บทที่ 5
ผลการวิจัยและการอภิปรายผล

การศึกษาในครั้งนี้เพื่อออกแบบและพัฒนาโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม และศึกษาความเข้าใจโปรแกรมของผู้เรียนที่เรียนด้วยโมเดลฯ ความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯและผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลฯ ปรากฏผลการวิจัยดังที่จะนำเสนอผลการวิจัย ดังรายละเอียดต่อไปนี้

1. ผลการออกแบบและพัฒนาโมเดลสิ่งแวดล้อมทางการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม
2. ผลการศึกษาความเข้าใจโปรแกรมของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม
3. ผลการศึกษาความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม
4. ผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม
5. สรุปและอภิปรายผลการวิจัย

1. ผลการออกแบบและพัฒนาโมเดลสิ่งแวดล้อมทางการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม

การออกแบบและพัฒนาโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม ผู้วิจัยได้ดำเนินการการวิจัยเชิงพัฒนา (Developmental research) แบบ Type II (Richey and Klein, 2007) ซึ่งประกอบด้วย 3 ระยะ (Phase) คือ ระยะที่ 1 การพัฒนาโมเดล (Model development phase) ระยะที่ 2 การตรวจสอบความตรงของโมเดล (Model validation phase) และระยะที่ 3 การใช้โมเดล (Model use phase) ผลที่ได้ทั้ง 3 ระยะ สามารถสรุปผลได้ตามลำดับดังนี้

1.1 ระยะที่ 1 การพัฒนาโมเดล (Model development phase)

การพัฒนาโมเดล เป็นระยะที่ผู้วิจัยออกแบบและพัฒนาโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรมและศึกษากระบวนการพัฒนาโมเดลฯ ผู้วิจัยได้ดำเนินการตามลำดับดังนี้ 1) การศึกษาหลักการ ทฤษฎี และงานวิจัยที่เกี่ยวข้อง 2) การศึกษาบริบทการเรียนการสอนวิชา CT212 โครงสร้างโปรแกรม 3) การสังเคราะห์กรอบแนวคิดเชิงทฤษฎี และกรอบแนวคิดในการออกแบบโมเดลฯ 4) การออกแบบและพัฒนาโมเดลฯสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม 5) การศึกษาบริบทการออกแบบและพัฒนาโมเดลฯ 6)การ

ประเมินโมเดลฯ 7) การศึกษาความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ ผลที่ได้จากทั้ง 7 ขั้นตอนทำให้ได้โมเดลต้นแบบสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม ซึ่งมีองค์ประกอบที่สำคัญดังนี้

1.1.1 การกระตุ้นโครงสร้างทางปัญญา

พื้นฐานการออกแบบสถานการณ์ปัญหานั้นใช้วิธีการการกระตุ้นโครงสร้างทางปัญญาโดยนำพื้นฐานแนวคิดของ Cognitive constructivism ของ Piaget โดยอาศัยหลักการของ Open learning environment(Hannafin, 1999) ในการออกแบบผู้วิจัยได้กระตุ้นใช้หลักการความเข้าใจโปรแกรมของBrook(1983) จากการนำเสนอโปรแกรมภาษา C++ ในเรื่องการเขียนโปรแกรมเรียกตัวเอง ที่เน้นผู้เรียนเป็นศูนย์กลางโดยจัดให้ผู้เรียนเข้าไปเผชิญกับปัญหาในบริบทที่กำหนด ซึ่งจากผลการวิจัย พบว่า สถานการณ์ปัญหาที่เตรียมให้นั้นสามารถส่งเสริมความเข้าใจโปรแกรมให้แก่ผู้เรียน ผู้เรียนต้องใช้ความสามารถทางสติปัญญาในการทำความเข้าใจโปรแกรมด้วยตนเองเพื่อให้เกิดการเรียนรู้ ซึ่งหาได้จากองค์ประกอบอื่น ๆที่ได้จัดเตรียมไว้ให้ ผู้เรียนสามารถเข้าใจวัตถุประสงค์การทำงานของโปรแกรม สามารถแก้ไขความผิดพลาดของโปรแกรมได้ สามารถเปลี่ยนแปลงโครงสร้างของโปรแกรมให้เขียนแบบเรียกตัวเองได้ ดังนั้นสถานการณ์ปัญหาสามารถกระตุ้นโครงสร้างทางปัญญาแก่ผู้เรียนได้อย่างมีประสิทธิภาพ

1.1.2 การสนับสนุนการปรับสมดุลทางปัญญา

จากหลักการ Cognitive constructivism ของ Piaget ในการออกแบบผู้วิจัยได้ให้ผู้เรียนเริ่มศึกษาจากสถานการณ์ปัญหาเป็นหลัก หลังจากที่ผู้เรียนศึกษาสถานการณ์ปัญหา สถานการณ์ปัญหาจะกระตุ้นให้ผู้เรียนเสียสมดุลทางปัญญา และการกำหนดภารกิจที่ส่งเสริมความเข้าใจโปรแกรม ผู้เรียนจะเกิดแรงจูงใจภายในที่จะปรับเข้าสู่ภาวะสมดุลทางปัญญาโดยการค้นคว้าหาความรู้จากแหล่งการเรียนรู้ และกรณีใกล้เคียง ซึ่งผู้วิจัยได้จัดเตรียมแหล่งสารสนเทศที่จำเป็นต่อการเรียนรู้ เพื่อการค้นพบคำตอบในการแก้ปัญหา การสร้างความรู้และการส่งเสริมความเข้าใจโปรแกรม ดังรายละเอียดต่อไปนี้

1.1.2.1 แหล่งการเรียนรู้(Resource) ผู้วิจัยได้ออกแบบให้ผู้เรียนได้เรียนรู้โดยการค้นพบ (Discovery learning) จากการเสาะแสวงหาความรู้จากแหล่งต่าง ๆ โดยการจัดแหล่งเรียนรู้ไว้ภายในสิ่งแวดล้อมการเรียนรู้ตามหลักการของ Open Learning Environments แบ่งเป็น แหล่งการเรียนรู้คงที่(Static Resource) และแหล่งการเรียนรู้พลวัตร(Dynamic Resource) ผู้เรียนสามารถสร้างความรู้ใหม่ขึ้นมาได้จากการเสาะแสวงหาจากแหล่งการเรียนรู้ที่ได้จัดเตรียมไว้ให้ภายในโมเดลฯ ประกอบด้วย 1) ธนาคารความรู้เป็นที่รวบรวมความรู้ของรูปแบบไวยากรณ์ภาษา C++ การเขียนโปรแกรมเรียกตัวเอง และตัวอย่างฟังก์ชันเรียกตัวเอง 2) เคล็ดไม่ลับ เป็นคำอธิบายจากผู้เชี่ยวชาญถึงกระบวนการทำความเข้าใจโปรแกรมเรียกตัวเอง 3) มือใหม่เชิญที่นี่ เป็นคำอธิบายและแสดงถึงกระบวนการทดสอบและทวนสอบความถูกต้องของโปรแกรมเรียกตัวเอง 4) ลิงค์ที่เกี่ยวข้องซึ่งเป็นที่รวบรวมลิงค์ต่าง ๆ เป็นแหล่งการเรียนที่เปลี่ยนแปลงอยู่ตลอดเวลา 5) เครื่องมือสืบค้น(Search engine)สำหรับใช้ในการค้นหา

แหล่งข้อมูลเพิ่มเติมที่ต้องการและสามารถนำมาใช้ในการสร้างความรู้นอกเหนือจากที่จัดไว้ภายในสิ่งแวดล้อมทางการเรียนรู้บนเครือข่าย ผลการวิจัย พบว่า ผู้เรียนใช้แหล่งการเรียนรู้เพื่อสนับสนุนการปรับโครงสร้างทางปัญญาทำให้สามารถแก้ไขภารกิจตามที่ได้รับมอบหมายได้ดังข้อมูลการสัมภาษณ์ที่ว่า

ธนาคารความรู้เป็นองค์ประกอบที่ใช้สำหรับศึกษา ดูรูปแบบไวยากรณ์ และตัวอย่างเพื่อนำมาใช้ในการแก้ไขภารกิจ ดูจากข้อมูลในองค์ประกอบธนาคารความรู้แล้วเห็นว่ามีข้อมูลหลากหลายตั้งแต่ความรู้พื้นฐานจนถึงความรู้เกี่ยวกับการเขียนโปรแกรมเรียกตัวเองซึ่งเพียงพอสำหรับนำมาใช้ในการแก้ไขปัญหาได้

1.1.2.2 กรณีใกล้เคียง (Related-case) เป็นการนำเสนอแหล่งสนับสนุนทางการเรียนรู้ที่เป็นประสบการณ์ให้แก่ผู้เรียน ในกรณีที่ผู้เรียนไม่มีประสบการณ์เรื่องที่ใกล้เคียงกับปัญหาที่เผชิญอยู่เพื่อเป็นแนวทางให้แก่ผู้เรียนในการเชื่อมโยงประสบการณ์ที่ได้รับไปสู่การเรียนรู้ ซึ่งการออกแบบกรณีใกล้เคียง(Related case)ในสิ่งแวดล้อมทางการเรียนรู้ใช้หลักการ Constructivist learning environments(Jonassen, 1999) โดยกำหนดกรณีศึกษาให้ผู้เรียนเข้าไปศึกษาเพื่อเรียนรู้ถึงกระบวนการคิดแก้ปัญหาทำให้ผู้เรียนเข้าใจประเด็นของปัญหาได้ชัดเจน ซึ่งช่วยสนับสนุนผู้เรียนใน 2 ทางคือ 1) ช่วยให้ผู้เรียนจดจำได้ดี 2) เป็นแหล่งอ้างอิงที่สำคัญที่ให้ผู้เรียนสามารถเปรียบเทียบกับประสบการณ์ที่ตนเองเคยผ่านมาหรือที่เคยแก้ไขมาแล้ว จากผลการวิจัย พบว่าเมื่อผู้เรียนศึกษาสถานการณ์ปัญหาและไม่สามารถแก้ไขภารกิจการเรียนรู้ได้ กรณีใกล้เคียงเป็นทางเลือกหนึ่งที่ผู้เรียนสามารถศึกษาและสร้างความรู้ได้ด้วยตนเอง ผู้เรียนสามารถนำความรู้ที่ได้จากการศึกษามาใช้อ้างอิงโดยเชื่อมโยงนำประสบการณ์ที่ใกล้เคียงมาใช้ในการแก้ปัญหาได้ ดังข้อมูลการสัมภาษณ์ที่ว่า "ใช้องค์ประกอบ กรณีใกล้เคียงมากที่สุดเพราะสามารถศึกษาโครงสร้างรูปแบบของโปรแกรม วิธีการคำนวณในการแก้ปัญหาและวิธีการเขียนโปรแกรมแบบเรียกตัวเองเพื่อนำมาเปรียบเทียบกับสถานการณ์ปัญหาทำให้สามารถแก้ไขปัญหาตามภารกิจได้"

1.1.3 การส่งเสริมการขยายโครงสร้างทางปัญญา

การส่งเสริมการขยายโครงสร้างทางปัญญามุ่งเน้นการส่งเสริมให้ผู้เรียนได้สร้างความเข้าใจโปรแกรมด้วยตนเอง ด้วยการลงมือกระทำผ่านกระบวนการรู้คิด และอาศัยจากประสบการณ์หรือความรู้เดิมที่มีอยู่แล้วเชื่อมโยงกับประสบการณ์ใหม่เพื่อขยายโครงสร้างทางปัญญาของผู้เรียน(สุมาลี ชัยเจริญ, 2551) ในการออกแบบจะอาศัยหลักการ Social constructivist ที่เน้นการมีปฏิสัมพันธ์ทางสังคม ดังหลักการของ Collaboration ที่เน้นการร่วมมือกันแก้ปัญหา รวมถึงหลักการออกแบบเครื่องมือทางปัญญา ที่เป็นตัวกลางซึ่งจะสนับสนุนเพิ่มพูนหรือขยายการคิดของ Hannafin(1999) ซึ่งมีองค์ประกอบที่สำคัญ 4 องค์ประกอบ คือ

1.1.3.1 แลกเปลี่ยนเรียนรู้ความเข้าใจโปรแกรม ผู้วิจัยได้ออกแบบสิ่งแวดล้อมที่ส่งเสริมการขยายโครงสร้างทางปัญญาการสร้างความเข้าใจโปรแกรมของผู้เรียนโดยนำพื้นฐานแนวคิดมาจาก Social constructivism ใช้หลักการของการร่วมมือกันเรียนรู้

(Collaborative learning) เพื่อเน้นการร่วมมือกันแก้ปัญหา สนับสนุนให้ผู้เรียนได้แลกเปลี่ยนประสบการณ์กับผู้อื่นเพื่อขยายมุมมองของความคิดทำให้เกิดความเข้าใจโปรแกรม นอกจากนี้ใช้หลักการของ OLEs model โดยจัดเตรียมเครื่องมือสื่อสาร(Communication tool) เพื่อสื่อสารระหว่างผู้เรียนกับผู้เรียน และผู้เรียนกับผู้เชี่ยวชาญ ประกอบด้วย 1)ไปรษณีย์อิเล็กทรอนิกส์ (Electronics mail: E-mail) ใช้สำหรับให้ผู้เรียนติดต่อปรึกษากับผู้เชี่ยวชาญ 2)กระดานสนทนา (Web board) ใช้กำหนดประเด็นหรือกระทู้ตามที่ผู้เรียนกำหนด เพื่อช่วยกันอภิปรายตอบประเด็น หรือกระทู้ ระหว่างผู้เรียนกับผู้เรียน และผู้เรียนกับผู้เชี่ยวชาญ ผลการวิจัย พบว่า ผู้เรียนสามารถตอบคำถามสถานการณ์ทั้ง 6 สถานการณ์ผ่านกระดานสนทนา ผู้เรียนสามารถติดต่อกับผู้เชี่ยวชาญหรือผู้เรียน โดยใช้ไปรษณีย์อิเลคโทรนิกส์(E-mail) ดังข้อมูลการสัมภาษณ์ที่ว่า “องค์ประกอบที่ใช้มากที่สุดคือการแลกเปลี่ยนเรียนรู้ ชอบ ปรึกษาเพื่อน ๆในกลุ่ม หรือก็ขอคำแนะนำจากอาจารย์ผู้ช่วยสอนหรือถามเพื่อน ๆกลุ่มอื่น ชอบการแลกเปลี่ยนเรียนรู้ เพราะจะได้รับคำตอบตรงประเด็นทำให้เสียเวลาน้อยลง”

