดเก็บ</detail>
      </relationship>
    </ReqRelationshipConstrain>
    <StakeholderViewpoint>
      <stakeholder seq="1">
        <role>ผู้โดยสาร</role>
        <views>
          <req>
            <re_ref>F1</re_ref>
            <view>ไม่ต้องการที่จะต่อคิวซื้อตั๋วนาน และหากเป็นไปได้ก็ต้องการใบเสร็จในการซื้อตั๋วโดยสารด้วย</view>
```

```
          </req>
          <req>
            <re_ref>F2</re_ref>
            <view>การซื้อตั๋วโดยสารทุกประเภทน่าจะทำได้ผ่านจุดจำหน่ายตั๋วจุดเดียวกัน</view>
          </req>
        </views>
      </stakeholder>
      <stakeholder seq="2">
        <role>พนักงานจำหน่ายตั๋วที่สถานีโดยสาร</role>
        <views>
          <req>
            <re_ref>F1</re_ref>
            <view>ระบบในการจำหน่ายตั๋วควรจะใช้งานง่าย และควรสามารถแก้ปัญหาเมื่อมีการยกเลิกการ
ขายที่ผิดพลาดหรือคืนตั๋วได้</view>
          </req>
          <req>
            <re_ref>F2</re_ref>
            <view>การที่มีตั๋วโดยสารหลายประเภท ระบบควรจะนำมาช่วยในจุดนี้เพื่อลดความสับสนในการ
จำหน่ายตั๋วได้ และไม่ว่าจะเป็นการจำหน่ายตั๋วแบบใดการใช้งานก็ไม่ควรจะแตกต่างกันมากนัก</view>
          </req>
          <req>
            <re_ref>F3</re_ref>
            <view>ในการบันทึกข้อมูลจากการจำหน่ายตั๋ว การบันทึกและผลของการบันทึกน่าจะแสดงออกมา
ให้ทราบด้วย อาจแสดงสรุปยอดขายจากข้อมูลที่บันทึกระหว่างการจำหน่ายตั๋วด้วย</view>
          </req>
        </views>
      </stakeholder>
      <stakeholder seq="3">
        <role>เจ้าหน้าที่วางแผนระบบขนส่ง</role>
        <views>
          <req>
            <re_ref>F3</re_ref>
            <view>การบันทึกข้อมูลต้องคงความถูกต้อง หากระบบมีปัญหาข้อมูลที่มีการบันทึกควรจะต้องคง
อยู่</view>
          </req>
          <req>
            <re_ref>NF4</re_ref>
            <view>ระบบจะต้องรองรับผู้โดยสารในช่วงปกติได้โดยไม่มีปัญหา และในช่วงที่มีผู้โดยสารมาก
เป็นพิเศษการทำงานของระบบจะต้องเพียงพอ และควรมีการล่าช้าได้เล็กน้อยเท่านั้น</view>
          </req>
          <req>
            <re_ref>F4</re_ref>
            <view>การสรุปยอดการจำหน่ายตั๋วจะต้องเป็นไปตามมุมมองที่สามารถนำมาใช้วิเคราะห์ระบบ
ขนส่งได้ โดยข้อมูลที่ได้ต้องมีความถูกต้องครบถ้วนตามที่แจ้งรูปแบบรายงาน</view>
          </req>
        </views>
      </stakeholder>
    </StakeholderViewpoint>
    <ProcessPatterns />
```

```
    <DomainExample />
  </DomainPattern>
  <ProcessPattern Pattern ID="PPt001">
    <CreateDate>2006-03-21</CreateDate>
    <ModifyDate>2006-03-21</ModifyDate>
    <Author>เอกชัย แซ่ตั้ง</Author>
    <Category>
      <cat>elicitation</cat>
    </Category>
    <Keywords>
      <key>define environment</key>
      <key>กำหนดสภาพแวดล้อม</key>
    </Keywords>
    <Comment>สร้างขึ้นเพื่อใช้ในการทดลอง</Comment>
    <Agreements>
      <agree>3</agree>
      <disagree>0</disagree>
    </Agreements>
    <PatternState>pattern</PatternState>
    <PatternName>Define Environments</PatternName>
    <PatternIntroduction />
    <Force>
      <F seq="1">ไม่มีการกำหนดระบบหรือสภาพแวดล้อมที่เกี่ยวข้องกับระบบอย่างแท้จริง</F>
      <F seq="2">มีความเป็นไปได้ว่าจะมีการเปลี่ยนแปลงในอนาคต</F>
    </Force>
    <ResultContext>ระบบที่พัฒนามีความชัเจนมากขึ้นในขอบเขตและข้อจำกัดในการพัฒนา สามารถ
ประเมินความต้องการตามลักษณะสภาพแวดล้อมที่กำหนดขึ้น มีความชัดเจนและที่มาที่ไปในเรื่องของผู้กำหนดและ
จุดประสงค์ของสภาพแวดล้อมของระบบ ระบบที่สร้างขึ้นตามสภาพแวดล้อมที่กำหนดจะสามารถทำงานเข้ากันได้กับส่วน
อื่นๆหรือระบบอื่นๆที่เกี่ยวข้อง</ResultContext>
    <KnownUse />
    <RelatedPatterns />
    <Objective>
      <obj seq="1">เพื่อกำหนดสภาแวดล้อม ซึ่งยังไม่ได้มีการกำหนดไว้</obj>
      <obj seq="2">เพื่อลดความเปลี่ยนแปลงของคุณลักษณะของสภาพแวดล้อม</obj>
      <obj seq="3">เพื่อกำหนดสภาพแวดล้อมที่เหมาะสมในการพัฒนา</obj>
    </Objective>
    <ProcessProblem>
      <PCP seq="1">
        <problem>จะกำหนดสภาพแวดล้อมได้อย่างไร</problem>
        <detail>เนื่องจากสภาพแวดล้อมอาจไม่ได้มีการกำหนดไว้เมื่อเริ่มต้นการพัฒนา การกำหนดสภพ
แวดล้อมของระบบจึงตกเป็นหน้าที่ของผู้พัฒนาในการเก็บรายละเอียดของความต้องการเพื่อกำหนดเป็นสภาพแวดล้อมที่
เหมาะสม ซึ่งการที่จะกำหนดสภาพแวดล้อมขึ้นมานั้นอาจได้รับสิทธิ์เต็มที่จากผู้สนับสนุนโครงการซึ่งก็จะเป็นเรื่องง่าย
ของทีมพัฒนา แต่หากบางส่วนผู้สนับสนุนโครงการหรือผู้ใช้งานมีการกำหนดสภาพแวดล้อมบางอย่างเอาไว้ การจะ
สรุปรายละเอียดของสภาพแวดล้อมสำหรับระบบจึงเป็นปัญหาที่กระทบไปยังส่วนอื่นๆตลอดการพัฒนาระบบ
</detail>
```

<rootcause>ยังไม่มีความชัดเจนในการกำหนดสภาพแวดล้อม และอาจไม่ได้รับสิทธิ์และอำนาจเต็มในการตัดสินใจเลือกสภาพแวดล้อม</rootcause>
</PCP>
<PCP seq="2">
<problem>จะต้องพิจารณาสภาพแวดล้อมในด้านใดบ้าง</problem>
<detail>การพัฒนาระบบจะมีส่วนของสภาพแวดล้อมที่มีส่วนที่เกี่ยวเนื่องในด้านต่างๆจำนวนมาก ทำให้เกิดปัญหาว่ามีจุดใดบ้องของสภาพแวดล้อมที่ควรจะกำหนดเพื่อนำมาใช้ การเก็บความต้องการอาจไม่ได้รับรายละเอียดที่ครบถ้วนหากผู้ให้รายละเอียดไม่ได้พิจารณาครอบคลุมปัจจัยของสภาพแวดล้อมทั้งหมดที่มีผลเกี่ยวเนื่อง
</detail>
<rootcause>ไม่สามารถแจกแจงให้ครอบคลุมสภาพแวดล้อมที่จะทำการกำหนดได้</rootcause>
</PCP>
<PCP seq="3">
<problem>จะต้องกำหนดสภาพแวดล้อมให้ครอบคลุมระดับไหน เพื่อประโยชน์ในแง่ใดบ้าง
</problem>
<detail>การกำหนดสภาพแวดล้อมมักจะพิจารณาเพียงแค่เหตุการณ์เฉพาะหน้า และช่วงต้นของการดำเนินการพัฒนาระบบ ทำให้การกำหนดสภาพแวดล้อมมีปัญหาหลังจากผ่านไปในระยะเวลาหนึ่ง การกำหนดสภาพแวดล้อมในหลายๆครั้งก็ไม่ก่อให้เกิดประโยชน์ต่อการพัฒนา บ่อยครั้งที่การกำหนดสภาพแวดล้อมไม่ได้เล็งเห็นผลประโยชน์ในการนำไปใช้ ขอบเขตที่กำหนดก็มักจะครอบคลุมแค่สิ่งผู้เก็บความต้องการนั้นมองเห็นเท่านั้น เช่น การละเลยการกำหนดสะภาพแวดล้อมเพื่อให้เอื้อไปถึงการพัฒนาและทดสอบระบบด้วย</detail>
<rootcause>ขาดมุมมองขนาดใหญ่ในการกำหนดสภาพแวดล้อมเพื่ออำนวยประโยชน์ไปยังส่วนอื่นๆของการพัฒนาซอฟต์แวร์</rootcause>
</PCP>
</ProcessProblem>
<Situation>
<sit seq="1">ระบบไม่มีการกำหนดสภาพแวดล้อมที่ชัดเจนในเบื้องต้น</sit>
<sit seq="2">ระบบมีความเป็นไปได้ที่จะมีความเปลี่ยนแปลงเกิดขึ้นในอนาคต</sit>
<sit seq="3">ไม่ทราบรายละเอียดของระบบหรือส่วนประกอบอื่นๆที่ระบบจะต้องทำงานร่วมด้วย
</sit>
</Situation>
<ApplicationArea>ระบบทั่วไปที่สามารถพัฒนาในสภาพแวดล้อมที่มีให้เลือกหรือเป็นไปได้หลายรูปแบบ
</ApplicationArea>
<Method>
<process seq="1">กำหนดคุณลักษณะของฮาร์ดแวร์</process>
<process seq="2">กำหนด operating System</process>
<process seq="3">กำหนด Library ของ software</process>
<process seq="4">กำหนดเวอร์ชันของ OS และ Library</process>
<process seq="5">ตรวจสอบความเข้ากันได้และความเหมาะสมของ platform
(Hardware+OS+Library)</process>
<process seq="6">พิจารณาและเลือก interface ที่เกี่ยวข้อง เช่น database driver,
ERP, Protocal</process>
<process seq="7">จัดแจง interface ให้สามารถทำงานประสานกันได้</process>

<process seq="8">กำหนด software ที่จะมีการใช้งานเป็นพื้นฐานของระบบ เช่น ระบบต้องมีซอฟต์แวร์อ่านเอกสาร .pdf</process>

<process seq="9">กำหนดเวอร์ชันของซอฟต์แวร์ที่นำมาใช้ให้เหมาะสมกับระบบ</process>

<process seq="10">ตรวจสอบข้อมูลระบบส่วนอื่นๆที่เกี่ยวข้องกับการทำงาน</process>

<process seq="11">กำหนดสภาพการทำงาน (ในระบบที่เกี่ยวข้องกับลักษณะการทำงานด้วย)</process>

```
    </Method>
    <Tools />
    <Cost>
      <money>มีค่าใช้จ่ายน้อย</money>
      <peple>ต้องใช้ทีมงานที่วางโครงสร้างสถาปัตยกรรมระบบเป็นผู้กำหนด</peple>
      <skill>ใช้ความสามารถในการวางโครงสร้างสถาปัตยกรรมระบบ</skill>
      <time>ใช้เวลาไม่มากนัก</time>
    </Cost>
    <Experiece>
```

การกำหนดสภาพแวดล้อมต้องพิจารณาไปถึงปัจจัยที่จะเข้ามามีบทบาททำให้ระบบการเปลี่ยนแปลงด้วย ซึ่งนอกจากจะกำหนดตอนต้นของการพัฒนาโดยมีจุดประสงค์หลังให้สามารถพัฒนาระบบได้ง่ายสภาพแวดล้อมมีความชัดเจน ซึ่งก็มักจะกำหนดสภาพแวดล้อมให้มีความยืดหยุ่นในระดับหนึ่ง เผื่อว่าระบบเปลี่ยนแปลงไปตามลักษณะที่คาดว่าเป็นไปได้ที่จะเกิดขึ้น สภาพแวดล้อมที่ได้พิจารณาเผื่อไว้แล้วจะได้สามารถรองรับได้โดยไม่มีการเปลี่ยนแปลง ปัญหาสำคัญในการกำหนดสภาพแวดล้อมคือต้องคุยกับเจ้าของโครงการให้เข้าใจว่าเหตุใดต้องการสภาพแวดล้อมแบบนี้ และเหตุใดจึกกำหนดแบบนี้ไม่ได้ ทั้งนี้บางครั้งการกำหนดบางอย่างอาจถูกกำหนดไว้ตั้งแต่การเริ่มโครงการแล้ว ซึ่งก็ควรที่จะกลับไปแก้ไขการกำหนดตอนต้นเสีย แม้อาจจะดูไม่สมควรแต่ด้วยเหตุผลก็ควรจะสามารถทำได้เพื่อให้เกิดประโยชน์ร่วมทั้งสองส่วนและในระยะยาว

```
    </Experiece>
    <ProcessExample />
  </ProcessPattern>
  <SafetyRelatedPattern Pattern_ID="SPt001">
    <CreateDate>2006-03-21</CreateDate>
    <ModifyDate>2006-03-21</ModifyDate>
    <Author>เอกชัย แซ่ตั้ง</Author>
    <Category>
      <cat>ATP</cat>
      <cat>Train Control</cat>
    </Category>
    <Keywords />
    <Comment>สร้างขึ้นเพื่อใช้ในการทดลอง</Comment>
    <Agreements>
      <agree>3</agree>
      <disagree>0</disagree>
    </Agreements>
    <PatternState>pattern</PatternState>
    <PatternName>ATP: Safety Train separation</PatternName>
    <PatternIntroduction>Safety train separate เป็นการกำหนดระยะห่างหรือช่องว่างที่
```

สามารถป้องกันหรือลดความเสี่ยงจากอุบัติภัยที่เกิดขึ้น โดยเป็นการตั้งค่าเพื่อกำหนดการเว้นระยะขบวนที่รองรับขบวน

รูปแบบต่างๆ ซึ่งจะทำงานสอดคล้องกับระบบที่เกี่ยวข้องจากภายนอกหรือแม้แต่รถไฟที่ไม่อยู่ในการควบคุมด้วยระบบอัตโนมัติ โดยการทำงานจะอิงตามรูปแบบการหยุดรถอย่างปลอดภัย</PatternIntroduction>

<Force>
<F seq="1">ระบบรถไฟที่ควบคุมตัวรถได้แบบอัตโนมัติหรือกึ่งอัตโนมัติ</F>
<F seq="2">เป็นระบบที่ใช้การบริหารรางโดยการอิงระยะห่างของราง (ไม่ใช่รางแบบแบ่งช่วงจองการใช้งาน)</F>
</Force>
<ResultContext>ระบบที่เว้นระยะรถไฟตามความเร็วและตำแหน่งในขณะเวลานั้น ทำให้ใช้รางได้อย่างมีประสิทธภาพ ระบบซึ่งสามารถหยุดหรือชะลอความเร็วของรถไฟโดยอัตโนมัติเพื่อให้เกิดความปลอดภัยในระยะการเคลื่อนที่
</ResultContext>
<KnownUse />
<RelatedPatterns />
<Goal>
<G seq="1">
<detail>การเว้นระยะห่างที่เหมาะสมที่จะทำให้เกิดความปลอดภัยจากระยะที่จะสามารถหยุดรถหรือหลีกเลี่ยงการชนของรถไฟได้ โดยรถไฟจะต้องสามารถไปอยู่ในตำแหน่งปลอดภัยได้เมื่อเกิดเหตุการณ์ใดๆขึ้น โดยอาศัยการกำหนดควบคุมระยะห่างที่จะหลีกเลี่ยงจากเหตุร้ายที่ตามมา</detail>