1.1.3.2 เครื่องมือส่งเสริมความเข้าใจโปรแกรม (Tools enhanced Program Comprehension) ผู้วิจัยได้ออกแบบสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมการขยายโครงสร้างทางปัญญาการสร้างความเข้าใจโปรแกรม โดยใช้หลักการ OLEs ซึ่งเป็นรูปแบบหนึ่งของการจัดสิ่งแวดล้อมทางการเรียนรู้ตามแนวทางคอนสตรัคติวิสต์ ออกแบบและพัฒนาโดย Hannafin (1999) และทฤษฎีความเข้าใจโปรแกรม มาใช้ส่งเสริมการขยายโครงสร้างทางปัญญาการสร้างความเข้าใจโปรแกรม ผู้เรียนสามารถสืบเสาะแสวงหาความรู้(Inquiry) และดาวน์โหลดโปรแกรมที่ต้องการได้ด้วยตนเอง โดยผู้วิจัยได้จัดเตรียมลิงค์เพื่อเชื่อมโยงไปยังเว็บไซต์ของเครื่องมือ(Tool)ต่าง ๆผู้เรียนสามารถนำมาใช้ในการทำความเข้าใจโปรแกรมแบ่งเป็น 1) เครื่องมือกระบวนการ (Processing tool) ซึ่งสนับสนุนกระบวนการรู้คิดของผู้เรียน 2) เครื่องมือจัดกระทำ (Manipulation tool) เพื่อตรวจสอบความตรงหรือหรือความเชื่อหรือสำรวจ 3) เครื่องมือความเข้าใจโปรแกรม(Program comprehension tool) 4) เครื่องมือในการค้นหา (Seeking tool) ดังแสดงในตารางที่ 5.1 ผลการวิจัย พบว่า ผู้เรียนใช้เครื่องมือส่งเสริมความเข้าใจโปรแกรมเพื่อส่งเสริมการขยายโครงสร้างทางปัญญาการสร้างความเข้าใจโปรแกรม ผู้เรียนบางกลุ่มใช้ Google เพื่อสามารถสืบเสาะแสวงหาความรู้(Inquiry) และผู้เรียนทุกกลุ่มใช้เครื่องมือจัดกระทำโดยใช้โปรแกรมตัวแปลภาษา C++ เพื่อตรวจสอบกระบวนการทำความเข้าใจโปรแกรม ดังข้อมูลการสัมภาษณ์ที่ว่า

การใช้งานโมเดลฯช่วยให้ผมเข้าใจโปรแกรมเรียกตัวเองได้มากขึ้น ช่วยให้มีช่องทางในการแก้ปัญหามากขึ้น ผมใช้กระดานสนทนา(Webboard) เพื่อศึกษาแนวทางการแก้ปัญหากับเพื่อน ๆ กลุ่มอื่น ใช้ Google ในการสืบค้นข้อมูล และปรึกษาผู้เชี่ยวชาญเพราะไม่เข้าใจประเด็นปัญหา และใช้ตัวแปลภาษาเป็นเครื่องมือในการทดสอบความถูกต้องของโปรแกรม เพราะกรณีที่ผลลัพธ์ไม่ ถูกต้องแสดงว่าผมคิดผิดผมต้องคิดใหม่จนกระทั่งผลลัพธ์ถูกต้องในทุกกรณี

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

ตารางที่ 5.1 แสดงเครื่องมือส่งเสริมความเข้าใจโปรแกรม

เครื่องมือ	หน้าที่	การออกแบบ
กระบวนการ	สนับสนุนกระบวนการรู้คิด	เตรียมลิงค์เพื่อให้ดาวน์โหลด 1) Crimson editor 2) EDITPAD 3) NODEPAD++ 4) PAGEBreeze
จัดกระทำ	ตรวจสอบความตรงหรือความเชื่อ	เตรียมลิงค์ตัวแปลภาษา 1) Devc++ 2) C++
ความเข้าใจโปรแกรม	สนับสนุนความเข้าใจโปรแกรม	เตรียมลิงค์เพื่อให้ดาวน์โหลด 1) PCTs 2) Source Code 3) CRYSTAL FLOW 4) ASM Scope
การค้นหา	ค้นหาคำสำคัญ	Search engines ใช้ Google

1.1.3.3 ห้องปฏิบัติการส่งเสริมความเข้าใจโปรแกรม การออกแบบห้องปฏิบัติการส่งเสริมความเข้าใจโปรแกรมนี้ ผู้วิจัยได้ทำการสังเคราะห์กลไกกระบวนการทำความเข้าใจของ Bottom-up Comprehension Model(Pennington, 1987) และ Top-down Comprehension Model(Brook, 1983) เพื่อนำมาออกแบบห้องปฏิบัติการเพื่อส่งเสริมการขยายโครงสร้างทางปัญญาการสร้างความเข้าใจโปรแกรมของผู้เรียน โดยมุ่งเน้นให้ผู้เรียนสามารถพัฒนากระบวนการทำความเข้าใจโปรแกรมเรียกตัวเองในบริบทต่าง ๆได้ด้วยตนเอง ผู้วิจัยจัดเตรียมกรณีศึกษาแบ่งเป็น 3 ระดับจากระดับต่ำไปจนถึงระดับสูงตามความยากง่ายของกระบวนการทำความเข้าใจเพื่อพัฒนากระบวนการคิดระดับสูงให้แก่ผู้เรียน ผลงานวิจัย พบว่า ผู้เรียนสามารถทำความเข้าใจโปรแกรม โดยการคิดแก้ปัญหาและติดตามการทำงานของโปรแกรมเป็นลำดับขั้นตอนตามคำแนะนำจากการทำความเข้าใจหนึ่งคำสั่งเป็นทำความเข้าใจเป็นกลุ่มของคำสั่ง ผู้เรียนสามารถก่อรูปความเข้าใจคำสั่งโปรแกรมมากขึ้น ๆ จนกระทั่งเข้าใจคำสั่งโปรแกรมได้ทั้งหมด และสามารถทำความเข้าใจโปรแกรมในภาพรวมสรุปเป็นวัตถุประสงค์การทำงานของโปรแกรมได้อย่างถูกต้อง สรุปตีความได้จากผลของการแก้ไขภารกิจการเรียนรู้ที่ผู้เรียนสามารถวิเคราะห์ปัญหา อธิบายและตอบคำถามได้อย่างถูกต้อง

1.1.3.4 ฐานการช่วยเหลือ(Scaffolding) ผู้วิจัยได้ออกแบบฐานความช่วยเหลือสำหรับผู้เรียนที่อยู่ต่ำกว่า Zone of Proximal Development โดยใช้หลักการโมเดลสิ่งแวดล้อมทางการเรียนรู้ตามแนวทางคอนสตรัคติวิสต์ Open Leaning Environments ซึ่งออกแบบและพัฒนาโดย Michael Hannafin(1999) เพื่อให้ผู้เรียนสืบเสาะแสวงหาความรู้ได้ด้วยตนเองและควบคุมการเรียนรู้โดยใช้เมตาคอกนิชั่น(Metacognition) ประกอบด้วย ฐานการช่วยเหลือการสร้างความคิดรวบยอด(Conceptual scaffolding) ฐานการช่วยเหลือด้านความคิด (Metacognitive scaffolding) ฐานการช่วยเหลือด้านกระบวนการ(Procedural scaffolding) และฐานความช่วยเหลือด้านกลยุทธ์(Strategic scaffolding) ผลการวิจัย พบว่า ผู้เรียนใช้ฐานความช่วยเหลือเพื่อส่งเสริมการขยายโครงสร้างทางปัญญาการสร้างความเข้าใจโปรแกรม ผู้เรียนสามารถสืบเสาะแสวงหาความรู้(Inquiry) ได้ด้วยตนเองเพื่อสร้างความรู้ ดังข้อมูลการสัมภาษณ์ที่ว่า

ถ้าผมไม่มีความรู้เพียงพอในการแก้ปัญหา ส่วนมากแล้วจะคิดไม่ค่อยได้ ผมต้องหากระบวนการคิดที่ถูกต้อง องค์ประกอบที่เลือกศึกษาคือ ฐานความช่วยเหลือเพราะมีการอธิบายวิธีการคิดอย่างเช่นสถานการณ์ที่ 2 มีการอธิบายวิธีการคิดดอกเบี้ยทบต้นทำให้ทราบวิธีการคิดที่ถูกต้อง ผมสามารถนำความรู้นี้ไปพิจารณาคำสั่งโปรแกรมปัญหาที่ให้มาได้ว่าส่วนใดของโปรแกรมที่ผิดช่วยทำให้ผมสามารถแก้ไขคำสั่งที่ผิดให้ถูกต้องได้

1.2 ระยะที่ 2 การตรวจสอบความตรงของโมเดล (Model validation phase)

การตรวจสอบความตรงของโมเดล เป็นระยะที่ผู้วิจัยตรวจสอบความตรงภายในโมเดลและตรวจสอบความตรงภายนอกโมเดล ผู้วิจัยได้ดำเนินการตามลำดับดังนี้ 1) การประเมินโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมโดยผู้เชี่ยวชาญ 2) การประเมินความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ 3) การประเมินความเข้าใจโปรแกรมของผู้เรียนที่เรียนด้วยโมเดลฯ 4) การประเมินบริบทการเรียนด้วยโมเดลฯของผู้เรียน 5) การประเมินบริบทการสอนด้วยโมเดลฯของผู้สอน 6) การประเมินผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม ผลที่ได้ทำให้ได้โมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมที่มีคุณภาพซึ่งผ่านการประเมินความตรงทางทฤษฎีจากผู้เชี่ยวชาญ ผู้เรียนสามารถใช้โมเดลฯเพื่อพัฒนาการเรียนรู้เรื่องการเขียนโปรแกรมเรียกตัวเองได้อย่างมีประสิทธิผลและทราบถึงผลกระทบของการใช้โมเดลฯที่ส่งผลต่อการเปลี่ยนพฤติกรรมของผู้เรียน ผู้สอนและองค์กร สรุปผลได้ดังนี้

1.2.1 ผลการตรวจสอบความตรงภายในโมเดล

การตรวจสอบความตรงภายในโมเดล โดยผลการศึกษาดังกล่าวได้จากแบบประเมินโมเดลฯสำหรับผู้เชี่ยวชาญในด้านเนื้อหา ด้านสื่อ และด้านการออกแบบสิ่งแวดล้อมการเรียนรู้ แบบบันทึกการใช้โมเดลของผู้เรียนและการวิเคราะห์โปรโตคอลจากแบบสัมภาษณ์การใช้โมเดลฯของผู้เรียน ผลที่ได้ทำให้ได้โมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมที่มีคุณภาพซึ่งผ่านการประเมินความตรงทางทฤษฎีจากผู้เชี่ยวชาญ ผู้เรียน

สามารถใช้โมเดลฯเพื่อพัฒนาการเรียนรู้เรื่องการเขียนโปรแกรมเรียกตัวเองได้อย่างมีประสิทธิผล (Cost-effective) สรุปผลได้ดังนี้

1.2.1.1 ผลการประเมินโมเดลฯ เป็นผลจากการประเมินคุณภาพของสิ่งแวดล้อมการเรียนรู้ โดยการตรวจสอบคุณภาพด้านต่าง ๆโดยผู้เชี่ยวชาญ ได้แก่ด้านเนื้อหา ด้านสื่อ ด้านการออกแบบสิ่งแวดล้อมฯ สรุปผลได้ดังนี้

(1) ด้านเนื้อหา พบว่า เนื้อหามีความถูกต้อง น่าสนใจ ทันสมัย ครอบคลุม ชัดเจน กะทัดรัด เป็นลำดับขั้นและง่ายต่อการทำความเข้าใจ ภาพประกอบมีความเหมาะสมสอดคล้องกับหลักการทฤษฎี

(2) ด้านสื่อ พบว่า การออกแบบเครื่องนำทาง(Navigator) สัญลักษณ์ที่เป็นไอคอน(Icon) การเชื่อมโยง(Link) องค์ประกอบทางศิลปะ(Architecture) ขนาดของตัวอักษร สี ภาพประกอบ ภาพเคลื่อนไหว เสียงคำอธิบายมีความเหมาะสม สอดคล้องกับหลักการทฤษฎี

(3) ด้านการออกแบบสิ่งแวดล้อมการเรียนรู้ พบว่า โมเดลฯเปิดโอกาสให้ผู้เรียนได้แสวงหาความรู้ได้ด้วยตนเอง ได้ทุกเวลา ทุกสถานที่ ตามที่ผู้เรียนต้องการ ผู้เรียนสามารถควบคุมการเรียนรู้ได้ด้วยตนเอง การออกแบบสิ่งแวดล้อมต่าง ๆสอดคล้องกับหลักการทฤษฎี ได้แก่ สถานการณ์ปัญหา กรณีใกล้เคียง (Related case) ฐานความช่วยเหลือ (Scaffolding) แหล่งการเรียนรู้ (Resource) ห้องปฏิบัติการส่งเสริมความเข้าใจโปรแกรม เครื่องมือส่งเสริมความเข้าใจโปรแกรม แลกเปลี่ยนเรียนรู้

ผลการวิจัย พบว่า โมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมมีความตรงภายในโมเดล

1.2.1.2 ผลการประเมินการใช้โมเดลฯของผู้เรียน เป็นผลจากการศึกษาการใช้โมเดลของผู้เรียนในการแก้ปัญหาและความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ ด้านการใช้โมเดลฯ ด้านประโยชน์ที่ได้รับ สรุปผลได้ดังนี้

(1) ด้านการใช้โมเดลฯ พบว่า โมเดลฯมีการออกแบบองค์ประกอบต่าง ๆได้อย่างเหมาะสมและชัดเจน โมเดลฯมีความคงที่และเป็นมาตรฐานที่ดี ผู้เรียนสามารถใช้โมเดลฯเพื่อแก้ไขสถานการณ์ต่าง ๆได้ง่ายและได้ผลลัพธ์ถูกต้องสมบูรณ์ โดยเลือกศึกษาองค์ประกอบต่าง ๆได้แก่ สถานการณ์ปัญหา กรณีใกล้เคียง ฐานความช่วยเหลือ ธนาคารความรู้ เครื่องมือส่งเสริมความเข้าใจโปรแกรม ห้องปฏิบัติการส่งเสริมความเข้าใจโปรแกรม และแลกเปลี่ยนเรียนรู้ เพื่อเติมเต็มความรู้ที่ขาดหายไปได้ตามความต้องการตามศักยภาพของตนอย่างอิสระและยืดหยุ่น ผลการวิจัย พบว่า การใช้โมเดลฯของผู้เรียนแบ่งได้เป็น 3 ลักษณะ ได้แก่ ลักษณะที่ 1 พบว่า กลุ่มผู้เรียนที่มีความรู้เดิมอยู่ในระดับดี เกี่ยวกับหลักการ ทฤษฎี(Declarative knowledge) แต่ขาดความรู้ที่เป็นกระบวนการ(Procedural Knowledge) จะพบว่า ผู้เรียนมีการใช้โมเดลโดยเลือกเข้าศึกษาเพียง 2-3 องค์ประกอบ การเลือกองค์ประกอบเป็นไปตามลำดับ ลักษณะที่ 2 พบว่า ผู้เรียนกลุ่มที่มีความรู้เดิมเกี่ยวกับหลักการ ทฤษฎีและความรู้ที่เป็น

กระบวนการ อยู่ในระดับปานกลาง จะพบว่าผู้เรียนส่วนใหญ่มีการใช้โมเดลโดยเลือกเข้าศึกษาเกือบทุกองค์ประกอบ การเลือกองค์ประกอบเป็นไปตามลำดับ ลักษณะที่ 3 พบว่า ผู้เรียนกลุ่มที่มีความรู้เดิมทั้งที่เกี่ยวกับหลักการ ทฤษฎีและความรู้ที่เป็นกระบวนการ อยู่ในระดับต่ำ จะพบว่าผู้เรียนมีการใช้โมเดลโดยเลือกเข้าศึกษาทุก ๆองค์ประกอบ การเลือกองค์ประกอบไม่เป็นไปตามลำดับที่คงที่

(2) ด้านประโยชน์ที่ได้รับ พบว่า โมเดลฯเปิดโอกาสให้เกิดการปฏิสัมพันธ์(Interactive)ระหว่างผู้เรียนกับผู้สอน และผู้เรียนกับผู้เรียน โมเดลฯเป็นระบบเปิดซึ่งอนุญาตให้ผู้เรียนมีอิสระในการเข้าถึงข้อมูลได้ทั่วโลก โมเดลฯมีแหล่งการเรียนรู้ที่หลากหลายเพื่อให้ผู้เรียนสามารถสืบค้นได้ตามศักยภาพของตน ผู้เรียนสามารถเรียนรู้ได้โดยไม่มีข้อจำกัดทางสถานที่และเวลา ผู้เรียนเป็นผู้ควบคุมการเรียนรู้ตามความพร้อมความถนัดและความสนใจของตนเอง

ผลการตรวจสอบความตรงภายในโมเดล จากการประเมินโมเดลฯสำหรับผู้เชี่ยวชาญในด้านเนื้อหา ด้านสื่อ และด้านการออกแบบสิ่งแวดล้อมการเรียนรู้ การใช้โมเดลของผู้เรียน สรุปได้ว่าโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมมีความตรงภายในโมเดล

1.2.2 ผลการตรวจสอบความตรงภายนอกโมเดล

การตรวจสอบความตรงภายนอกโมเดล เป็นผลจากการสัมภาษณ์ผู้เรียนเกี่ยวกับความเข้าใจโปรแกรม ผลการสำรวจบริบทการเรียนด้วยโมเดลฯสำหรับผู้เรียน ผลสำรวจบริบทการจัดการเรียนรู้ด้วยโมเดลฯสำหรับผู้สอนและผลสัมฤทธิ์ทางการเรียนวิชา CT212 โครงสร้างโปรแกรม เรื่องการเขียนโปรแกรมเรียกตัวเอง ผลที่ได้ทำให้ทราบถึงผลกระทบของการใช้โมเดลฯที่ส่งผลต่อการเปลี่ยนพฤติกรรมของผู้เรียน ผู้สอนและองค์กร สรุปผลได้ดังนี้

1.2.2.1 ผลเกี่ยวกับความเข้าใจโปรแกรมของผู้เรียนที่เรียนด้วยโมเดลฯ เป็นผลจากการวิเคราะห์โปรโตคอลจากแบบสัมภาษณ์ผู้เรียนเกี่ยวกับความเข้าใจโปรแกรม สรุปผลได้ดังนี้กระบวนการทำความเข้าใจโปรแกรมของผู้เรียนแบ่งได้เป็น 3 ลักษณะ ดังนี้

ลักษณะที่ 1 พบว่า ผู้เรียนสร้างความเข้าใจโปรแกรมแบบล่างขึ้นบนตามกรอบของ Pennington(1987) เริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่คำสั่งโปรแกรม(Program code) ที่ผู้เรียนเพ่งพินิจ(Scan) และให้ความสนใจกับคำที่มีลักษณะเด่น(Beacon)ได้แก่ชื่อฟังก์ชัน ชื่อตัวแปร หรือคำเฉพาะ(Reserved word) ต่อจากนั้นผู้เรียนจะเริ่มสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model) โดยนำชื่อหรือคำที่สนใจมาจับคู่(Match)กับความรู้เดิมที่มีอยู่เป็นความรู้รูปแบบไวยากรณ์(Syntactic knowledge)เพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดเล็ก(Microstructure) จากนั้นผู้เรียนพิจารณาคำสั่งต่อ ๆไปตามลำดับเพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นกลุ่ม(Chunk) เช่นกลุ่มของคำสั่งที่เป็นโครงสร้าง หรือกลุ่มของการประกาศตัวแปรที่มีหลาย ๆคำสั่งที่ปรากฏในเอกสารโปรแกรม ซึ่งเรียกว่าโครงสร้างขนาดใหญ่(Macrostructure) ในขณะเดียวกันการสร้างสิ่งแทนความเข้าใจ

สถานการณ์เริ่มเกิดขึ้นเมื่อผู้เรียนพยายามนำความรู้เดิมที่มีลักษณะเป็นแผนงาน (Plan knowledge)หรือแนวคิดในการเขียนโปรแกรม(Programming concepts) มาจับคู่กับสิ่งแทนความเข้าใจคำสั่งที่สร้างขึ้นเพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans) ได้แก่วัตถุประสงค์ย่อย(Subgoal)ของคำสั่งหรือสมมุติฐาน(Hypothesis) จะพบว่าผู้เรียนจะกระทำกระบวนการทำความเข้าใจ(Comprehension process)อย่างต่อเนื่องและท้ายสุดผู้เรียนสามารถสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model)อยู่ในรูปของทิศทางหรือกระแสควบคุม(Control flow)ที่แสดงกระบวนการทำงานของคำสั่ง(Procedural) และสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ที่มีลักษณะเป็นภาพกระแสของข้อมูล(Dataflow view)ซึ่งสามารถเปลี่ยนแปลงไปตามกระบวนการทำงานของคำสั่ง จนกระทั่งผู้เรียนสามารถสรุปหน้าที่ของโปรแกรมได้(Functional)

ลักษณะที่ 2 พบว่า ผู้เรียนสร้างความเข้าใจโปรแกรมแตกต่างจากกรอบของ Pennington(1987) เริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่คำสั่งโปรแกรม (Program code) ที่ผู้เรียนเพ่งพินิจ(Scan)และให้ความสนใจกับคำที่มีลักษณะเด่น (Beacon)ได้แก่ชื่อฟังก์ชัน ชื่อตัวแปร หรือคำเฉพาะ(Reserved word) ต่อจากนั้นผู้เรียนจะเริ่มสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ก่อน โดยนำชื่อหรือคำที่สนใจไปเชื่อมโยง (Match)กับความรู้เดิมที่มีอยู่เป็นประสบการณ์ที่เกี่ยวกับปัญหา(Domain knowledge)ที่คุ้นเคย ผู้เรียนสามารถคาดเดาหรือตั้งสมมุติฐานถึงเหตุการณ์หรือวัตถุประสงค์ของกระบวนการทำงานของโปรแกรม(Functional)ได้โดยทันที ต่อจากนั้นผู้เรียนจะเริ่มตรวจสอบความถูกต้องของสมมุติฐาน(Hypothesis)ที่ได้คาดเดาไว้ โดยนำชื่อหรือคำที่สนใจมาจับคู่ (Match)กับความรู้เดิมที่มีอยู่เป็นความรู้รูปแบบไวยากรณ์(Syntactic knowledge)เพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดเล็ก(Microstructure) จากนั้นผู้เรียนพิจารณาคำสั่งต่อ ๆไปตามลำดับเพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นกลุ่ม(Chunk) เช่นกลุ่มของคำสั่งที่เป็นโครงสร้าง หรือกลุ่มของการประกาศตัวแปรที่มีหลาย ๆคำสั่งที่ปรากฏในเอกสารโปรแกรม ซึ่งเรียกว่าโครงสร้างขนาดใหญ่ (Macrostructure) ในขณะเดียวกันการสร้างสิ่งแทนความเข้าใจสถานการณ์เริ่มเกิดขึ้นเมื่อผู้เรียนพยายามนำความรู้เดิมที่มีลักษณะเป็นแผนงาน (Plan knowledge)หรือแนวคิดในการเขียนโปรแกรม(Programming concepts) มาจับคู่กับสิ่งแทนความเข้าใจคำสั่งที่สร้างขึ้นเพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans) ได้แก่วัตถุประสงค์ย่อย(Subgoal)ของคำสั่งหรือสมมุติฐาน(Hypothesis) จะพบว่าผู้เรียนจะกระทำกระบวนการทำความเข้าใจ(Comprehension process)อย่างต่อเนื่องและท้ายสุดผู้เรียนสามารถสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model)อยู่ในรูปของทิศทางหรือกระแสควบคุม(Control flow)ที่แสดงกระบวนการทำงานของคำสั่ง(Procedural) และสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ที่มีลักษณะเป็นภาพกระแสของข้อมูล(Dataflow view)ซึ่งสามารถเปลี่ยนแปลงไปตามกระบวนการทำงานของคำสั่ง จนกระทั่งผู้เรียนสามารถสรุปหน้าที่ของโปรแกรมได้(Functional)

ลักษณะที่ 3 พบว่า ผู้เรียนสร้างความเข้าใจโปรแกรมแตกต่างจากกรอบของ Pennington(1987) เริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่คำสั่งโปรแกรม

(Program code) ที่ผู้เรียนเพ่งพินิจ(Scan)และให้ความสนใจกับคำที่มีลักษณะเด่น (Beacon)ได้แก่ชื่อฟังก์ชัน ชื่อตัวแปร หรือคำเฉพาะ(Reserved word) ต่อจากนั้นผู้เรียนจะเริ่มสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ก่อน โดยนำชื่อหรือคำที่สนใจไปเชื่อมโยง (Match)กับความรู้เดิมที่มีอยู่เป็นประสบการณ์ที่เกี่ยวกับปัญหา(Domain knowledge)ที่คุ้นเคย ผู้เรียนสามารถคาดเดาหรือตั้งสมมุติฐานถึงเหตุการณ์หรือวัตถุประสงค์ของกระบวนการทำงานของโปรแกรม(Functional)ได้โดยทันที ต่อจากนั้นผู้เรียนจะเริ่มตรวจสอบความถูกต้องของสมมุติฐาน(Hypothesis)ที่ได้คาดเดาไว้ โดยนำชื่อหรือคำที่สนใจมาจับคู่ (Match)กับความรู้เดิมที่มีอยู่เป็นความรู้รูปแบบไวยากรณ์(Syntactic knowledge)เพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดเล็ก(Microstructure) จากนั้นผู้เรียนพิจารณาคำสั่งต่อ ๆไปตามลำดับเพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นกลุ่ม(Chunk) เช่นกลุ่มของคำสั่งที่เป็นโครงสร้าง หรือกลุ่มของการประกาศตัวแปรที่มีหลาย ๆคำสั่งที่ปรากฏในเอกสารโปรแกรม ซึ่งเรียกว่าโครงสร้างขนาดใหญ่ (Macrostructure) ในขณะเดียวกันการสร้างสิ่งแทนความเข้าใจสถานการณ์เริ่มเกิดขึ้นเมื่อผู้เรียนพยายามนำความรู้เดิมที่มีลักษณะเป็นสคริปต์แผนงาน (Plan knowledge)หรือแนวคิดในการเขียนโปรแกรม(Programming concepts) มาจับคู่กับสิ่งแทนความเข้าใจคำสั่งที่สร้างขึ้นเพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans) ได้แก่วัตถุประสงค์ย่อย(Subgoal)ของคำสั่งหรือสมมุติฐาน(Hypothesis) จะพบว่าผู้เรียนจะกระทำกระบวนการทำความเข้าใจ(Comprehension process)อย่างต่อเนื่องและท้ายสุดผู้เรียนสามารถสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model)อยู่ในรูปของทิศทางหรือกระแสควบคุม(Control flow)ที่แสดงกระบวนการทำงานของคำสั่ง(Procedural) และสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ที่มีลักษณะเป็นภาพกระแสของข้อมูล(Dataflow view)ซึ่งสามารถเปลี่ยนแปลงไปตามกระบวนการทำงานของคำสั่ง จนกระทั่งผู้เรียนสามารถสรุปหน้าที่ของโปรแกรมได้(Functional) ความแตกต่างระหว่างลักษณะที่ 2 และลักษณะที่ 3 นั้น กระบวนการทำความเข้าใจของผู้เรียนในลักษณะที่ 2 ผู้เรียนจะทำความเข้าใจโปรแกรมทั้งหมด แต่กระบวนการทำความเข้าใจของผู้เรียนในลักษณะที่ 3 ผู้เรียนจะมุ่งเน้นไปยังกลุ่มข้อความหรือฟังก์ชันที่มีความสำคัญเท่านั้น ผู้เรียนไม่ทำความเข้าใจโปรแกรมทั้งหมด

ผลการวิจัย พบว่า ผลของการเรียนด้วยโมเดลฯส่งผลต่อกระบวนการทำความเข้าใจโปรแกรมของผู้เรียนที่มีลักษณะแตกต่างกันขึ้นอยู่กับความรู้กระบวนการ(Procedural knowledge)ได้แก่ความรู้แผนงานหรือสคริปต์ ความรู้ที่เกี่ยวกับปัญหา(Domain knowledge) และความรู้แนวคิดในการเขียนโปรแกรมซึ่งเป็นความรู้เดิมที่ผู้เรียนมีอยู่ ถ้าผู้เรียนมีความรู้เดิมอยู่ในระดับดีจะสามารถนำชื่อหรือคำที่สนใจในโปรแกรมจับคู่กับความรู้เดิมที่มีอยู่ส่งผลให้ผู้เรียนสามารถคาดเดาและสร้างสิ่งแทนความรู้สถานการณ์ได้ แตกต่างจากผู้เรียนที่มีความรู้กระบวนการอยู่ในระดับต่ำต้องสร้างสิ่งแทนความรู้โปรแกรมและสิ่งแทนความรู้สถานการณ์อย่างเป็นลำดับจนกระทั่งสามารถเข้าใจโปรแกรมได้ในที่สุด

1.2.2.2 ผลด้านบริบทการเรียนด้วยโมเดลฯสำหรับผู้เรียน เป็นผลจากการศึกษาบริบทการเรียนด้วยโมเดลฯสำหรับผู้เรียนที่เรียนวิชา CT212 ของนักศึกษาวิทยาการคอมพิวเตอร์ภาคพิเศษ ปีการศึกษา 2552 ภาคเรียนที่ 1 จำนวน 36 คน ใน 4 ด้านได้แก่ ด้านข้อมูลทั่วไป ด้านบริบทการเรียน ด้านความสามารถในการจัดการเรียนการสอนของอาจารย์ และด้านประเมินตนเอง ผลการวิจัย พบว่า บริบทที่เหมาะสมกับการเรียนด้วยโมเดลฯ ดังรายละเอียดต่อไปนี้

(1) คุณลักษณะผู้เรียนที่เหมาะสม พบว่า ผู้เรียนควรเข้าชั้นเรียนและฝึกปฏิบัติอย่างสม่ำเสมอ และตรงเวลา มีความตั้งใจเรียนและมีความพึงพอใจในการเรียนรู้ด้วยโมเดลฯ มีความรู้ทั้ง 2 ลักษณะคือความรู้ทางทฤษฎี ได้แก่ ความรู้รูปแบบไวยากรณ์ภาษา C++ แนวคิดในการพัฒนาโปรแกรม การเขียนโปรแกรมเรียกตัวเอง และความรู้ทางปฏิบัติ ได้แก่ ทักษะและประสบการณ์ในการใช้คอมพิวเตอร์

(2) ด้านบริบทการเรียน พบว่า ควรเตรียมสภาพแวดล้อมที่สนับสนุนการเรียนด้วยโมเดลฯให้เหมาะสมกับผู้เรียน ได้แก่ ระบบคอมพิวเตอร์ที่มีสมรรถนะสูงเพื่อรองรับการเรียนด้วยโมเดลฯห้องเรียน อุณหภูมิ แสงสว่าง คุณภาพของเสียง อุปกรณ์สำหรับการเรียนการสอน แผนการเรียนเวลาพักเรียน การจัดกลุ่มผู้เรียน และจำนวนผู้ช่วยสอนที่เหมาะสม