<eval>ระยะห่างระหว่างรถจะถูกกำหนดให้อยู่ในช่วงที่ปลอดภัยเสมอ</eval>
</G>
<G seq="2">
<detail>เมื่อระบบเกิดเหตุการณ์ใดๆขึ้น ซึ่งเป็นผลให้รถไฟต้องหยุดเพื่อความปลอดภัย เช่น มีรถจอดเสีย หรือเข้าสู่ทางที่ใช้ร่วม ระบบจะต้องคำนวณระยะปลอดภัยในการหยุดรถ และเมื่อเข้าสู่ระยะที่วิกฤตเข้าสู่อันตรายได้ระบบก็จะต้องดำเนินการในส่วนของระบบหยุดรถโดยทันที</detail>
<eval>ระบบจะนำไปสู่การหยุดถได้อย่างทันท่วงที</eval>
</G>
</Goal>
<DomainProblem>
<DMP seq="1">
<problem>ระยะห่างที่เหมาะสมจะพิจาณาจากอะไร</problem>
<type>improve</type>
<detail>การพิจารณาระยะห่างที่เหมาะสมต้องพิจารณาไปตามลักษณะของของอีกสิ่งหนึ่งประกอบ เช่น เป็นรถไฟด้วยกัน เป็นสุดทาง เป็นทางชำรุด และยังมีปัจจัยที่ขึ้นกับตัวรถไฟ ณ ขณะเวลา ความเร็ว และตำแหน่งที่แตกต่างกันออกไปด้วย</detail>
<from>
<who>ฝ่ายวิศวกรที่ออกแบบระบบ</who>
<when>ทำการกำหนดระยะปลอดภัย</when>
</from>
<rootcause>การกำหนดระยะห่างมีปัจจัยาจากหลายส่วน</rootcause>
</DMP>
</DomainProblem>
<DevelopmentProblem>
<DEVP seq="1">

<problem>การคำนวณระยะห่างที่ปลอดภัยควรทำในขณะใด</problem>
<type>performance</type>
<detail>การคำนวณหาระยะห่างที่มีปัจจัยที่เกี่ยวข้องหลายส่วน จึงเป็นเรื่องยากที่จะกำหนดเป็นระยะห่างที่คงตัว ดังนั้นระยะห่างที่ปลอดภัยจึงเกิดจากการคำนวณตามปัจจัยที่ถูกต้องในช่วงเวลานั้นๆ ซึ่งเป็นปัญหาว่าการคำนวณควรจะเกิดขึ้นเมื่อไรหรือตลอดเวลา</detail>
</DEVP>
<DEVP seq="2">
<problem>ระบบข้อมูลและอุปกรณ์ตรวจจับต่างๆจะถูกต้องเพียงพอหรือไม่</problem>
<type>technology</type>
<detail>การคำนวณเกี่ยวกับระยะจะต้องอาศัยข้อมูลจากหลายๆส่วนประกอบกันเพื่อคำนวณระยะห่างที่จะปลอดภัย ข้อมูลที่จะนำมาใช้ในการคำนวณจึงมีความสำคัญมาก และมีความเป็นไปได้ที่จะได้ข้อมูลไม่ครบหรือไม่ถูกต้อง ในการพัฒนาต้องเอาประเด็นเหล่านี้มาใช้ประกอบด้วย ต้องวิเคราะห์ถึงลักษณะของอุปกรณ์และความแม่นยำประกอบด้วย โดยอุปกรณ์ในตรวจวัดค่าบางอย่างอาจต้องการความแม่นยำมากกว่าปกติด้วย</detail>
</DEVP>
</DevelopmentProblem>
<BusinessModel>
<diagram seq="1">
<name>ลักษณะของระยะห่างที่ปลอดภัย</name>
<description>แสดงระยะห่างที่รถไฟจะสามารถหยุดได้อย่างปลอดภัยโดยไม่ชน (มีระยะเพียงพอในการหยุดรถ)</description>
<source>SPt001-01.jpg</source>
</diagram>
<diagram seq="2">
<name>การสั่งการหยุดรถไฟเมื่อถึงจุดเสี่ยง</name>
<description>ระบบจะดำเนินการส่วนของระบบหยุดรถโดยอัตโนมัติเมื่อรถไฟเข้าสู่จุดที่ระยะห่างที่เหลืออาจเกิดอันตราย (ระยะไม่เพียงพอในการหยุเรถ)</description>
<source>SPt001-02.jpg</source>
</diagram>
<topicDetail>
<topic>การพิจารณาระยะหยุดรถ</topic>
<description>ระยะหยุดรถจะพิจารณาโดยอ้างอิงตามรูปแบบการหยุดรถอย่างปลอดภัย(Safety Braking Model)</description>
</topicDetail>
</BusinessModel>
<Environments>
<ent seq="1">
<type>system</type>
<detail>ระบบกำกับการเดินรถไฟ(Train Supervisions) ซึ่งจะเป็นตัวกำหนดการใช้รางการปล่อยรถ ซึ่งมีผลต่อระยะห่างและการใช้รางของรถไฟแต่ละขบวน</detail>
</ent>
</Environments>
<SystemConstrain>
<const seq="1">การทำงานจะสามารถนำไปสู่การหยุดรถได้อย่างทันท่วงที</const>
<const seq="2">ยกเลิกการทำงานในกรณีดำเนินการเพื่อทำให้ระบบที่เสียกลับมาใช้งานได้</const>

<const seq="3">ค่าความไม่แน่นอนของตำแหน่ง ค่าความเร็ว ความคลาดเคลื่อน จะต้องทำมาพิจารณาร่วมด้วย</const>
</SystemConstrain>
<Requirements>
<functionReq Req_ID="F1">
<from>ผู้จัดการระบบ</from>
<detail>จะต้องคำนวณระยะปลอดภัยได้อย่างถูกต้อง</detail>
</functionReq>
<functionReq Req_ID="F2">
<from>ผู้จัดการระบบ</from>
<detail>จะต้องครอบคุมจุดที่มีรถเสียอยู่</detail>
</functionReq>
<functionReq Req_ID="F3">
<from>ผู้จัดการระบบ</from>
<detail>จะต้องทำงานร่วมกับรถไฟที่ไม่อยู่ในระบบได้</detail>
</functionReq>
<functionReq Req_ID="F4">
<from>ผู้จัดการระบบ</from>
<detail>จะต้องควบคุมในจุดที่มีการใช้รางร่วมกันได้</detail>
</functionReq>
<functionReq Req_ID="F5">
<from>ผู้จัดการระบบ</from>
<detail>จะต้องนำไปสู่การหยุดรถหากมีการเคลื่อนที่เกินระยะปลอดภัย</detail>
</functionReq>
<non-functionReq Req_ID="NF1">
<from>ผู้จัดการระบบ</from>
<detail>จะต้องทำงานได้ทั้งกับรถไฟที่มีความยาวแบบตายตัวและแบบไม่แน่นอน</detail>
<type>product_req</type>
</non-functionReq>
<non-functionReq Req_ID="NF2">
<from>ผู้จัดการระบบ</from>
<detail>จะต้องสอดคล้องกับรูปแบบการหยุดรถอย่างปลอดภัย (Safety braking model)</detail>
<type>interface_req</type>
</non-functionReq>
</Requirements>
<DomainSolution>
<sol seq="1">
<topic>การคำนวณระยะห่าง</topic>
<detail>ระยะทางที่ปลอดภัยสำหรับการหยุดรถจะขึ้นอยู่กับตำแหน่งของรถกับวัตถุ ลักษณะราง ความเร็วและความสามารถในการหยุดของรถ ซึ่งการคำนวณหาระยะที่ปลอดภัยจะเป็นการพิจารณาระยะที่เพียงพอในการหยุดในขณะนั้น หรือการประมาณระยะที่ไม่ควรเข้าไปใกล้รถขบวนหน้าจนเกินไป</detail>
<deal>วิธีการหนึ่งที่มักถูกนำมาใช้คือการกำหนดระยะของเส้นทางเป็นช่วงๆ เช่น ช่วงละ 100 เมตร โดยการคำนวณจะคำนวณเมื่อรถไฟเข้าสู่ช่วงถัดไปของทาง ระยะทางก็จะถูกคำนวณจากความเร็วสูงสุด (ตามที่