1.2.2.3 ผลการประเมินบริบทการจัดการเรียนรู้ด้วยโมเดลฯสำหรับผู้สอน เป็นผลจากการศึกษาบริบทการจัดการเรียนรู้ด้วยโมเดลฯสำหรับผู้สอน ใน 3 ด้านได้แก่ ด้านข้อมูลทั่วไปของผู้สอน ด้านความคิดเห็นของผู้สอน และด้านบริบทการถ่ายโยง ผลการวิจัยพบว่าบริบทการจัดการเรียนรู้ที่เหมาะสมกับการเรียนด้วยโมเดลฯ ดังรายละเอียดต่อไปนี้

(1) คุณลักษณะของผู้สอนที่เหมาะสม พบว่า ผู้สอนควรมี มีประสบการณ์ในการสอนเกี่ยวกับการเขียนโปรแกรมภาษา C++ โดยเฉพาะเรื่องการเขียนโปรแกรมเรียกตัวเอง มีความสามารถในการวางแผนการจัดการเรียนการสอน การเลือกกิจกรรมการเรียนการสอน สื่อหรือแหล่งการเรียนรู้ที่เหมาะสมให้กับผู้เรียน รวมทั้งมีความรับผิดชอบ

(2) การสนับสนุนของผู้สอน พบว่า ผู้สอนต้องเสียสละอุทิศเวลาเพิ่มมากขึ้นกว่าการสอนปกติ เช่นการจัดเตรียมความพร้อมในด้านต่าง ๆ ได้แก่ห้องปฏิบัติการคอมพิวเตอร์ โมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม การวางแผนการจัดการเรียนการสอนที่ใช้โมเดลฯ การกำกับติดตามการเรียนรู้ของผู้เรียน การตรวจผลงาน ฯลฯ

(3) บริบทการจัดการเรียนรู้ที่ใช้โมเดลฯที่เหมาะสม พบว่าผู้บริหาร และผู้ร่วมงานควรให้การสนับสนุนการเรียนการสอนด้วยโมเดลฯในขณะเดียวกันผู้บริหารควรมีการสนับสนุนค่าใช้จ่ายหรืองบประมาณสำหรับการพัฒนาโมเดลฯรวมทั้งสิ่งอำนวยความสะดวกต่าง ๆโดยเฉพาะระบบคอมพิวเตอร์ให้ทันสมัยและมีประสิทธิภาพเพื่อรองรับการใช้โมเดลฯของผู้เรียนจำนวนมาก

1.2.2.4 ผลสัมฤทธิ์ทางการเรียนวิชา CT212 เรื่องการเขียนโปรแกรมเรียกตัวเอง เป็นผลจากการทดสอบในรายวิชา CT212 เรื่องการเขียนโปรแกรมเรียกตัวเอง ของนักศึกษาภาคพิเศษภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง ภาคเรียนที่ 1 ปีการศึกษา 2552 จำนวน 36 คน ทำการทดสอบก่อนและหลังเรียนโดยใช้แบบทดสอบชุดเดียวกัน เป็นข้อสอบอัตนัยจำนวน 10 ข้อ รวม 30 คะแนน สรุปผลดังแสดงในตารางที่ 5.2

ตารางที่ 5.2 แสดงค่าคะแนนเฉลี่ย และส่วนเบี่ยงเบนมาตรฐาน ในระยะที่ 2

	$\sum X$	$\overline{X}$	S.D.
คะแนนทดสอบก่อนเรียน	360	10.00	1.57
คะแนนทดสอบหลังเรียน	765	21.25	1.25

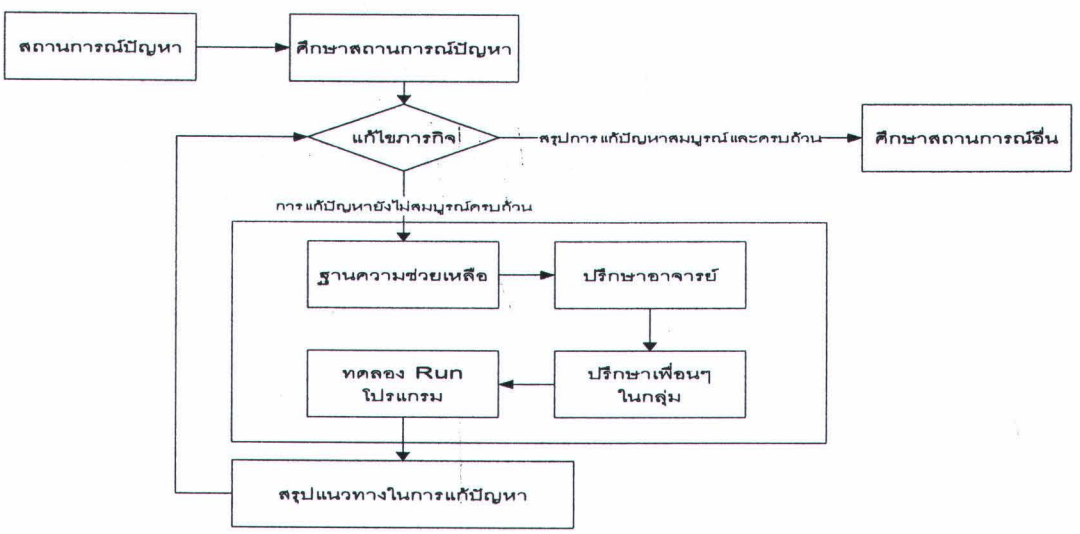
ผลการวิจัยพบว่า ผู้เรียนมีผลสัมฤทธิ์ทางการเรียนคะแนนเฉลี่ยการทดสอบหลังเรียนเท่ากับ 21.25 ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.25 สูงกว่าคะแนนเฉลี่ยการทดสอบก่อนเรียน ซึ่งค่าคะแนนเฉลี่ยการทดสอบก่อนเรียนเท่ากับ 10.00 ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.57 นอกจากนี้ยังพบว่าผู้เรียนมีผลสัมฤทธิ์ทางการเรียนคิดเป็นร้อยละ 70 สูงกว่าเกณฑ์ที่กำหนด ดังนั้นผู้เรียนมีผลสัมฤทธิ์ทางการเรียน เรื่องการเขียนโปรแกรมเรียกตัวเองสูงขึ้น

1.3 ระยะที่ 3 การใช้โมเดล (Model use phase)

การใช้โมเดล เป็นระยะที่ผู้วิจัยศึกษากระบวนการใช้โมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม ศึกษาปัจจัยที่ส่งเสริมในการใช้โมเดลที่ประสบความสำเร็จ และ ศึกษาผลสำเร็จของการใช้โมเดลฯ ได้แก่ความเข้าใจโปรแกรม ความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ และผลสัมฤทธิ์ทางการเรียน ผู้วิจัยได้ดำเนินการตามลำดับดังนี้ 1) การสัมภาษณ์ผู้เรียนเกี่ยวกับการใช้โมเดลฯ 2) การบันทึกการใช้โมเดลฯของผู้เรียน 3) การสำรวจบริบทการใช้โมเดลฯสำหรับผู้เรียน 4) การสำรวจบริบทการใช้โมเดลฯสำหรับผู้สอน 5) การสัมภาษณ์ผู้เรียนเกี่ยวกับความเข้าใจโปรแกรม 6) การทดสอบผลสัมฤทธิ์ทางการเรียนรู้ของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม 7) การสำรวจความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ ผลที่ได้ทำให้ทราบถึงกระบวนการที่ควรปฏิบัติตาม(Follow)เมื่อใช้โมเดลฯ ปัจจัยที่ส่งเสริม(Promote)ในการใช้โมเดลฯที่ประสบความสำเร็จ และสามารถอธิบายความสำเร็จหรือความล้มเหลวที่สามารถเกิดขึ้นเมื่อใช้โมเดลฯ สรุปผลได้ดังนี้

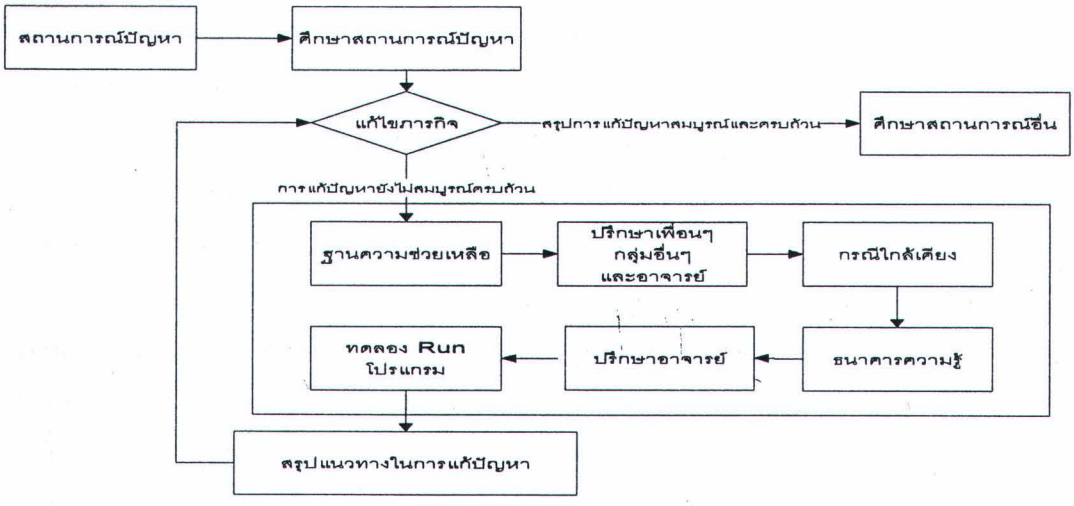
1.3.1 ผลการศึกษากระบวนการใช้โมเดล เป็นผลจากการศึกษาการใช้โมเดลฯของผู้เรียน การถอดโปรโตคอลแบบสัมภาษณ์ผู้เรียนเกี่ยวกับการใช้โมเดลฯ สามารถสรุปกระบวนการใช้โมเดลฯแบ่งได้เป็น 3 ลักษณะได้แก่

ลักษณะที่ 1 พบว่า กลุ่มผู้เรียนที่มีความรู้เดิมอยู่ในระดับดี เกี่ยวกับหลักการทฤษฎี(Declarative knowledge)เช่น ความรู้รูปแบบไวยากรณ์ภาษา C++ แต่ขาดความรู้ที่เป็นกระบวนการ(Procedural Knowledge) เช่น ความรู้ที่มีลักษณะที่เป็นแผนงานหรือสคริปต์แนวคิดในการเขียนโปรแกรม ความรู้ที่เกี่ยวข้องกับปัญหา จะพบว่า ผู้เรียนมีการใช้โมเดลโดยเลือกเข้าศึกษาบางองค์ประกอบ จะสังเกตเห็นว่าการเลือกองค์ประกอบเป็นตามลำดับโดยเลือกแต่ละองค์ประกอบเพียงหนึ่งครั้ง ดังตัวอย่างเช่น เมื่อทำการศึกษาสถานการณ์ปัญหาแล้วไม่สามารถแก้ปัญหาตามภารกิจได้ ต่อมาผู้เรียนส่วนใหญ่ เลือกใช้องค์ประกอบ ฐานความช่วยเหลือ แลกเปลี่ยนเรียนรู้ เครื่องมือทางปัญญา เพื่อเติมเต็มความรู้ที่เป็นกระบวนการ ซึ่งช่วยสนับสนุนให้ผู้เรียนสามารถแก้ไขปัญหาตามภารกิจการเรียนรู้ได้ สังเกตได้ว่าไม่ได้เข้าไปศึกษาธนาคารความรู้ กรณีใกล้เคียง และห้องปฏิบัติการส่งเสริมความเข้าใจโปรแกรม ดังแสดงในภาพที่ 5.1



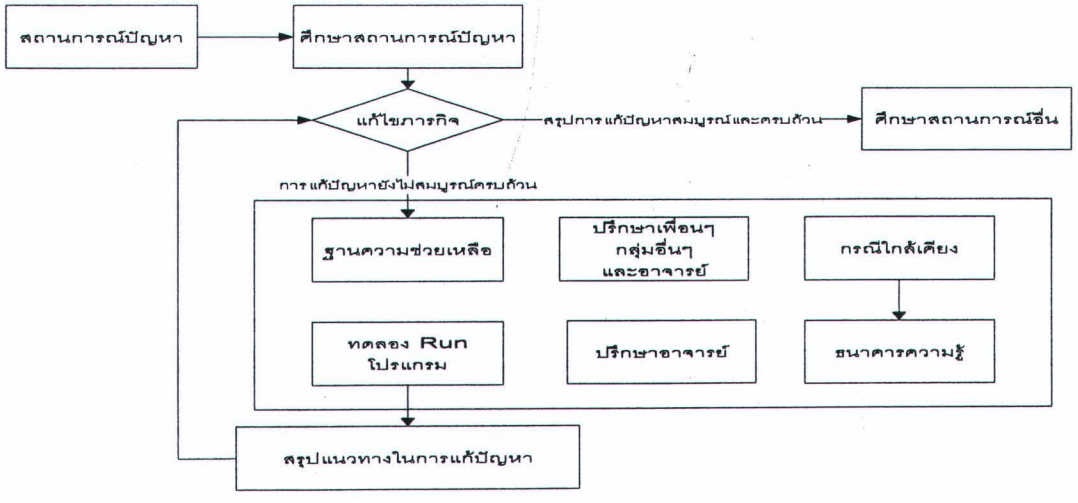
ภาพที่ 5.1 แสดงกระบวนการใช้โมเดลฯของผู้เรียนลักษณะที่ 1

ลักษณะที่ 2 พบว่า กลุ่มผู้เรียนที่มีความรู้เดิมเกี่ยวกับหลักการ ทฤษฎีและความรู้ที่เป็นกระบวนการอยู่ในระดับปานกลาง จะพบว่า ผู้เรียนส่วนใหญ่มีการใช้โมเดลโดยเลือกเข้าศึกษาเกือบทุกองค์ประกอบ จะสังเกตเห็นว่าผู้เรียนทำการเลือกองค์ประกอบตามลำดับ ดังตัวอย่าง เช่น เมื่อทำการศึกษาสถานการณ์ปัญหาแล้วไม่สามารถแก้ปัญหาตามภารกิจได้ ต่อมาผู้เรียนส่วนใหญ่ เลือกใช้องค์ประกอบ ฐานความช่วยเหลือ แลกเปลี่ยนเรียนรู้ เครื่องมือทางปัญญา เพื่อเติมเต็มความรู้ที่เป็นกระบวนการ ซึ่งช่วยสนับสนุนให้ผู้เรียนสามารถแก้ไขปัญหาตามภารกิจการเรียนรู้ได้ระดับหนึ่ง ผู้เรียนเลือกองค์ประกอบกรณีใกล้เคียง หรือ องค์ประกอบธนาคารความรู้ เพื่อเติมเต็มความรู้ที่เป็นหลักการ ทฤษฎี ซึ่งส่งเสริมการปรับสมดุลทางปัญญาให้ผู้เรียนในที่สุดผู้เรียนสามารถแก้ไขปัญหาตามภารกิจได้สำเร็จ ดังแสดงในภาพที่ 5.2



ภาพที่ 5.2 แสดงกระบวนการใช้โมเดลฯของผู้เรียนลักษณะที่ 2

ลักษณะที่ 3 พบว่า กลุ่มผู้เรียนที่มีความรู้เดิมทั้งที่เกี่ยวกับหลักการ ทฤษฎี และความรู้ที่เป็นกระบวนการอยู่ในระดับต่ำ จะพบว่า ผู้เรียนมีการใช้โมเดลโดยเลือกเข้าศึกษาเกือบทุก ๆองค์ประกอบ และเข้าไปศึกษาในแต่ละองค์ประกอบหลายครั้ง บางครั้งเลือกสลับไปมาหลายรอบ เพื่อทำความเข้าใจ จะสังเกตเห็นว่า การเลือกองค์ประกอบไม่เป็นไปตามลำดับที่คงที่ ดังตัวอย่างเช่น เมื่อทำการศึกษาสถานการณ์ปัญหาแล้วไม่สามารถแก้ปัญหาตามภารกิจได้ ต่อมาผู้เรียนส่วนใหญ่ เลือกใช้องค์ประกอบ ฐานความช่วยเหลือ แลกเปลี่ยนเรียนรู้ เครื่องมือทางปัญญา กรณีใกล้เคียง และธนาคารความรู้ โดยเลือกแต่ละองค์ประกอบหลายรอบซ้ำ ๆหรือบางครั้งสลับไปมาเพื่อเติมเต็มความรู้ที่เป็นหลักการ ทฤษฎี และความรู้ที่เป็นกระบวนการ ซึ่งช่วยสนับสนุนให้ผู้เรียนสามารถแก้ไขปัญหาตามภารกิจการเรียนรู้ได้ในที่สุด ดังแสดงในภาพที่ 5.3



ภาพที่ 5.3 แสดงกระบวนการใช้โมเดลฯของผู้เรียนลักษณะที่ 3

จากผลการศึกษากระบวนการใช้โมเดลฯ พบว่า ผู้เรียนมีกระบวนการใช้โมเดลแตกต่างกัน 3 ลักษณะขึ้นอยู่กับความรู้เดิมของผู้เรียน เกี่ยวกับหลักการ ทฤษฎี(Declarative knowledge)เช่น ความรู้รูปแบบไวยากรณ์ภาษา C++ แต่ขาดความรู้ที่เป็นกระบวนการ (Procedural Knowledge) เช่น ความรู้ที่มีลักษณะที่เป็นแผนงานหรือสคริปต์ แนวคิดในการเขียนโปรแกรม และความรู้ที่เกี่ยวข้องกับปัญหา ถ้าผู้เรียนมีความรู้เดิมในระดับดีมีการใช้โมเดลฯตามลักษณะที่ 1 ถ้าผู้เรียนมีความรู้เดิมในระดับปานกลางมีการใช้โมเดลฯตามลักษณะที่ 2 แต่ถ้าผู้เรียนมีความรู้เดิมในระดับต่ำมีการใช้โมเดลฯตามลักษณะที่ 3 จากการสังเกตการใช้โมเดลฯของผู้เรียนเห็นได้ว่า ผู้เรียนเกือบทุกกลุ่มเลือกองค์ ประกอบฐานความช่วยเหลือ แลกเปลี่ยนเรียนรู้ และเครื่องมือส่งเสริมความเข้าใจโปรแกรม ในการแก้ปัญหาเพราะองค์ประกอบฐานความช่วยเหลือ แนะนำแนวความคิดและความรู้ในการแก้ไขปัญหาเป็นความรู้ที่เกี่ยวข้องกับปัญหา(Problem domain) แลกเปลี่ยนเรียนรู้ เป็นองค์ประกอบที่ช่วยผู้เรียนในการระดมสมองช่วยกันแก้ไขปัญหา แบ่งปันความรู้ในโจทย์ที่ไม่สามารถแก้ปัญหาได้หรือปรึกษาผู้เชี่ยวชาญ เพื่อให้ได้รับคำแนะนำในการแก้ปัญหาที่ดี องค์ประกอบเครื่องมือส่งเสริมความเข้าใจโปรแกรมนั้นผู้เรียนทุกกลุ่มจะเลือกใช้ตัวแปลภาษา C++ เพื่อทดสอบความถูกต้องของโปรแกรมเพื่อตรวจสอบกระบวนการคิดในการแก้ปัญหาของผู้เรียนสำหรับองค์ประกอบอื่น ๆผู้เรียนสามารถเลือกศึกษาตามความต้องการตามศักยภาพของตนได้อย่างอิสระและยืดหยุ่น

1.3.2 ผลการศึกษาปัจจัยที่ส่งเสริมให้การใช้โมเดลที่ประสบความสำเร็จ เป็นผลจากการศึกษาบริบทการใช้โมเดลฯของผู้เรียนที่เรียนวิชา CT212 ของนักศึกษาวิทยาการคอมพิวเตอร์ ภาคปกติ ภาคเรียนที่ 2 ปีการศึกษา 2552 จำนวน 23 คน และการศึกษาบริบทการใช้โมเดลฯของผู้สอน ผลการวิจัย พบว่า ปัจจัยที่ส่งเสริมให้การใช้โมเดลฯที่ประสบความสำเร็จ ประกอบด้วย

1.3.2.1 ด้านผู้เรียน พบว่า คุณลักษณะของผู้เรียนที่ช่วยเอื้อให้การใช้โมเดลที่ประสบความสำเร็จประกอบด้วย

(1) คุณลักษณะผู้เรียนที่เหมาะสม ได้แก่ การเข้าชั้นเรียนและฝึกปฏิบัติอย่างสม่ำเสมอ และตรงเวลา มีความตั้งใจเรียนและมีความพึงพอใจในการเรียนรู้ด้วยโมเดลฯ

(2) ความรู้ พบว่า ผู้เรียนควรมีความรู้ทั้ง 2 ลักษณะ คือ ความรู้เดิมเกี่ยวกับทฤษฎีหลักการโปรแกรม ได้แก่ การเขียนโปรแกรมเรียกตัวเอง การพัฒนาโปรแกรมภาษา C++ และความรู้ในการปฏิบัติกับระบบคอมพิวเตอร์

1.3.2.2 ด้านผู้สอน พบว่า คุณลักษณะของผู้สอนที่ช่วยส่งเสริมให้การใช้โมเดลที่ประสบความสำเร็จ ประกอบด้วย

(1) คุณลักษณะทั่วไปผู้สอน ได้แก่ มีทักษะในการใช้ระบบคอมพิวเตอร์ มีประสบการณ์ในการสอนเกี่ยวกับการเขียนโปรแกรม ในระดับดี โดยเฉพาะ เรื่องการเขียนโปรแกรมเรียกตัวเอง เขียนโปรแกรมภาษา C++ และมีความรับผิดชอบในการสอน

(2) ความสามารถทางด้านการจัดการเรียนการสอน ได้แก่ ความสามารถในการวางแผนการจัดการเรียนการสอน การเลือกกิจกรรมการเรียนการสอน สื่อหรือแหล่งการเรียนรู้ที่เหมาะสมกับผู้เรียน

(3) การสนับสนุนของผู้สอน พบว่า การใช้โมเดลที่ประสบความสำเร็จ ผู้สอนต้องเสียสละอุทิศเวลาเพิ่มมากขึ้นกว่าการสอนปกติ เช่น การจัดเตรียมความพร้อมด้านต่าง ๆ ได้แก่ ห้องปฏิบัติการคอมพิวเตอร์ สิ่งแวดล้อมทางการเรียนรู้ ฯ การวางแผนการจัดการเรียนการสอนที่ใช้โมเดลฯ การกำกับติดตามการเรียนรู้ของผู้เรียน การตรวจผลงาน ฯลฯ

1.3.2.3 ด้านสภาพแวดล้อมที่สนับสนุนการใช้โมเดลฯ พบว่า ผู้สอนควรเตรียมสภาพแวดล้อมในการใช้โมเดลฯ เช่น ระบบคอมพิวเตอร์ที่มีสมรรถนะสูงเพื่อรองรับการใช้โมเดลฯ ห้องเรียน ที่นั่ง แสงสว่าง เครื่องเสียงให้เหมาะสมกับสภาพแวดล้อมในการเรียน

1.3.3.3 ด้านบริบทการถ่ายโยง พบว่า

(1) การสนับสนุนของผู้บริหารและผู้ร่วมงาน พบว่า ผู้บริหารและผู้ร่วมงานควรให้การสนับสนุนการใช้โมเดลฯในการเรียนการสอน

(2) การถ่ายโยง พบว่า ผู้บริหารควรมีการสนับสนุนค่าใช้จ่ายหรืองบประมาณสำหรับการพัฒนาและการใช้โมเดลฯรวมทั้งสิ่งอำนวยความสะดวกต่าง ๆโดยเฉพาะระบบคอมพิวเตอร์ให้มีประสิทธิภาพ

1.3.3 ผลการศึกษาผลสำเร็จของการใช้โมเดลฯ

ผลการศึกษาผลสำเร็จของการใช้โมเดลฯ โดยผลการศึกษาดังกล่าวได้จากการสัมภาษณ์ผู้เรียนเกี่ยวกับความเข้าใจโปรแกรม การทดสอบผลสัมฤทธิ์ทางการเรียนวิชา CT212 เรื่องการเขียนโปรแกรมเรียกตัวเอง และความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ ผลที่ได้ทำให้สามารถอธิบายความสำเร็จหรือความล้มเหลวที่สามารถเกิดขึ้นเมื่อใช้โมเดลฯ สรุปผลได้ดังนี้

1.3.3.1 ผลเกี่ยวกับความเข้าใจโปรแกรมของผู้เรียนที่เรียนด้วยโมเดลฯ เป็นผลจากวิเคราะห์โปรโตคอลจากแบบสัมภาษณ์ผู้เรียนเกี่ยวกับความเข้าใจโปรแกรม สรุปผลได้ดังนี้กระบวนการทำความเข้าใจโปรแกรมของผู้เรียนแบ่งได้เป็น 2 ลักษณะ ดังนี้

ลักษณะที่ 1 พบว่า ผู้เรียนสร้างความเข้าใจโปรแกรมแบบล่างขึ้นบนตามกรอบของ Pennington(1987) เริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่ คำสั่งโปรแกรม(Program code) ที่ผู้เรียนเพ่งพินิจ(Scan) และให้ความสนใจกับคำที่มีลักษณะเด่น(Beacon)ได้แก่ชื่อฟังก์ชัน ชื่อตัวแปร หรือคำเฉพาะ(Reserved word) ต่อจากนั้นผู้เรียนจะเริ่มสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model) โดยนำชื่อหรือคำที่สนใจมาจับคู่(Match)กับความรู้เดิมที่มีอยู่เป็นความรู้รูปแบบไวยากรณ์(Syntactic knowledge)เพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดเล็ก(Microstructure) จากนั้นผู้เรียนพิจารณาคำสั่งต่อ ๆไปตามลำดับเพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นกลุ่ม(Chunk) เช่นกลุ่มของคำสั่งที่เป็น

โครงสร้าง หรือกลุ่มของการประกาศตัวแปรที่มีหลาย ๆคำสั่งที่ปรากฏในเอกสารโปรแกรม ซึ่งเรียกว่าโครงสร้างขนาดใหญ่(Macrostructure) ในขณะเดียวกันการสร้างสิ่งแทนความเข้าใจสถานการณ์เริ่มเกิดขึ้นเมื่อผู้เรียนพยายามนำความรู้เดิมที่มีลักษณะเป็นแผนงาน (Plan knowledge)หรือแนวคิดในการเขียนโปรแกรม(Programming concepts) มาจับคู่กับสิ่งแทนความเข้าใจคำสั่งที่สร้างขึ้นเพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans) ได้แก่วัตถุประสงค์ย่อย(Subgoal)ของคำสั่งหรือสมมุติฐาน(Hypothesis) จะพบว่าผู้เรียนจะกระทำกระบวนการทำความเข้าใจ(Comprehension process)อย่างต่อเนื่องและท้ายสุดผู้เรียนสามารถสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model)อยู่ในรูปของทิศทางหรือกระแสควบคุม(Control flow)ที่แสดงกระบวนการทำงานของคำสั่ง(Procedural) และสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ที่มีลักษณะเป็นภาพกระแสของข้อมูล(Dataflow view)ซึ่งสามารถเปลี่ยนแปลงไปตามกระบวนการทำงานของคำสั่ง จนกระทั่งผู้เรียนสามารถสรุปหน้าที่ของโปรแกรมได้(Functional)

ลักษณะที่ 2 พบว่า ผู้เรียนสร้างความเข้าใจโปรแกรมแตกต่างจากกรอบของ Pennington(1987) เริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่คำสั่งโปรแกรม(Program code) ที่ผู้เรียนเพ่งพินิจ(Scan)และให้ความสนใจกับคำที่มีลักษณะเด่น(Beacon)ได้แก่ชื่อฟังก์ชัน ชื่อตัวแปร หรือคำเฉพาะ(Reserved word) ต่อจากนั้นผู้เรียนจะเริ่มสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ก่อน โดยนำชื่อหรือคำที่สนใจไปเชื่อมโยง(Match)กับความรู้เดิมที่มีอยู่เป็นประสบการณ์ที่เกี่ยวกับปัญหา(Domain knowledge)ที่คุ้นเคย ผู้เรียนสามารถคาดเดาหรือตั้งสมมุติฐานถึงเหตุการณ์หรือวัตถุประสงค์ของกระบวนการทำงานของโปรแกรม(Functional)ได้โดยทันที ต่อจากนั้นผู้เรียนจะเริ่มตรวจสอบความถูกต้องของสมมุติฐาน(Hypothesis)ที่ได้คาดเดาไว้ โดยนำชื่อหรือคำที่สนใจมาจับคู่ (Match)กับความรู้เดิมที่มีอยู่เป็นความรู้รูปแบบไวยากรณ์(Syntactic knowledge)เพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดเล็ก(Microstructure) จากนั้นผู้เรียนพิจารณาคำสั่งต่อ ๆไปตามลำดับเพื่อสร้างสิ่งแทนความเข้าใจคำสั่งเป็นกลุ่ม(Chunk) เช่นกลุ่มของคำสั่งที่เป็นโครงสร้าง หรือกลุ่มของการประกาศตัวแปรที่มีหลาย ๆคำสั่งที่ปรากฏในเอกสารโปรแกรม ซึ่งเรียกว่าโครงสร้างขนาดใหญ่(Macrostructure) ในขณะเดียวกันการสร้างสิ่งแทนความเข้าใจสถานการณ์เริ่มเกิดขึ้นเมื่อผู้เรียนพยายามนำความรู้เดิมที่มีลักษณะเป็นแผนงาน (Plan knowledge)หรือแนวคิดในการเขียนโปรแกรม(Programming concepts) มาจับคู่กับสิ่งแทนความเข้าใจคำสั่งที่สร้างขึ้นเพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans) ได้แก่วัตถุประสงค์ย่อย(Subgoal)ของคำสั่งหรือสมมุติฐาน(Hypothesis) จะพบว่าผู้เรียนจะกระทำกระบวนการทำความเข้าใจ(Comprehension process)อย่างต่อเนื่องและท้ายสุดผู้เรียนสามารถสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model)อยู่ในรูปของทิศทางหรือกระแสควบคุม(Control flow)ที่แสดงกระบวนการทำงานของคำสั่ง(Procedural) และสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ที่มีลักษณะเป็นภาพกระแสของข้อมูล(Dataflow view)ซึ่งสามารถเปลี่ยนแปลงไปตามกระบวนการทำงานของคำสั่ง จนกระทั่งผู้เรียนสามารถสรุปหน้าที่ของโปรแกรมได้(Functional)