กำหนดให้วิ่งได้) หรือความเร็วรถไฟขณะนั้นเพื่อคำนวณระยะหยุดรถ และเมื่อเปรียบเทียบระยะทางที่ใช้กับทางของรางก็จะได้ช่วงของรางที่รถจะหยุดได้ทัน โดยหากเกิดกรณีช่วงของรางนั้นมันช่วงทับกับช่วงการหยุดของรถไฟขบวนอื่นหรือไม่อยู่ในช่วงที่ต้องการก็จะถือว่าเกิดกรณีที่จะต้องดำเนินการแจ้งเพื่อดำเนินการหยุดรถทั้งแบบปกติและแบบฉุกเฉิน ในทางปฏิบัติการแบ่งช่วงจะพิจารณาตาม track circuit</deal>

<conclude>พิจารณาโดยแบ่งรางเป็นช่วง คำนวณระยะขณะนั้นเพื่อตรวจสอบ</conclude>
<reqs>F1</reqs>
</sol>
<sol seq="2">
<topic>การกำหนดและควบคุมการตอบสนอง</topic>
<detail>เมื่อรถไฟไม่อยู่ในขีดจำกัดของการกำหนดระยะ นั่นคือระยะและความเร็วเลยขอบเขตที่กำหนดเป็นระยะที่มีความปลอดภัย ระบบจะต้องทำการตอบสนองเพื่อป้องกันผลที่ตามมา</detail>
<deal>ระบบจะต้องทำการตอบสนองเมื่อ ระยะหยุดจวนจะเหลือน้อยกว่าระยะปลอดภัย (ไม่ยอมให้น้อยกว่า) ความเร็วเกินกว่าความเร็วที่ใช้คำนวณตามระยะปลอดภัย และเกิดการกีดขวางฉับพลันในส่วนระยะปลอดภัย โดยระบบจะต้องดำเนินการแจ้งให้ผู้ควบคุมรถไฟทราบในทันที โดยการเตรียมระบบหยุดรถให้พร้อมทำงาน ไม่ยอมให้เพิ่มความเร็ว ดำเนินการลดความเร็ว รวมไปถึงดำเนินการหยุดรถแบบฉุกเฉิน</deal>
<conclude>กำหนดปัจจัยที่ต้องตอบสนองและวิธีการตอบสนองที่จะดำเนินการตาม</conclude>
<reqs>F5</reqs>
</sol>
</DomainSolution>
<ReqRelationshipConstrain>
<relationship seq="1">
<topic>การหยุดรถเมื่อไม่อยู่ในขอบเขตตามรูปแบบการหยุดรถอย่างปลอดภัย</topic>
<req>
<req_rel>
<re_ref>F5</re_ref>
<piority>1</piority>
</req_rel>
<req_rel>
<re_ref>NF2</re_ref>
<piority>1</piority>
</req_rel>
</req>
<detail>การทำงานของระบบจะอิงกับรูปแบบการหยุดรถอย่างปลอดภัย ซึ่งจะเป็นรายละเอียดของระยะที่จะต้องทำการหยุดรถตามความเร็วในขณะนั้น เมื่อเห็นว่าอยู่ในระยะที่ไม่ปลอดภัยแล้ว การหยุดรถก็จะถูกดำเนินการขึ้นในทันที โดยบวกค่าระยะที่เผื่อไว้ตามที่มีการกำหนดด้วย การหยุดรถและการควบคุมตามรูปแบบการหยุดอย่างปลอดภัยจะต้องพิจารณาควบคู่กันไป</detail>
</relationship>
<relationship seq="2">
<topic>การคำนวณระยะที่ปลอดภัยเกี่ยวกับรถไฟที่อยู่และไม่อยู่ในระบบ</topic>
<req>
<req_rel>
<re_ref>F2</re_ref>
<piority>1</piority>
</req_rel>
<req_rel>
<re_ref>F3</re_ref>

```
            <piority>1</piority>
          </req_rel>
          <req_rel>
            <re_ref>F1</re_ref>
            <piority>2</piority>
          </req_rel>
          <req_rel>
            <re_ref>F4</re_ref>
            <piority>2</piority>
          </req_rel>
        </req>
        <detail>เมื่อรางในระบบมีการใช้งานร่วมกับรถไฟอื่นๆที่ไม่ได้ควบคุมโดยระบบ เช่น รถไฟที่
เคลื่อนที่เป็นกรณีพิเศษ รถไฟที่ไม่มีระบบกิ่งอัตโนมัติในการควบคุม เมื่อเป็นรถไฟในระบบข้อมูลต่างๆจะมีความ
สมบูรณ์กว่าและสามารถควบคุมและกำหนดการทำงานของรถไฟได้ แต่เมื่อทำงานร่วมกับรถไฟที่ไม่อยู่ในระบบอาจทำ
ให้เกิดปัญหาในการควบคุมรถไฟที่อยู่นอกระบบ การใช้รางรวมกันจึงต้องให้ความสำคัญกับรถไฟที่อยู่นอกระบบด้วย
และเนื่องจากขาดความสามารถในการจัดการอย่างสมบูรณ์ ในการควบคุมระยะห่างซึ่งไม่สมาถกำหนดที่รถไฟที่อยู่นอก
ระบบได้ รถไฟในระบบจึงต้องได้รับการควบคุมอย่างเคร่งคัดและให้การเว้นระยะเป็นไปตามรถไฟนอกระบบเมื่อมีการ
ใช้รางร่วมกัน</detail>
      </relationship>
    </ReqRelationshipConstrain>
    <StakeholderViewpoint>
      <stakeholder seq="1">
        <role>พนักงานควบคุมไฟ</role>
        <views>
          <req>
            <re_ref>F3</re_ref>
            <view>หากมีรถไฟขบวนอื่นๆที่นอกเนือจากที่ถูกควบคุมอยู่โดยระบบ ระบบควรสามารถบอกให้
ทราบได้ด้ว เนื่องจากการควบคุมรถไฟเหล่านี้เป็นลักษณะ manual จึงอาจจะต้องติต่อกับนย์คบคุมและอาจต้อง
ระมัดระวังเป็นิเศษ</view>
          </req>
          <req>
            <re_ref>NF2</re_ref>
            <view>เมื่อรถไฟกำลังจะเข้าสู่ระยะที่ทำการจำกัดก็ควรจะเตือนในทันที เช่น หากวิ่งด้วยความเร็ว
ปัจจุบันต่อไปอีก 5 นาที จะทำให้รถชนกับรถขบวนข้างหน้า และหากเข้าสู่ระยะที่วิกฤติแล้วระบบก็ควรจะสามารถ
ชะลอความเร็วเองได้ หรือทำการหยุดอย่างกะทันหันโดยอัตโนมัติเมื่อเกิดเหตุการณ์สุดวิสัย</view>
          </req>
        </views>
      </stakeholder>
    </StakeholderViewpoint>
    <ProcessPatterns />
    <DomainExample />
    <SafetyAcceptance>
      <acceptance seq="1">
        <detail>ระยะห่างอยู่ในช่วงไม่ปลอดภัย</detail>
        <cataloglevel>Serious</cataloglevel>
        <operatelevel>ผู้บริหาร</operatelevel>
      </acceptance>
      <acceptance seq="2">
```

```
      <detail>การหยุดหรือชะลอความเร็วโดยอัตโนมัติ</detail>
      <cataloglevel>Hight</cataloglevel>
      <operatelevel>หัวหน้าโครงการ</operatelevel>
    </acceptance>
    <acceptance seq="3">
      <detail>การควบคุมให้รถไฟให้เว้นระยะอย่างเหมาะสม</detail>
      <cataloglevel>Hight</cataloglevel>
      <operatelevel>พนักงานควบคุมรถไฟ</operatelevel>
    </acceptance>
  </SafetyAcceptance>
  <SafetyRequirements>
    <sre SR_ID="SR1">
      <type>protect</type>
      <detail>ระยะห่างระหว่างรถไฟน้อยเกินกว่าจะปลอดภัยจากอุบัติเหตุ</detail>
    </sre>
    <sre SR_ID="SR2">
      <type>protect</type>
      <detail>ระยะจากรถไฟถึงสิ่งกีดขวางหรือจุดสิ้นสุดของรางน้อยเกินกว่าจะหยุดรถได้ทัน</detail>
    </sre>
    <sre SR_ID="SR3">
      <type>detect</type>
      <detail>การเพิ่มความเร็วที่ทำให้ระยะห่างระหว่างรถไฟไม่เพียงพอต่อการหยุดรถ</detail>
    </sre>
    <sre SR_ID="SR4">
      <type>react</type>
      <detail>ระยะที่เหลือไม่เพียงพอสำหรับการเว้นระยะที่ปลอดภัย</detail>
    </sre>
    <sre SR_ID="SR5">
      <type>react</type>
      <detail>การเพิ่มความเร็วที่ทำให้ระยะห่างระหว่างรถไฟไม่เพียงพอต่อการหยุดรถ</detail>
    </sre>
  </SafetyRequirements>
  <SafetySignificant>
    <ssig seq="1">
      <sreq>การคำนวณระยะห่างผิดพลาด</sreq>
      <probability>probable</probability>
      <severity>critical</severity>
    </ssig>
    <ssig seq="2">
      <sreq>ความคลาดเคลื่อนของการวัดความเร็วและตำแหน่ง</sreq>
      <probability>probable</probability>
      <severity>marginal</severity>
    </ssig>
  </SafetySignificant>
  <AccidentIncident>
    <ac seq="1">
      <type>hardware</type>
      <detail>ตัวตรวจจับความเร็วของรถทำงานผิดพลาด</detail>
    </ac>
    <ac seq="2">
      <type>hardware</type>
```

```
        <detail>ตัวตรวจจับตำแหน่งของรถทำงานผิดพลาด</detail>
      </ac>
      <ac seq="3">
        <type>software</type>
        <detail>การคำนวณระยะห่างไม่ถูกต้อง</detail>
      </ac>
      <ac seq="4">
        <type>human</type>
        <detail>ทำการควบคุมรถโดยไม่ระมัดระวัง</detail>
      </ac>
      <ac seq="5">
        <type>software</type>
        <detail>ระบบเตือนการหยุดรถหรือการชะลอรถไม่ทำงาน</detail>
      </ac>
    </AccidentIncident>
    <RiskHazard>
      <rh seq="1">
        <detail>รถไฟชนกัน</detail>
        <effect>รถไฟเสียหายโดยอาจทำให้รถไฟตกรางอีกด้วย ทำให้ผู้โดยสารได้รับอันตรายโดยเฉพาะการ
ชนกันที่มีความเร็วหรือในลักษณะประสานงาหรือตกราง ทำให้ระบบหรือเครือข่ายหยุดชะงัก และอาจก่อให้เกิด
อุบัติเหตุอื่นๆตามมา</effect>
      </rh>
      <rh seq="2">
        <detail>รถไฟชนสิ่งกีดขวางหรือวิ่งเลยราง</detail>
        <effect>รถไฟเสียหายและตกราง ทำให้ผู้โดยสารได้รับอันตราย</effect>
      </rh>
    </RiskHazard>
    <SafetyRelatedExample />
  </SafetyRelatedPattern>
</RequirementsPatternSystem>
```

# ภาคผนวก ค
# คำถามประกอบการทดลอง

## ค.1 แนวทางในการตั้งคำถาม

**ตอนที่ 1 คำถามปรนัย**

ประกอบด้วยคำถามแบบตัวเลือกจำนวน 10 ข้อ แต่ละข้อมี 5 ตัวเลือก (ก. ถึง จ.)

คำถามข้อที่ 1 ทดสอบว่าผู้ตอบคำถามเข้าใจเป้าหมายของงานหรือไม่

คำถามข้อที่ 2 ทดสอบว่าผู้ตอบคำถามเข้าใจปัญหาที่เกิดขึ้นในแง่ของผู้ให้ความต้องการและผู้พัฒนา

คำถามข้อที่ 3 ทดสอบว่าผู้ตอบคำถามสามารถระบุข้อจำกัดของระบบได้

คำถามข้อที่ 4 ทดสอบว่าผู้ตอบคำถามทราบว่าใครคือผู้ใช้และผู้ให้ความต้องการของระบบบ้าง

คำถามข้อที่ 5 ทดสอบว่าผู้ตอบคำถามสามารถหาความต้องการได้

คำถามข้อที่ 6 ทดสอบว่าผู้ตอบคำถามเล็งเห็นปัจจัยและความเกี่ยวข้องอื่นๆในการทำ Requirements

คำถามข้อที่ 7 ทดสอบว่าผู้ตอบคำถามสามารถหาความต้องการความปลอดภัยได้

คำถามข้อที่ 8 ทดสอบว่าผู้ตอบคำถามสามาตรระบุอันตรายที่อาจเกิดขึ้นได้

คำถามข้อที่ 9 ทดสอบว่าผู้ตอบคำถามเข้าใจระดับการยอมรับทางด้านความปลอดภัย

คำถามข้อที่ 10 ทดสอบว่าผู้ตอบคำถามสามารถเลือกเทคนิควิธีการที่จะนำมาใช้ในงานได้เป็นอย่างดี

**ตอนที่ 2 คำถามอัตนัย**

ประกอบด้วยคำถาม 5 ข้อ ดังนี้

คำถามข้อที่ 11 ทดสอบว่าผู้ตอบคำถามเล็งเห็นปัญหาสามารถแก้ปัญหาที่มีอยู่ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

คำถามข้อที่ 12 ทดสอบว่าผู้ตอบคำถามมีความเข้าใจและเห็นความสัมพันธ์ระหว่างความต้องการ

คำถามข้อที่ 13 ทดสอบว่าผู้ตอบคำถามสามารถหาและสรุปต้นเหตุที่นำไปสู่อันตรายได้

คำถามข้อที่ 14 ทดสอบว่าผู้ตอบคำถามสามารถระบุความต้องการที่ไม่ใช่หน้าที่ได้

คำถามข้อที่ 15 ทดสอบว่าผู้ตอบคำถามสามารถเชื่อมโยงไปสู่ขั้นการออกแบบระบบอย่างปลอดภัยได้

**ตอนที่ 3 คำถามที่ใช้ประเมินส่วนเนื้อหาในแบบรูป**

ใช้ในการประเมินความถูกต้องของแบบรูป เมื่อมีการใช้แบบรูปในการทำความต้องการเท่านั้น โดยให้คะแนนเป็น 1 ถึง 5 แทนความหมาย น้อยมาก น้อย ปานกลาง ดี และดีมาก โดยประกอบด้วยคำถาม 5 ข้อ ดังนี้

คำถามข้อที่ 1 ความครบถ้วนสมบูรณ์ของเนื้อหาในแบบรูป

คำถามข้อที่ 2 ความถูกต้องของเนื้อหาในแบบรูป

คำถามข้อที่ 2 ความง่ายในการใช้และทำความเข้าใจแบบรูป

คำถามข้อที่ 3 ความชัดเจนของเนื้อหาในแบบรูป

คำถามข้อที่ 4 ความมีประสิทธิผลจากการใช้งานแบบรูป

**ชุดคำถาม ประเมินและแสดงความเห็นเกี่ยวกับแบบรูป**

คำถามในส่วนนี้จะประเมินความเห็น โดยให้ลำดับความเห็นเป็น น้อยที่สุด น้อย ปานกลาง มาก และมากที่สุด และให้คะแนนการประเมินเป็น 1 ถึง 5 ตามลำดับ

1. ประเมินเพื่อวัดความเห็นต่อการทำงาน โดยใช้เป็นคำถามข้อ 1 ถึง 4
2. ประเมินเพื่อวัดความเห็นเกี่ยวกับโครงสร้างและองค์ประกอบ โดยใช้เป็นคำถามข้อ 5 ถึง 7
3. ประเมินเพื่อวัดความรู้สึกของผู้ใช้งาน โดยใช้เป็นคำถามข้อ 8 ถึง 10

**ค.2 ชุดคำถามที่ 1 Safety Train Separate**

**ตอนที่ 1 จงเลือกคำตอบที่ถูกต้องที่สุดเพียง 1 ข้อ**

1. ข้อใดคือเป้าหมายของการทำงานในส่วนของ Safety train separate
   - ก. กำหนดระยะห่างระหว่างรถไฟ
   - ข. ควบคุมระยะห่างให้สามารถหยุดอย่างปลอดภัย
   - ค. ควบคุมตำแหน่งรถไฟให้มีความปลอดภัย
   - ง. ข้อ ก. และ ข.
   - จ. ข้อ ก. และ ค.