จากผลการศึกษากระบวนการทำความเข้าใจโปรแกรม พบว่าผู้เรียนมีกระบวนการทำความเข้าใจโปรแกรมแตกต่างกัน 2 ลักษณะขึ้นอยู่กับความรู้กระบวนการ (Procedural knowledge)ได้แก่ความรู้แผนงานหรือสคริปต์ ความรู้ที่เกี่ยวกับปัญหา(Domain knowledge) และความรู้แนวคิดในการเขียนโปรแกรมซึ่งเป็นความรู้เดิมที่ผู้เรียนมีอยู่ ถ้าผู้เรียนมีความรู้เดิมอยู่ในระดับดีจะสามารถนำชื่อหรือคำที่สนใจในโปรแกรมจับคู่กับความรู้เดิมที่มีอยู่ ส่งผลให้ผู้เรียนสามารถคาดเดาและสร้างสิ่งแทนความรู้สถานการณ์ได้ แตกต่างจากผู้เรียนที่มีความรู้กระบวนการอยู่ในระดับต่ำต้องสร้างสิ่งแทนความรู้โปรแกรมและสิ่งแทนความรู้สถานการณ์อย่างเป็นลำดับจนกระทั่งสามารถเข้าใจโปรแกรมได้ในที่สุด ดังข้อมูลการสัมภาษณ์ที่ว่า

บรรทัดที่สนใจเป็น int main() เพราะเป็นจุดเริ่มต้นของโปรแกรม จะประกอบไปด้วยกลุ่มของคำสั่งหลาย ๆคำสั่งที่สำคัญ ส่วนของคำสั่ง input ส่วนของคำสั่งประมวลผล ส่วนของคำสั่งในการแสดงผลลัพธ์ออกทางจอภาพ จะอยู่ที่ฟังก์ชัน main ทั้งหมด ทำให้เราสามารถรู้ได้ว่าโปรแกรมนี้มีวัตถุประสงค์ทำอะไร ก่อนอื่นต้องพิจารณาคำสั่งที่อยู่ภายในฟังก์ชัน main ก่อนคำสั่งภายในฟังก์ชัน main นั้นก็เริ่มพิจารณาการทำงานใน { } โดยเริ่มจากบรรทัดแรก เป็นการประกาศตัวแปร x เป็นอาเรย์หนึ่งมิติขนาด 10 บรรทัดที่ 2 เป็นการประกาศ ตัวแปร sum ให้มีค่าเท่ากับ 0 บรรทัดที่ 3 เป็นการประกาศตัวแปร n เป็นเลขจำนวนเต็ม ให้มีค่าเริ่มต้นเท่ากับ 0 การที่จะรู้ว่าตัวแปรที่กำหนดในโปรแกรมทำอะไร ต้องพิจารณาคำสั่งต่อไปในที่นี้คือ input(x, n) เป็นการเรียกใช้ฟังก์ชัน โดยคำว่า input เป็นชื่อของฟังก์ชัน การทำงานภายในฟังก์ชันเมื่อพิจารณาแล้วเป็นทำงานที่เป็นการอ้างอิงตัวแปร x และ n โดยทำการรับข้อมูลที่เป็นเลขจำนวนเต็มทางแป้นพิมพ์เก็บในตัวแปรอาเรย์ x จำนวน n ตัว สามารถสรุปได้ว่าตัวแปร x ใช้เก็บเลขจำนวนเต็มที่เรียงต่อ ๆกันเป็นจำนวน n ตัว แต่ตัวแปร sum กำหนดมาทำไมยังไม่ทราบ ถ้าอยากรู้ว่าตัวแปร sum มีวัตถุประสงค์ทำอะไรให้ดูจากคำสั่งต่อไปในที่นี้เห็นว่า Sum = findSum(x, n); คำสั่งนี้เป็นการหาค่าของ Sum ซึ่งได้จากการเรียกใช้ฟังก์ชัน findSum ซึ่งมีการส่งผ่านข้อมูล x และ n จาก main อ้างอิงกับตัวแปร x และ n ในฟังก์ชัน findSum โดยข้อมูล x จะมีข้อมูลอยู่แล้วจำนวน n ตัวและไม่มีการเปลี่ยนแปลงค่าจากคำว่า const ที่มีความหมายว่าคงที่ การทำงานของ findSum นั้นจะมีการนำค่า n ไปเปรียบเทียบเพื่อเลือกคำสั่งในการทำงาน โดยจะทำการส่งผ่านค่ากลับไปยังจุดเรียกใช้ แต่ในการทำงานนี้จะมีการทำงานหลายรอบเนื่องจากถูกควบคุมโดยค่าของ n กรณีที่ n มีค่าเท่ากับ 1 จึงจะหยุดการทำงาน แต่ถ้าไม่ใช่ 1 จะกระทำซ้ำโดยมีการบวกสะสมค่าที่เก็บในตัวแปรอาเรย์ x ทีละค่า ๆจนกระทั่งครบทุกช่อง และส่งผ่านค่าผลรวมสุดท้ายกลับมาให้ตัวแปร Sum ในฟังก์ชัน main สำหรับคำสั่ง cout<<"Sum= "<<sum<<endl; เป็นการแสดงผลออกทางจอภาพโดยพิมพ์ข้อความว่า "Sum=" ออกมาก่อน และพิมพ์ค่าของตัวแปร sum ที่เก็บในหน่วยความจำตามมาด้วย return 0 ; เป็นการจบโปรแกรม จากการติดตามการทำงานของคำสั่ง และกลุ่มคำสั่งต่าง ๆที่อยู่ภายในฟังก์ชัน main เป็นลำดับ พิจารณาจากวัตถุประสงค์ของตัวแปร x , n และ s จนกระทั่งพิมพ์ค่า s ออกทางจอภาพ สามารถสรุปได้ว่าโปรแกรมนี้มีวัตถุประสงค์ในการหาค่า

ผลรวมของตัวเลขจำนวนเต็ม n จำนวนโดยให้ผู้ใช้ป้อนข้อมูล ไม่เกิน 10 จำนวน ที่ผู้ใช้ป้อนทางแป้นพิมพ์

1.3.3.2 ผลสัมฤทธิ์ทางการเรียนวิชา CT212 เรื่องการเขียนโปรแกรมเรียกตัวเอง เป็นผลจากการทดสอบในรายวิชา CT212 เรื่องการเขียนโปรแกรมเรียกตัวเอง ของนักศึกษาภาคปกติ ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยรามคำแหง ปีการศึกษา 2552 ภาคเรียนที่ 2จำนวน 23 คน ทำการทดสอบก่อนและหลังเรียนโดยใช้แบบทดสอบชุดเดียวกัน เป็นข้อสอบอัตนัยจำนวน 10 ข้อ รวม 30 คะแนน สรุปผลดังแสดงในตารางที่ 5.3

ตารางที่ 5.3 แสดงค่าคะแนนเฉลี่ย และส่วนเบี่ยงเบนมาตรฐาน ในระยะที่ 3

	$\sum X$	$\overline{X}$	S.D.
คะแนนทดสอบก่อนเรียน	261	11.35	1.09
คะแนนทดสอบหลังเรียน	501	21.78	1.47

ผลการวิจัยพบว่า ผู้เรียนมีผลสัมฤทธิ์ทางการเรียนคะแนนเฉลี่ยการทดสอบหลังเรียนเท่ากับ 21.78 ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.47 สูงกว่าคะแนนเฉลี่ยการทดสอบก่อนเรียน ซึ่งค่าคะแนนเฉลี่ยการทดสอบก่อนเรียนเท่ากับ 11.35 ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.09 นอกจากนี้ยังพบว่าผู้เรียนมีผลสัมฤทธิ์ทางการเรียนคิดเป็นร้อยละ 70 สูงกว่าเกณฑ์ที่กำหนด ดังนั้นผู้เรียนมีผลสัมฤทธิ์ทางการเรียน เรื่องการเขียนโปรแกรมเรียกตัวเองสูงขึ้น

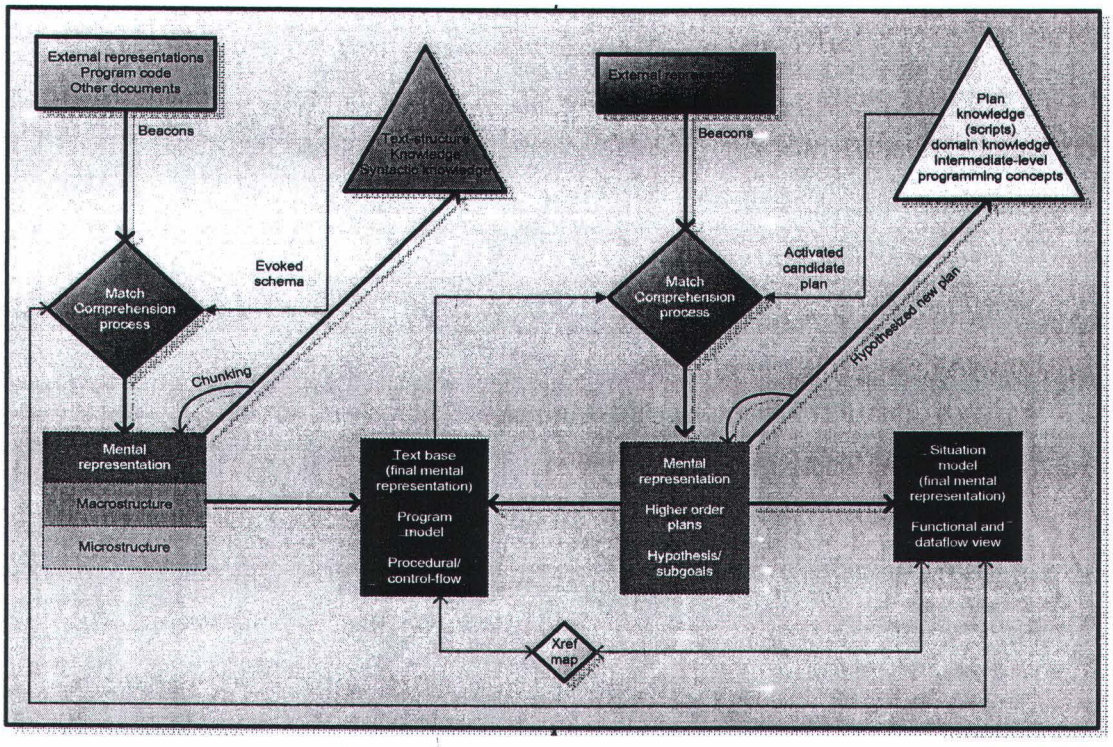
1.3.3.3 ผลสำรวจความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ เป็นผลจากการศึกษาความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ ด้านการใช้โมเดลฯ ด้านประโยชน์ที่ได้รับ สรุปผลได้ดังนี้

(1) ด้านการใช้โมเดลฯ พบว่า ผู้เรียนส่วนมากสามารถเลือกใช้องค์ประกอบต่าง ๆได้ง่าย ผู้เรียนสามารถควบคุมการเรียนรู้โดยใช้องค์ประกอบต่าง ๆของโมเดลฯ ได้อย่างอิสระ และโมเดลฯช่วยส่งเสริมและสนับสนุนผู้เรียนให้เข้าใจโปรแกรมเรียกตัวเอง

(2) ด้านประโยชน์ที่ได้รับ พบว่า โมเดลฯเปิดโอกาสให้เกิดการปฏิสัมพันธ์(Interactive)ระหว่างผู้เรียนกับผู้สอนและผู้เรียนกับผู้เรียนหรือผู้เรียนกับเนื้อหาบทเรียน โมเดลฯสามารถนำเสนอเนื้อหาในรูปแบบของสื่อหลายมิติ ช่วยทำให้ผู้เรียนเกิดความเข้าใจได้ดี มีแหล่งการเรียนรู้บนเครือข่ายที่หลากหลาย รวมทั้งผู้เรียนสามารถเรียนตามความพร้อมความถนัดและความสนใจของตน โดยไม่มีข้อจำกัดทางสถานที่และเวลาของการเรียนรู้

2. ผลการศึกษาความเข้าใจโปรแกรมของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรม

การศึกษาความเข้าใจโปรแกรมของผู้เรียนในครั้งนี้ ผู้วิจัยทำการเก็บรวบรวมข้อมูลโดยวิธีการสัมภาษณ์ผู้เรียนซึ่งเป็นกลุ่มเป้าหมายในระยะที่ 2 จำนวน 12 คนและกลุ่มเป้าหมายในระยะที่ 3 จำนวน 20 คน และนำข้อมูลการสัมภาษณ์ที่ได้มาวิเคราะห์โปรโตคอลตามกรอบของ Pennington(1987) ที่อธิบายว่า ความเข้าใจโปรแกรมเป็นการสร้างสิ่งแทนความเข้าใจ(Mental representation) 2 รูปแบบได้แก่ การสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model) และการสร้างสิ่งแทนความเข้าใจสถานการณ์ (Situation model) โดย การสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model) จะพัฒนาขึ้นก่อนการสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model) เริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่คำสั่งโปรแกรม(Program code)ที่พบเห็น โดยความสนใจจะมุ่งไปที่คำ(Beacon)ที่คิดว่าสำคัญ โดยนำคำ(Beacon) นี้มาจับคู่ (Match)กับความรู้เดิมที่มีอยู่(Syntactic knowledge) จะสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างเล็ก(Microstructure) และเมื่อมีการรวมกลุ่ม(Chunk)ของคำสั่งจะสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดใหญ่(Macrostructure) การสร้างสิ่งแทนความเข้าใจสถานการณ์เริ่มเกิดขึ้นเมื่อผู้เรียนกระตุ้นความรู้แผนงาน (Plan knowledge)ที่เป็นความรู้เดิมที่มีอยู่มาจับคู่กับสิ่งแทนความเข้าใจคำสั่งที่สร้างขึ้นเพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans) ก่อรูปเป็นสิ่งแทนความเข้าใจย่อย(Subgoal) กระบวนการทำความเข้าใจ(Comprehension process) จะกระทำอย่างต่อเนื่องโดยการสร้างสิ่งแทนความเข้าใจคำสั่งสุดท้าย(Final mental representation) อยู่ในรูปแบบกระแสควบคุม(Control flow)หรือกระบวนการทำงานของคำสั่ง (Procedural) ส่วนการสร้างสิ่งแทนความเข้าใจสถานการณ์สุดท้าย(Final mental representation) จะเป็นภาพกระแสข้อมูล(Dataflow view)ที่สามารถสรุปหน้าที่ของโปรแกรมได้(Functional) ดังแสดงในภาพที่ 5.4



ภาพที่ 5.4 แสดง Pennington Comprehension Model

ผลการศึกษาความเข้าใจโปรแกรมของผู้เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรม ในระยะที่ 2 และระยะที่ 3 สามารถสรุปกระบวนการทำความเข้าใจโปรแกรมของผู้เรียนแบ่งเป็น 3 ลักษณะที่แตกต่างกัน ดังนี้