2. ข้อใดไม่ใช่ปัญหาในงานวิศวกรรมความต้องการสำหรับ safety train separate
   - ก. ผู้ให้ความต้องการไม่สามารถระบุปัจจัยที่ใช้เป็นตัวกำหนดระยะห่างได้อย่างสมบูรณ์
   - ข. การกำกับเวลาหรือจังหวะที่จะมีการคำนวณเกี่ยวกับระยะห่าง
   - ค. ความถูกต้องของระบบข้อมูลและตัวอุปกรณ์

ง. ค่าความคลาดเคลื่อนที่เกิดขึ้นในการคำนวณระยะห่าง

จ. ข้อ ข. และ ง.

3. ข้อใดไม่ใช่ข้อจำกัด (System Constraint) ในส่วนของ safety train separate

ก. การทำงานจะนำไปสู่การหยุดรถที่ปลอดภัย

ข. การยอมให้ยกเลิกการทำงานในกรณีที่เกี่ยวข้องกับการทำให้ระบบที่เสียกลับมาใช้งาน

ค. การเว้นระยะห่างเป็นไปตามการกำกับการเดินรถไฟ

ง. พิจารณาค่าความไม่แน่นอนของตำแหน่งรถไฟให้อยู่ในช่วงที่กำหนด

จ. พิจารณาค่าความไม่แน่นอนของความเร็วรถไฟให้อยู่ในช่วงที่กำหนด

4. ข้อใดคือกลุ่มคนที่เกี่ยวข้องกับความต้องการของ Safety train separate

ก. พนักงานควบคุมรถไฟ

ข. พนักงานควบคุมรถไฟ และหัวหน้าโครงการ

ค. พนักงานควบคุมรถไฟ ผู้บริหาร และหัวหน้าโครงการ

ง. พนักงานกำหนดเส้นทาง

จ. พนักงานกำหนดเส้นทาง และหัวหน้าโครงการ

5. ข้อใดไม่จัดว่าเป็นความต้องการของ safety train separate

ก. จะต้องคำนวณระยะปลอดภัยได้อย่างถูกต้อง

ข. จะต้องทำงานร่วมกับรถไฟที่ไม่ได้มีการควบคุมได้

ค. จะต้องทำงานได้ทั้งกับรถไฟที่มีความยาวคงที่และไม่คงที่

ง. จะต้องนำไปสู่การหยุดหรือชะลอรถหากระยะห่างเข้าสู่จุดที่ไม่ปลอดภัย

จ. จะต้องช่วยในการกำหนดขีดความเร็วของรถไฟที่วิ่งได้

6. ข้อใดไม่มีความเกี่ยวข้องกับการพิจารณาความต้องการของ safety train separate

ก. ความยาวรถไฟ

ข. เส้นทางตัดใช้ร่วมกับทางอื่น

ค. สถานีโดยสารหรือชานชาลา

ง. ตำแหน่งสิ้นสุดของราง

จ. ตำแหน่งของรถไฟ

7. ข้อใดคือหน้าที่ที่เกี่ยวข้องกับความปลอดภัยของ safety train separate

ก. ใช้คำนวณระยะห่างระหว่างรถไฟ

ข. ใช้กำหนดระยะที่เพียงพอในการหยุดรถอย่างปลอดภัย

ค. ใช้ควบคุมการใช้รางร่วมกันของรถไฟ

ง. ใช้ป้องกันไม่ให้รถไฟชนกัน

จ. ใช้กำหนดควบคุมความเร็วให้รถไฟให้เหมาะสม

8. ข้อใดเป็นอันตรายที่อาจเกิดขึ้นจากความผิดพลาดของ safety train separate โดยตรง

ก. รถไฟวิ่งเลยจุดสิ้นสุดของราง และรถไฟชนสิ่งกีดขวาง

ข. รถไฟชนกันเอง และรถไฟตกราง

ค. ระยะห่างระหว่างรถไฟน้อยเกินไป

ง. รถไฟชนสิ่งกีดขวาง และรถไฟตกราง

จ. รถไฟวิ่งด้วยความเร็วสูงจนอันตราย

9. ข้อใดคือผลที่ไม่สามารถยอมรับรับได้และมีความสำคัญมากสำหรับ safety train separate

ก. การควบคุมระยะของรถไฟที่ไม่เหมาะสม

ข. การมีระยะห่างรถไฟอยู่ในช่วงอันตราย

ค. การหยุดหรือชะลอความเร็วโดยอัตโนมัติไม่ทำงาน

ง. การคำนวณความเร็วคลาดเคลื่อนจากความเป็นจริง

จ. การคำนวณตำแหน่งคลาดเคลื่อนจากความเป็นจริง

10. ข้อใดต่อไปนี้เป็นเทคนิคที่มีความเหมาะสมเป็นพิเศษในการนำมาใช้ในงานวิศวกรรมความต้องการสำหรับ safety train separate

ก. การบันทึกที่มาของความต้องการ

ข. การวิเคราะห์ลำดับความสำคัญของความต้องการ

ค. การใช้ทีมที่มีความหลากหลายในการตรวจสอบความต้องการ

ง. การแยกแยะและให้คำแนะนำผู้ใช้งานระบบ

จ. การกำหนดวัฒนธรรมด้านความปลอดภัยในองค์กร

**ตอนที่ 2 จงตอบคำถามต่อไปนี้**

11. จงยกตัวอย่างปัญหาในงานวิศวกรรมความต้องการที่พบในระบบที่กล่าวถึง พร้อมทั้งอธิบายวิธีการในการจัดการกับปัญหานั้น

12. จงยกตัวอย่างความต้องการที่มีความสัมพันธ์กันพร้อมทั้งอธิบาย

13. จงยกตัวอย่างความผิดพลาดจาก Software, hardware และ human ที่ส่งผลต่อความปลอดภัย มา 3 ข้อ

14. จงยกตัวอย่างความต้องการที่ไม่ใช่หน้าที่ มา 3 ข้อ

15. จงยกตัวอย่างอันตรายที่นำไปพิจารณาประกอบเมื่อถึงขั้นตอนการออกแบบระบบ มา 3 ข้อ

**ตอนที่ 3 คำถามที่ใช้ประเมินส่วนเนื้อหาในแบบรูป**

<u>ให้ตอบคำถามในตอนที่ 3 เมื่อมีการใช้แบบรูปประกอบการทดลองเท่านั้น</u>

ข้อ 16 – 20 ให้คะแนน 1 – 5 แทนความหมาย น้อยมาก น้อย ปานกลาง ดี และดีมาก ตามลำดับ

16. ท่านคิดว่าแบบรูปที่ท่านใช้มีความครบถ้วนสมบูรณ์ระดับใด
17. ท่านคิดว่าแบบรูปที่ท่านใช้มีความถูกต้องในระดับใด
18. ท่านคิดว่าแบบรูปที่ท่านใช้มีความง่ายในการใช้และทำความเข้าใจระดับใด
19. ท่านคิดว่าแบบรูปที่ท่านใช้มีความชัดเจนในระดับใด
20. ท่านคิดว่าแบบรูปที่ท่านใช้ช่วยแก้ปัญหาการทำความต้องการนี้ในระดับใด

**ค.3 ชุดคำถามที่ 2 Door open control protection interlocks**

**ตอนที่ 1 จงเลือกคำตอบที่ถูกต้องที่สุดเพียง 1 ข้อ**

1. ข้อใดเป็นเป้าหมายของ Door Open Control Protection Interlocks
    - ก. มีการควบคุมการเปิดประตู
    - ข. เปิดประตูด้วยเงื่อนไขที่ปลอดภัย
    - ค. สามารถเปิดประตูในกรณีฉุกเฉินได้
    - ง. ถูกเฉพาะข้อ ก. และ ข.
    - จ. ถูกทุกข้อ
2. ข้อใดคือปัญหาในงานวิศวกรรมความต้องการสำหรับ Door Open Control Protection Interlocks
    - ก. การหาปัจจัยที่เกี่ยวข้องกับความปลอดภัย
    - ข. รูปแบบหรือประเภทของประตู(เปิดทางเดียว เปิดสองทาง)
    - ค. ความยุ่งยากในการกำหนดความสัมพันธ์กับอุปกรณ์ที่เกี่ยวข้องกับการประตู
    - ง. ข้อ ก. และ ข.