2.1 ลักษณะที่ 1 พบว่า ผู้เรียนสร้างความเข้าใจโปรแกรมแบบล่างขึ้นบนตามกรอบของ Pennington(1987) เริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่คำสั่งโปรแกรม (Program code) ที่ผู้เรียนเห็น ดังการสัมภาษณ์ที่ว่า "ไม่สามารถรู้ผลลัพธ์ของการทำงานโปรแกรมได้ เพราะไม่เห็น code เลย การทำความเข้าใจโปรแกรมต้องเห็น code ก่อนและดูว่ามีคำสั่งอะไรบ้าง เช่น คำสั่ง if คำสั่ง while หรือคำสั่ง for หรือเปล่าเพื่อจับทางการทำงานของโปรแกรม" ต่อจากนั้นผู้เรียนจะเริ่มสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model) โดยพิจารณาจากคำที่มีลักษณะเด่น ๆที่คิดว่ามีความสำคัญ(Beacon)ต่อความเข้าใจโปรแกรม ความเข้าใจโปรแกรมลักษณะที่ 1 นี้ กระบวนการทำความเข้าใจของผู้เรียนจะเริ่มที่คำว่า "main" หรือคำแรกที่ผู้เรียนเห็นเป็นคำว่า "include" หรือชื่อฟังก์ชันที่ผู้เรียนสนใจ ดังการสัมภาษณ์ที่ว่า "การทำความเข้าใจโปรแกรมนี้ไม่จำเป็นต้องดูตั้งแต่บรรทัดแรกแต่ดูในส่วนที่สำคัญ ดูที่คำว่า main เพราะจะได้รู้ลำดับการทำงานของโปรแกรมว่ามีข้อมูลนำเข้าอะไร เรียกฟังก์ชันโดยส่งผ่านอะไรและแสดงผลอย่างไรซึ่งภายในฟังก์ชัน main จะประกอบด้วยคำสั่งในการทำงานหลายคำสั่ง" หรือ การสัมภาษณ์ที่ว่า "ก่อนอื่นต้องดูที่บรรทัดแรกก่อน #include <iostream> ที่เห็น" ต่อจากนั้นผู้เรียนนำคำว่า "main" หรือคำว่า "include" หรือชื่อฟังก์ชันที่ผู้เรียนสนใจมาจับคู่

(Match)กับความรู้เดิมที่มีอยู่เป็นความรู้รูปแบบไวยากรณ์(Syntactic knowledge)สร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างเล็ก(Microstructure) ดังการสัมภาษณ์ที่ว่า "ในบรรทัดนี้ประกอบด้วยคำสั่ง int main() เป็นหัวของฟังก์ชัน ทำให้ทราบว่าฟังก์ชันนี้ จะส่งผ่านค่าเป็นเลขจำนวนเต็มหรือ int กลับหลังจากจบการทำงาน" และ การสัมภาษณ์ที่ว่า

คำสั่ง #include<iostream> เป็นคำสั่งภาษา C++ หนึ่งคำสั่ง เป็นส่วนของโปรแกรมที่เรียกว่า compiler directive ที่เป็นการนำไลบรารี่ iostream เข้ามารวมในโปรแกรม ทำให้โปรแกรมนี้สามารถใช้คำสั่ง cin , cout ในโปรแกรมได้เป็นการรับข้อมูลทางแป้นพิมพ์และแสดงผลออกทางจอภาพ

ผู้เรียนสามารถสร้างสิ่งแทนความเข้าใจคำสั่งเหล่านี้ได้ ต้องมีความรู้ในเรื่องของรูปแบบคำสั่ง(Syntactic knowledge) ดังการสัมภาษณ์ที่ว่า

การรู้ว่าคำสั่งแต่ละคำสั่งทำอย่างไร ก่อนอื่นต้องดูรูปแบบของคำสั่งก่อนว่า เป็นรูปแบบคำสั่งใด เช่นคำสั่งคำนวณ คำสั่งเงื่อนไข หรือคำสั่งการทำงานวนรอบ ซึ่งรูปแบบเหล่านี้ ต้องรู้พอมองเห็นก็ต้องสามารถรู้ได้เลยว่าเป็นคำสั่งลักษณะไหน มันจะเป็นความรู้ที่ทุกคนต้องมีเป็นความรู้พื้นฐาน จะทำความเข้าใจโปรแกรมภาษาใด ก็ต้องรู้รูปแบบคำสั่ง ของคำสั่งภาษานั้น

ต่อจากนั้นผู้เรียนจะทำการพิจารณาคำสั่งต่อ ๆไปตามลำดับเพื่อทำความเข้าใจรูปแบบของทุกคำสั่งทุกบรรทัดในโปรแกรม ผู้เรียนทุกคนสนใจคำสั่งที่อยู่ภายในฟังก์ชัน main ทุกคน ดังการสัมภาษณ์ที่ว่า

บรรทัดที่สนใจเป็น int main() เพราะเป็นจุดเริ่มต้นของโปรแกรม จะประกอบไปด้วยกลุ่มของคำสั่งหลาย ๆคำสั่งที่สำคัญ ส่วนของคำสั่ง input ส่วนของคำสั่งประมวลผล ส่วนของคำสั่งในการแสดงผลลัพธ์ออกทางจอภาพ จะอยู่ที่ฟังก์ชัน main ทั้งหมด ทำให้เราสามารถรู้ได้ว่าโปรแกรมนี้มีวัตถุประสงค์ทำอะไร

และ การสัมภาษณ์ที่ว่า

คำสั่งแรกที่อยู่ในฟังก์ชัน main คือ int x; *คำสั่งนี้เขียนในรูปแบบของการประกาศตัวแปรที่ใช้ในโปรแกรม ประกอบด้วยคำที่แสดงถึงชนิดข้อมูล ตัวแปร และเครื่องหมายเซมิโคลอน ความรู้ในเรื่องของการประกาศตัวแปร เป็นความรู้ที่ต้องมีมาก่อน ผมเรียนจากอาจารย์ในห้องและยังจำได้ดีเพราะทุกโปรแกรมต้องมีการประกาศตัวแปร*

ผู้เรียนจะทำความเข้าใจรูปแบบของคำสั่งโดยมีการรวมกลุ่ม(Chunk)ของคำสั่งมากขึ้น เช่น กลุ่มของคำสั่งที่เป็นโครงสร้างต่าง ๆ หรือกลุ่มของการประกาศตัวแปรที่มีหลาย ๆคำสั่งที่ปรากฏในเอกสารโปรแกรม ผู้เรียนจะสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดใหญ่ (Macrostructure) ดังการสัมภาษณ์ ที่ว่า

ต่อจากนั้นก็พิจารณาคำสั่งต่อ ๆไป บรรทัดต่อไปเป็นคำสั่งในการประกาศตัวแปรเหมือนกัน แต่เป็นการประกาศตัวแปรที่เป็นข้อความ คำสั่งต่อไปเป็นเป็นคำสั่งในการรับข้อมูลทางแป้นพิมพ์ เพราะคำสั่งนี้เขียนตามรูปแบบกระบวนการการรับข้อมูลทางแป้นพิมพ์

ประกอบด้วยคำว่า cin>>x; ต่อจากนั้นก็เป็นคำสั่งเรียกใช้ฟังก์ชัน compute คำสั่งต่อไปเป็นคำสั่งในการพิมพ์ผลลัพธ์ออกทางจอภาพ

และการสัมภาษณ์ที่ว่า

คำสั่งภายในฟังก์ชัน main นั้นก็เริ่มพิจารณาการทำงานใน { } โดยเริ่มจากบรรทัดแรกเป็นการประกาศตัวแปร x เป็นอาเรย์หนึ่งมิติขนาด 10 บรรทัดที่ 2 เป็นการประกาศ ตัวแปร sum ให้มีค่าเท่ากับ 0 บรรทัดที่ 3 เป็นการประกาศตัวแปร n เป็นเลขจำนวน เต็ม ให้มีค่าเริ่มต้นเท่ากับ 0

การสร้างสิ่งแทนความเข้าใจสถานการณ์เริ่มเกิดขึ้นเมื่อผู้เรียนกระตุ้นความรู้แผนงาน (Plan knowledge)หรือแนวคิดในการเขียนโปรแกรม(Programming concepts) ที่เป็นความรู้เดิมที่มีอยู่มาจับคู่กับสิ่งแทนความเข้าใจคำสั่งที่สร้างขึ้น(Micro structure) เป็นความเข้าใจกระบวนการทำงานของคำสั่ง (Procedure) เพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans) ก่อรูปเป็นสิ่งแทนความเข้าใจย่อย(Subgoal) ดังการสัมภาษณ์ที่ว่า

จะพิจารณาคำสั่งที่อยู่ภายในฟังก์ชัน main เพราะจุดเริ่มต้นและจุดสิ้นสุดจะกระทำที่ฟังก์ชัน main ภายในฟังก์ชัน main จะมีคำสั่งในการรับข้อมูล คำสั่งในการคำนวณ และ คำสั่งในการแสดงผล ซึ่งเป็นแนวคิดการเขียนโปรแกรมพื้นฐานที่มีอยู่ทุกโปรแกรม ต่อจากนั้นจะทำความเข้าใจคำสั่งทีละคำสั่งต่อเนื่องกันไป คำสั่งแรกจะกำหนดให้ตัวแปร x เก็บข้อมูลเป็นเลขจำนวนเต็มใด ๆ คำสั่งที่สองกำหนดให้ตัวแปร s เก็บข้อมูลที่เป็นข้อความ คำสั่งต่อไปเป็นการแสดงข้อความ x= ออกทางจอภาพ คำสั่งต่อไปเป็นคำสั่งให้ผู้ใช้ป้อนเลขจำนวนเต็มใด ๆทางแป้นพิมพ์ โปรแกรมจะนำตัวเลขที่ผู้ใช้ป้อนเก็บในตัวแปร x ที่กำหนดไว้ตั้งแต่แรก ต่อจากนั้นเรียกใช้ฟังก์ชัน compute

กระบวนการทำความเข้าใจ(Comprehension process)จะกระทำอย่างต่อเนื่องโดยการสร้างสิ่งแทนความเข้าใจคำสั่งสุดท้าย(Final mental representation)อยู่ในรูปของทิศทางหรือกระแสควบคุม(Control flow)ที่แสดงกระบวนการทำงานของคำสั่ง(Procedural) ดังการสัมภาษณ์ที่ว่า

รู้ว่าคำสั่ง if มีทิศทางหรือกระแสการทำงานดังนี้ การทำงานคำสั่งใด ๆที่อยู่ในกลุ่มของคำสั่ง ifนั้นขึ้นกับเงื่อนไข ถ้าเงื่อนไขเป็นจริงจะหยุดการทำงานโดยส่งผ่านค่าข้อความว่าง ๆกลับไปยังจุดเรียกใช้ แต่ถ้าเป็นเท็จจะเรียกตัวเองโดยเปลี่ยนแปลงค่าเพื่อกระทำในรอบต่อไป เป็นการเลือกการทำงานที่จะต้องกระทำ

ในขณะที่ผู้เรียนสร้างสิ่งแทนความเข้าใจโปรแกรม(Program model) ผู้เรียนจะสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ไปด้วยจากภาพกระแสของข้อมูล(Dataflow view) ที่มีการเปลี่ยนแปลงไปตามกระบวนการทำงานของคำสั่งในโปรแกรม ดังการสัมภาษณ์ที่ว่า

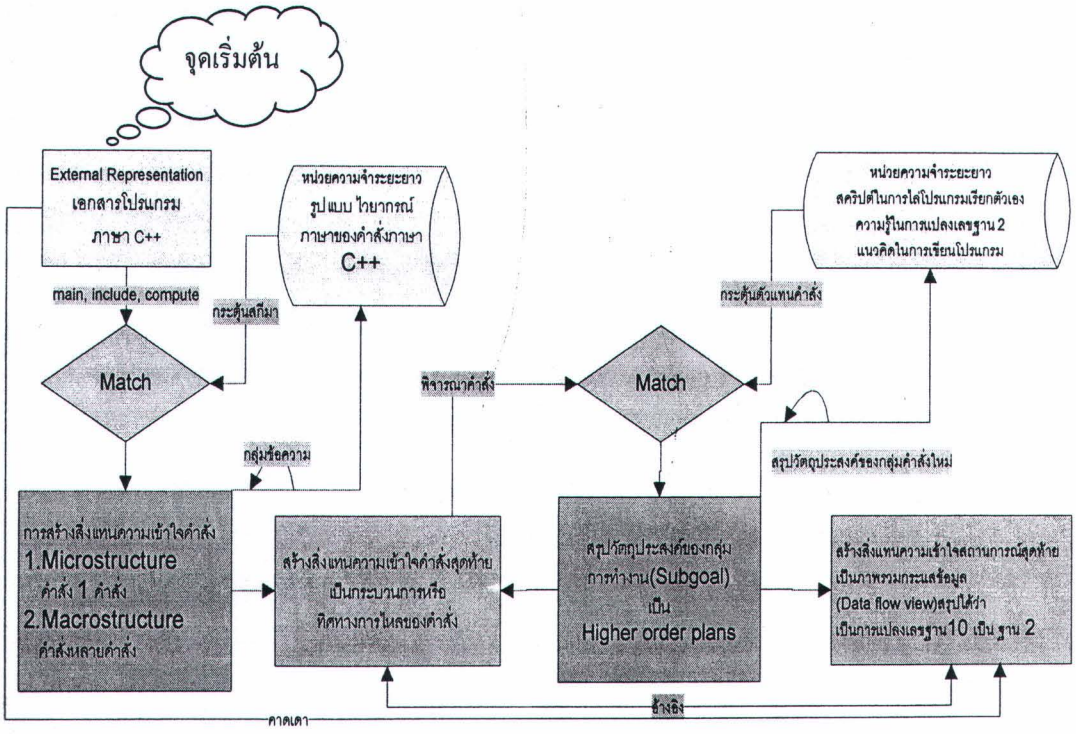
รับค่า x เป็นตัวเลขจำนวนเต็ม และนำค่า x กับ s ส่งไปคำนวณที่ฟังก์ชัน compute การทำงานในฟังก์ชัน compute สมมุติ y=4 จะเปรียบเทียบในเงื่อนไขแรก ถ้าไม่ใช่ 0 จะมาทำงานในคำสั่ง if ที่ 2 ก็เอา 4 ไป mod กับ 2 ว่าเท่ากับ 0 ไหม ถ้ามันเท่ากับ 0 จริง ค่า s ก็จะนำเอา 0 + s ตัวใหม่คือ 0 นั่นเองเป็น 00 แล้ว ต่อจากนั้นทำการเรียกฟังก์ชัน compute ใหม่โดยเอาค่า 4 มา

หารด้วย 2 มีค่าเท่ากับ 2 แล้วส่ง 2 ไปเทียบค่าใหม่ ก็คือ 2 เท่ากับ 0 ไหมในกรณีนี้ จะทำ if ที่สอง ก็คือ เอา 2 มา mod กับ 2 แล้วตรวจสอบว่าเท่ากับ 2 ไหม ในกรณีนี้มีค่าเท่ากับ 0 ตามเงื่อนไขก็จะเริ่มนำเอา 0+s เดิม เป็นการต่อสตริง และเรียกตัวเองอีกครั้งหนึ่ง ก็คือ 2/2 เท่ากับ 1 แล้ว ก็จะไปตรวจสอบเงื่อนไข และตรวจสอบว่า mod 2 เท่ากับ 0 ไหมในกรณีนี้ไม่เท่าจึงทำที่คำสั่ง else จะนำ 1 มาต่อกับค่า s เดิม และก็เรียกตัวเองใหม่อีก1/2 เท่ากับ 0.5

ผู้เรียนบางกลุ่มจะทำการกำหนดค่าให้กับตัวแปร x เพื่อทำการติดตามกระบวนการทำงานของคำสั่งในโปรแกรมเพื่อหาผลลัพธ์จากการทำงาน ดังคำสัมภาษณ์ที่ว่า