    - จ. ข้อ ก. และ ค.
3. ข้อใดคือข้อจำกัด (System Constraint) ในส่วนของ Door Open Control Protection Interlocks
    - ก. ตำแหน่งการเปิดประตูรถ
    - ข. ความเร็วในการเปิดประตู
    - ค. สัญญาณแจ้งการเปิดประตู
    - ง. การเปิดประตูต้องทำขณะที่รถหยุด
    - จ. การเปิดประตูฉุกเฉิน
4. ข้อใดคือกลุ่มคนที่เกี่ยวข้องกับความต้องการของ Door Open Control Protection Interlocks
    - ก. พนักงานควบคุมรถไฟ และผู้โดยสาร
    - ข. พนักงานควบคุมรถไฟ ผู้จัดการระบบ และผู้โดยสาร
    - ค. พนักงานกำหนดเส้นทางเดินรถ พนักงานควบคุมรถไฟ และผู้โดยสาร
    - ง. ผู้จัดการระบบ และผู้โดยสาร

จ. พนักงานประจำสถานีโดยสารรถไฟ พนักงานควบคุมรถไฟ และผู้โดยสาร

5. ข้อใดไม่จัดว่าเป็นความต้องการใน Door Open Control Protection Interlocks
   ก. ต้องสามารถเปิดประตูรถได้
   ข. ต้องสามารถเปิดประตูกรณีฉุกเฉินได้
   ค. ต้องมีการแจ้งการเปิดประตู
   ง. ต้องมีการตรวจสอบความปลอดภัยก่อนจะเปิดประตูได้
   จ. ต้องสามารถควบคุมการเปิดประตูจากภายนอกได้ด้วย
6. ข้อใดไม่มีความเกี่ยวข้องกับการพิจารณาความต้องการของ Door Open Control Protection Interlocks
   ก. ความยาวรถไฟ
   ข. ตำแหน่งจอดเทียบชานชาลา
   ค. ระยะเวลาการจอดรถ
   ง. การเคลื่อนที่ของรถไฟ
   จ. ตำแหน่งการจอดรถไฟ
7. ข้อใดคือหน้าที่ที่เกี่ยวข้องกับความปลอดภัยของ Door Open Control Protection Interlocks
   ก. การกำหนดระยะเวลาการเปิดประตู
   ข. การป้องกันการชนกันของรถไฟ
   ค. การเตือนการเปิดประตู
   ง. การป้องกันการเปิดประตูขณะรถยังวิ่งอยู่
   จ. การเปิดประตูกรณีฉุกเฉิน
8. ข้อใดเป็นอันตรายที่อาจเกิดขึ้นจากความผิดพลาดของ Door Open Control Protection Interlocks โดยตรง
   ก. ประตูหนีบผู้โดยสารที่ผ่านเข้าออก
   ข. ผู้โดยสารออกจากรถไฟบริเวณนอกสถานีโดยสาร
   ค. ประตูเปิดผิดทาง (เช่น ประตูเปิดซ้ายแต่ชานชาลาอยู่ขวา)
   ง. รถไฟหยุดเนื่องจากมีความพยายามในการเปิดประตูฉุกเฉิน
   จ. ผู้โดยสารพิงประตูขณะที่กำลังเปิด
9. ข้อใดคือผลที่ไม่สามารถยอมรับรับได้และมีความสำคัญมากสำหรับ Door Open Control Protection Interlocks
   ก. การเปิดประตูไม่สามารถทำได้
   ข. การเปิดประตูทำให้เกิดอันตราย

ค. รถหยุดอันเนื่องมาจากการเปิดประตูฉุกเฉิน

ง. ตำแหน่งการเปิดประตูไม่ตรงตามที่ชานชาลาระบุไว้

จ. การควบคุมการเปิดประตูจากภายนอกไม่สามารถทำได้

10. ข้อใดต่อไปนี้เป็นเทคนิคที่มีความเหมาะสมเป็นพิเศษในการนำมาใช้ในงานวิศวกรรมความต้องการสำหรับ Door Open Control Protection Interlocks

ก. การบันทึกที่มาของความต้องการ

ข. การวิเคราะห์ลำดับความสำคัญของความต้องการ

ค. การใช้ทีมที่มีความหลากหลายในการตรวจสอบความต้องการ

ง. การแยกแยะและให้คำแนะนำผู้ใช้งานระบบ

จ. รายการตรวจสอบความปลอดภัย

**ตอนที่ 2 จงตอบคำถามต่อไปนี้**

11. จงยกตัวอย่างปัญหาในงานวิศวกรรมความต้องการที่พบในระบบที่กล่าวถึง พร้อมทั้งอธิบายวิธีการในการจัดการกับปัญหานั้น
12. จงยกตัวอย่างความต้องการที่มีความสัมพันธ์กันพร้อมทั้งอธิบาย
13. จงยกตัวอย่างความผิดพลาดจาก Software, hardware และ human ที่ส่งผลต่อความปลอดภัย มา 3 ข้อ
14. จงยกตัวอย่างความต้องการที่ไม่ใช่หน้าที่ มา 3 ข้อ
15. จงยกตัวอย่างอันตรายที่นำไปพิจารณาประกอบเมื่อถึงขั้นตอนการออกแบบระบบ มา 3 ข้อ

**ตอนที่ 3 คำถามที่ใช้ประเมินส่วนเนื้อหาในแบบรูป**

<u>ให้ตอบคำถามในตอนที่ 3 เมื่อมีการใช้แบบรูปประกอบการทดลองเท่านั้น</u>

ข้อ 16 – 20 ให้คะแนน 1 – 5 แทนความหมาย น้อยมาก น้อย ปานกลาง ดี และดีมาก ตามลำดับ

16. ท่านคิดว่าแบบรูปที่ท่านใช้มีความครบถ้วนสมบูรณ์ระดับใด
17. ท่านคิดว่าแบบรูปที่ท่านใช้มีความถูกต้องในระดับใด
18. ท่านคิดว่าแบบรูปที่ท่านใช้มีความง่ายในการใช้และทำความเข้าใจระดับใด
19. ท่านคิดว่าแบบรูปที่ท่านใช้มีความชัดเจนในระดับใด
20. ท่านคิดว่าแบบรูปที่ท่านใช้ช่วยแก้ปัญหาการทำความต้องการนี้ในระดับใด

**ค.4 ชุดคำถามที่ 3 Train Location and Speed**

**ตอนที่ 1 จงเลือกคำตอบที่ถูกต้องที่สุดเพียง 1 ข้อ**

1. ข้อใดเป็นเป้าหมายของ Train Location and Speed Determination

ก. กำหนดตำแหน่ง ความเร็ว อัตราเร่ง และทิศทางได้

ข. ตรวจสอบและติดตามตำแหน่ง ความเร็ว อัตราเร่ง และทิศทาง ได้อย่างถูกต้อง

ค. ควบคุมให้รถไฟใช้ความเร็วตามโปรแกรมที่กำหนดไว้

ง. ข้อ ก. และ ข.

จ. ถูกทุกข้อ

2. ข้อใดคือปัญหาในงานวิศวกรรมความต้องการสำหรับ Train Location and Speed Determination

ก. การกำหนดค่าความปลอดภัยของตำแหน่งและความเร็ว

ข. การอธิบายวิธีการหาหรือคำนวณตำแหน่งของรถไฟ

ค. การพยายามคำนวณความเร็วของรถไฟให้มีความแม่นยำ

ง. การกำหนดการโปรแกรมการใช้ความเร็วของรถไฟ

จ. การหาวิธีการในการกำหนดตำแหน่งของรถไฟ

3. ข้อใดไม่ใช่ข้อจำกัด (System Constraint) ในส่วนของ Train Location and Speed Determination

ก. ความคลาดเคลื่อนในการประเมินตำแหน่งรถไฟ

ข. ค่าความแม่นยำในการวัดตำแหน่งรถไฟ

ค. ความคลาดเคลื่อนในการประเมินความเร็วรถไฟกรณีปกติ

ง. ค่าความแม่นยำในการวัดความเร็วรถไฟ

จ. ค่าความแม่นยำในการวัดอัตราเร่งของรถไฟ

4. ข้อใดคือกลุ่มคนที่เกี่ยวข้องกับความต้องการของ Train Location and Speed Determination

ก. พนักงานควบคุม และผู้โดยสาร

ข. พนักงานควบคุม และผู้จัดการระบบ

ค. พนักงานควบคุม ผู้จัดการระบบ และผู้โดยสาร

ง. พนักงานกำหนดเส้นทาง ผู้จัดการระบบ และผู้โดยสาร

จ. พนักงานควบคุม และผู้กำหนดเส้นทางเดินรถ

5. ข้อใดไม่จัดว่าเป็นความต้องการใน Train Location and Speed Determination

ก. ต้องควบคุมตำแหน่ง ความเร็ว และทิศทางได้

ข. ต้องเริ่มทำงานโดยอัตโนมัติเมื่อรถไฟเข้าสู่ระบบ

ค. ต้องพิจารณาความคลาดเคลื่อนประกอบการคำนวณด้วย

ง. ต้องกำหนดตำแหน่งทั้งหน้าและหลังรถไฟได้อย่างถูกต้อง

จ. ไม่มีข้อใดที่ไม่ใช่ความต้องการสำหรับ Train Location and Speed Determination

6. ข้อใดไม่มีความเกี่ยวข้องกับการพิจารณาความต้องการของ Train Location and Speed Determination

ก. ตำแหน่งรางรถไฟ

ข. สถานีโดยสารและชานชาลา

ค. ความยาวของรถไฟ

ง. ความสามารถในการทำความเร็วของรถไฟ

จ. การเคลื่อนที่ของรถไฟ

7. ข้อใดคือหน้าที่ที่เกี่ยวข้องกับความปลอดภัยของ Train Location and Speed Determination

ก. ทำการป้องกันการคำนวณความเร็วรถไฟผิดพลาด

ข. การป้องกันการไถลของรถไฟ

ค. การเตือนเมื่อมีการใช้ความเร็วเกินกำหนด

ง. การป้องกันการใช้รางร่วมกับรถไฟอื่นทั้งที่ไม่ได้รับอนุญาต

จ. การสนองตอบเมื่อตำแหน่งของรถไฟยังไม่มีการกำหนด

8. ข้อใดเป็นอันตรายที่อาจเกิดขึ้นจากความผิดพลาดของ Train Location and Speed Determination โดยตรง

ก. รถไฟวิ่งผิดทางโดยวิ่งสวนกับอีกทางหนึ่งจนชนกับอีกขบวนที่สวนกัน

ข. รถไฟอยู่ในตำแหน่งที่ไม่คาดคิดจากความผิดพลาดในการระบุตำแหน่งและทำให้ชนกันขึ้น

ค. รถไฟใช้ความเร็วเกินกว่าที่กำหนดจนเป็นผลให้รถไฟตกราง

ง. ข้อ ก. และ ข.

จ. ข้อ ข. และ ค.

9. ข้อใดคือผลที่ไม่สามารถยอมรับรับได้และมีความสำคัญมากสำหรับ Train Location and Speed Determination

ก. การกำหนดรวมทั้งค่าประเมินความคลาดเคลื่อนสำหรับรถไฟมีความผิดพลาด

ข. การวัดตำแหน่งและความเร็วของรถไฟคลาดเคลื่อนเกินที่กำหนด

ค. การควบคุมรถไฟให้สามารถจอดตามจุดที่กำหนดได้อย่างแม่นยำ

ง. ข้อ ก. และ ข.

จ. ข้อ ข. และ ค.

10. ข้อใดต่อไปนี้เป็นเทคนิคที่มีความเหมาะสมเป็นพิเศษในการนำมาใช้ในงานวิศวกรรมความต้องการสำหรับ Train Location and Speed Determination

ก. การบันทึกที่มาของความต้องการ

ข. การวิเคราะห์ลำดับความสำคัญของความต้องการ

ค. การใช้ทีมที่มีความหลากหลายในการตรวจสอบความต้องการ

ง. การแยกแยะและให้คำแนะนำผู้ใช้งานระบบ

จ. ไม่มีเทคนิคใดที่เหมาะสมเป็นพิเศษ

**ตอนที่ 2 จงตอบคำถามต่อไปนี้**

11. จงยกตัวอย่างปัญหาในงานวิศวกรรมความต้องการที่พบในระบบที่กล่าวถึง พร้อมทั้งอธิบายวิธีการในการจัดการกับปัญหานั้น
12. จงยกตัวอย่างความต้องการที่มีความสัมพันธ์กันพร้อมทั้งอธิบาย
13. จงยกตัวอย่างความผิดพลาดจาก Software, hardware และ human ที่ส่งผลต่อความปลอดภัย มา 3 ข้อ
14. จงยกตัวอย่างความต้องการที่ไม่ใช่หน้าที่ มา 3 ข้อ
15. จงยกตัวอย่างอันตรายที่นำไปพิจารณาประกอบเมื่อถึงขั้นตอนการออกแบบระบบ มา 3 ข้อ

**ตอนที่ 3 คำถามที่ใช้ประเมินส่วนเนื้อหาในแบบรูป**

ให้ตอบคำถามในตอนที่ 3 เมื่อมีการใช้แบบรูปประกอบการทดลองเท่านั้น

ข้อ 16 ถึง 20 ให้คะแนน 1 ถึง 5 แทนความหมาย น้อยมาก น้อย ปานกลาง ดี และดีมาก ตามลำดับ

16. ท่านคิดว่าแบบรูปที่ท่านใช้มีความครบถ้วนสมบูรณ์ระดับใด
17. ท่านคิดว่าแบบรูปที่ท่านใช้มีความถูกต้องในระดับใด
18. ท่านคิดว่าแบบรูปที่ท่านใช้มีความง่ายในการใช้และทำความเข้าใจระดับใด
19. ท่านคิดว่าแบบรูปที่ท่านใช้มีความชัดเจนในระดับใด
20. ท่านคิดว่าแบบรูปที่ท่านใช้ช่วยแก้ปัญหาการทำความต้องการนี้ในระดับใด

## ค.5 ชุดคำถาม ประเมินและแสดงความเห็นเกี่ยวกับแบบรูป

*ให้คะแนน 1 - 5 แทน น้อยที่สุด – มากที่สุด*

1. ท่านคิดว่าการนำแบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์จะช่วยให้การทำงานง่ายขึ้นมากกว่าการที่ไม่ใช้แบบรูปมากน้อยเพียงใด
2. ท่านคิดว่าการใช้แบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์จะช่วยให้การทำงานมีประสิทธิภาพมากกว่าการที่ไม่ใช้แบบรูปมากน้อยเพียงใด
3. ท่านคิดว่าการใช้แบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์จะช่วยให้การลดเวลาในการทำงานมากกว่าการที่ไม่ใช้แบบรูปมากน้อยเพียงใด
4. ท่านคิดว่าการนำแบบรูปมาใช้กับความต้องการความปลอดภัยสัมพันธ์มีความเหมาะสมมากน้อยเพียงใด
5. ท่านคิดว่าโครงสร้างของแบบรูปมีความเหมาะสมมากน้อยเพียงใด
6. ท่านคิดว่าองค์ประกอบและเนื้อความในแต่ละส่วนขององค์ประกอบที่ใช้ในแบบรูปมีความเหมาะสมมากน้อยเพียงใด

7. ท่านคิดว่าการนำแบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์โดยเฉพาะเมื่อเปรียบเทียบกับการนำกลับมาใช้ใหม่วิธีอื่นๆที่ท่านรู้จักมีความเหมาะสมมากน้อยเพียงใด อย่างไร
8. ท่านรู้สึกดีหรือพึงพอใจในการใช้งานแบบรูปมากน้อยเพียงใด
9. ท่านรู้สึกว่าแบบรูปที่เสนอนี้ก่อให้เกิดการนำไปใช้อย่างต่อเนื่องมากน้อยเพียงใด
10. ท่านรู้สึกถึงความคุ้มค่าในการใช้แบบรูปที่เสนอนี้มากน้อยเพียงใด

# ประวัติผู้เขียนวิทยานิพนธ์

นายเอกชัย แซ่ตั้ง เกิดเมื่อวันที่ 6 กันยายน 2524 ที่จังหวัดอุบลราชธานี สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2545 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2546