กระบวนการทำงานของโปรแกรมนั้นต้องทดลองกำหนดข้อมูลใด ๆสมมุติเป็น 3 ให้ ตัวแปร y โดยตัวแปร s เป็นผลตอนแรกเป็นค่าว่าง ต่อจากนั้นนำค่า 3 ไปเปรียบเทียบเพื่อเลือกการทำงาน ปรากฏว่า ต้องกระทำซ้ำ โดยมีการเรียกฟังก์ชันใหม่ ในรอบต่อไป ซึ่ง y ในรอบใหม่มีค่าเท่ากับ 1 ก็จะเรียกใหม่อีก ในรอบสุดท้ายนี้ y จะมีค่าเป็น 0 จะหยุดการทำงานส่งค่าว่าง ๆกลับทำให้ผลของ s มีค่าเท่ากับ 11

ผู้เรียนสร้างสิ่งแทนความเข้าใจสถานการณ์สุดท้าย(Final mental representation) จากภาพกระแสข้อมูล(Dataflow view)ที่สามารถสรุปหน้าที่ของโปรแกรมได้(Functional) ซึ่งผู้เรียนต้องมีความรู้ที่เกี่ยวกับปัญหา(Domain knowledge)มาก่อน ดังการสัมภาษณ์ที่ว่า "การสรุปว่าเป็นการแปลงเลขฐานสิบเป็นเลขฐานสอง บางคนคิดได้ บางคนคิดไม่ได้ ต้องมีความรู้ในการแปลงเลขฐาน2มาก่อน เช่น เลข 5 ต้องเป็นเลขอะไรกับอะไร 0 อยู่หลักไหน 1 อยู่หลักไหน ถ้าไม่รู้เรื่องการแปลงเลขฐานไม่สามารถรู้ได้" สรุปดังแสดงในภาพที่ 5.5



ภาพที่ 5.5 แสดงกระบวนการทำความเข้าใจโปรแกรมของผู้เรียน ลักษณะที่ 1

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

2.2 ลักษณะที่ 2 พบว่า ผู้เรียนสร้างความเข้าใจโปรแกรมที่แตกต่างจากกรอบของ Pennington (1987) โดยเริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่คำสั่งโปรแกรม (Program code) ที่ผู้เรียนเห็น ดังการสัมภาษณ์ที่ว่า "เมื่อพิจารณาโปรแกรมที่ให้ ดูแบบผ่าน ๆ ตาน่าจะเป็นการหาเลข binary" ต่อจากนั้นผู้เรียนจะเริ่มสร้างสิ่งแทนความเข้าใจสถานการณ์ (Situation model)ก่อน โดยพิจารณาจากคำสั่งที่มีลักษณะเด่น ๆที่คิดว่ามีความสำคัญ(Beacon) คือ s='0'+s และ s='1'+s นำไปเชื่อมโยง(Match) กับความรู้เดิมที่มีอยู่เป็นประสบการณ์กับปัญหา(Domain knowledge)และคาดเดาถึงวัตถุประสงค์ของกระบวนการทำงานของโปรแกรม (Functional) ดังการสัมภาษณ์ที่ว่า

คิดว่าโปรแกรมนี้น่าจะเป็นการหาเลข binary เพราะเห็นว่ามีคำสั่ง s='1'+s และ s = '0'+s มันสะดุดตา จำได้ว่าอาจารย์เคยสอนการแปลงเลขฐานมีการเชื่อมต่อตัวเลขเฉพาะ 0 หรือ 1 แต่ไม่ได้เขียนวิธีการเรียกตัวเอง โปรแกรมนี้มีการเชื่อมต่อสตริงที่เป็นตัวเลขคิดว่า การแปลงเลขฐาน อาจเขียนเป็นแบบเรียกตัวเองได้ เพราะมีการทำงานวนรอบเหมือนกัน

ผู้เรียนจะเริ่มตรวจสอบความถูกต้องของสมมุติฐาน(Hypothesis) ที่ได้คาดเดาไว้โดยเริ่มพิจารณารูปแบบคำสั่งโปรแกรมอย่างจริงจัง ดังคำสัมภาษณ์ที่ว่า

เมื่อเริ่มพิจารณาคำสั่งภายในโปรแกรมจะเริ่มจาก main เป็นอันดับแรกเพื่อดูว่ามีตัวแปรหรือฟังก์ชันอะไรบ้าง โปรแกรมนี้มีฟังก์ชันเดียวชื่อ compute มีคำสั่งคำนวณหาค่า ที่รู้เพราะ อาจารย์สอนในชั้นเรียน เช่น การกำหนดตัวแปร การสร้างฟังก์ชัน การคำนวณ เป็นรูปแบบที่กำหนดตายตัวเห็นคำสั่งก็ทราบได้ทันที

ผู้เรียนจะนำข้อความที่สนใจ(Beacon)นำมาจับคู่(Match)กับความรู้รูปแบบคำสั่ง(Syntactic knowledge) สร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างเล็ก(Microstructure) ต่อจากนั้นผู้เรียนจะดูดซึมความเข้าใจรูปแบบของคำสั่งโดยมีการรวมกลุ่ม(Chunk)ของคำสั่งมากขึ้น เช่นกลุ่มของคำสั่งที่เป็นโครงสร้างหรือกลุ่มของการประกาศตัวแปรที่มีหลาย ๆคำสั่งที่ปรากฏในเอกสารโปรแกรม ผู้เรียนจะสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดใหญ่(Macrostructure) ดังการสัมภาษณ์ที่ว่า "คำสั่งในบรรทัดที่ 13 คือ int x; และบรรทัดที่ 14 คือ string s; เป็นการประกาศตัวแปร โปรแกรมนี้มีการกำหนดตัวแปร 2 ตัวคือ x และ s โดย x เป็นตัวแปรที่เก็บเลขจำนวนเต็มใด ๆส่วนตัวแปร s เป็นตัวแปรสตริงที่เก็บข้อความ" การสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model) เริ่มเกิดขึ้นเมื่อผู้เรียนกระตุ้นความรู้แผนงาน (Plan knowledge) หรือแนวคิดในการเขียนโปรแกรม(Programming concepts)หรือความรู้ที่เป็นปัญหา(Problem domain) ที่เป็นความรู้เดิมที่ผู้เรียนมีอยู่มาจับคู่(Match)กับสิ่งแทนความเข้าใจรูปแบบคำสั่งที่สร้างขึ้นกลายเป็นความเข้าใจกระบวนการทำงานของคำสั่ง(Procedure) เพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans)ก่อรูปเป็นสิ่งแทนความเข้าใจย่อย(Subgoal) ดังการสัมภาษณ์ที่ว่า

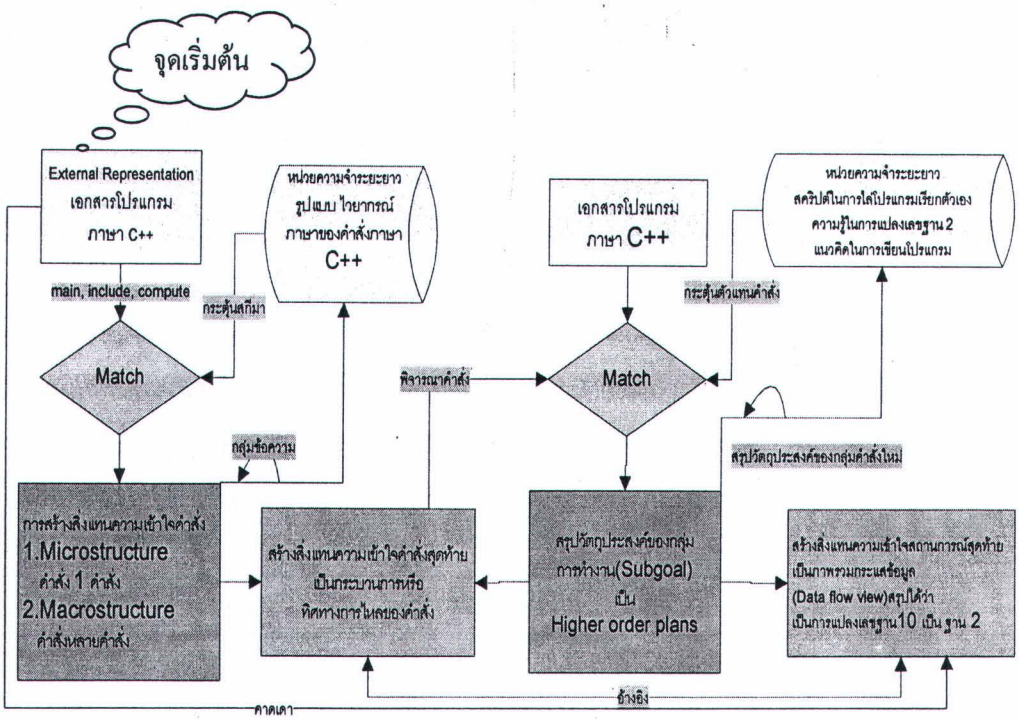
ความรู้ที่ต้องมีในการทำความเข้าใจโปรแกรม ในความคิดของผมที่สำคัญคือรู้กระบวนการแปลงเลขฐานต้องทราบอัลกอริทึมในการแปลง รู้คำสั่งภาษา C++ที่นำมาใช้ในการ

แปลงเลขฐาน คือต้องคิดให้ได้ก่อนว่าทำอย่างไงแล้วก็แปลงจากสิ่งที่คิดเป็นคำสั่งที่เขียนแบบเรียกตัวเอง

กระบวนการสร้างความเข้าใจโปรแกรมของผู้เรียนจะกระทำอย่างต่อเนื่องจากกระบวนการทำงานของคำสั่งต่าง ๆที่มีการเปลี่ยนแปลงไป ผู้เรียนจะใช้ความคิดอย่างมากในการติดตามการทำงานของกระแสข้อมูล(Dataflow view) ดังคำสัมภาษณ์ที่ว่า

การทำความเข้าใจโปรแกรมต้องใช้ความคิดอย่างมากเพราะต้องใช้สมองในการจำค่าของตัวแปรต่าง ๆที่มีอยู่ในโปรแกรมติดตามการเปลี่ยนแปลงที่เกิดขึ้นในจากการทำงานของแต่ละคำสั่งที่ส่งผลให้ตัวแปรเปลี่ยนแปลงไปตั้งแต่คำสั่งแรกจนกระทั่งจบโปรแกรมจึงจะสามารถสรุปได้ว่าโปรแกรมที่กำลังพิจารณาอยู่นี่ทำอะไร กระบวนการทำงานของโปรแกรมนั้นต้องทดลองกำหนดข้อมูลใด ๆสมมุติเป็น 3 ให้ตัวแปร y โดยตัวแปร s เป็นผลตอนแรกเป็นค่าว่าง ต่อจากนั้นนำค่า 3 ไปเปรียบเทียบเพื่อเลือกการทำงาน ปรากฏว่า ต้องกระทำซ้ำ โดยมีการเรียกฟังก์ชันใหม่ ในรอบต่อไป ซึ่ง y ในรอบใหม่มีค่าเท่ากับ 1 ก็จะเรียกใหม่อีก ในรอบสุดท้ายนี้ y จะมีค่าเป็น 0 จะหยุดการทำงานส่งค่าว่าง ๆกลับ ทำให้ผลของ s มีค่าเท่ากับ 11

ผู้เรียนสร้างสิ่งแทนความเข้าสถานการณ์ของกระบวนการทำงานของรูปแบบคำสั่งต่าง ๆโดยสรุปเป็นวัตถุประสงค์การทำงานย่อย(Subgoal) กระบวนการทำความเข้าใจจะมีการทำงานอย่างต่อเนื่องจนกระทั่งผู้เรียนสามารถสร้างสิ่งแทนความเข้าใจสถานการณ์สุดท้าย (Final mental representation) จากภาพกระแสข้อมูล(Dataflow view)ทั้งหมดของโปรแกรม และสรุปหน้าที่ของโปรแกรมได้(Functional) ดังการสัมภาษณ์ที่ว่า"สรุปการทำงานของโปรแกรมนี้เป็นการแปลงเลข 3 ให้เป็น 11 คือแปลงเลข 3 ฐานสิบ ให้เป็น 11 ฐานสอง" สรุปดังแสดงในภาพที่ 5.6



ภาพที่ 5.6 กระบวนการทำความเข้าใจโปรแกรมของผู้เรียน ลักษณะที่ 2

2.3 ลักษณะที่ 3 พบว่า ผู้เรียนสร้างความเข้าใจโปรแกรมที่แตกต่างจากกรอบของ Pennington (1987) โดยเริ่มจากสิ่งแทนภายนอก(External representation)ได้แก่คำสั่งโปรแกรม (Program code) ที่ผู้เรียนเห็น ดังการสัมภาษณ์ที่ว่า "ดูโปรแกรมที่อาจารย์ให้แบบผ่าน ๆตาคิดว่าเกี่ยวกับการแปลงเลขฐาน น่าจะเกี่ยวกับเลขฐาน 2" ต่อจากนั้นผู้เรียนจะเริ่มสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model)ก่อน โดยพิจารณาจากคำสั่งที่มีลักษณะเด่น ๆที่คิดว่ามีความสำคัญ(Beacon) คือ "if (x%2==0) s='0'+s; else s='1'+s;" และคาดเดาถึงวัตถุประสงค์ของกระบวนการทำงานทั้งหมด(Functional) ดังการสัมภาษณ์ที่ว่า "ไม่ทราบว่าโปรแกรมมีวัตถุประสงค์ทำอะไรจริง ๆ แต่คาดเดาว่าน่าจะเกี่ยวกับการแปลงเลขฐาน โดยดูจากคำสั่งที่มีการที่มีการหารเอาเศษลงตัวและไม่ลงตัว แล้วได้ผลเป็น 0 หรือ 1 มาต่อกันหน้าสตริง s ตัวเก่า if (x%2==0) s='0'+s; else s='1'+s;" การคาดเดาของผู้เรียนเกิดจากผู้เรียนนำความรู้ที่มีอยู่เดิม(Domain knowledge)ที่มีอยู่มาเปรียบเทียบ(Match)กับสิ่งแทนความเข้าใจโปรแกรม(Program model)ที่พบเห็นในเอกสารโปรแกรม(Program document) ดังการสัมภาษณ์ที่ว่า "ที่คาดเดาแบบนี้เพราะยังจำได้อยู่มีหลายวิชาที่สอนเกี่ยวกับการแปลงเลขฐาน และต้องรู้คำสั่งโปรแกรมภาษา C++ ด้วยจึงจะทราบ" และการสัมภาษณ์ที่ว่า

ดูจากโปรแกรมแล้วเป็นโปรแกรมที่มีขนาดเล็ก มีเพียงฟังก์ชัน main และ compute เท่านั้น ส่วนที่สำคัญคือการเข้าใจกระบวนการทำงานของคำสั่ง compute เท่านั้นก็สามารถทราบได้ว่าโปรแกรมทำอะไร เพราะกระบวนการทำงานของโปรแกรมไม่มีอะไรที่ซับซ้อน เป็นคำสั่งพื้นฐานธรรมดา

ผู้เรียนจะเริ่มตรวจสอบความถูกต้องของสมมุติฐาน(Hypothesis) ที่ได้คาดเดาไว้โดยเริ่มพิจารณารูปแบบคำสั่งโปรแกรมอย่างจริงจังเพื่อให้แน่ใจว่าสมมุติฐานที่คาดเดามีความถูกต้องจริง ดังคำสัมภาษณ์ที่ว่า

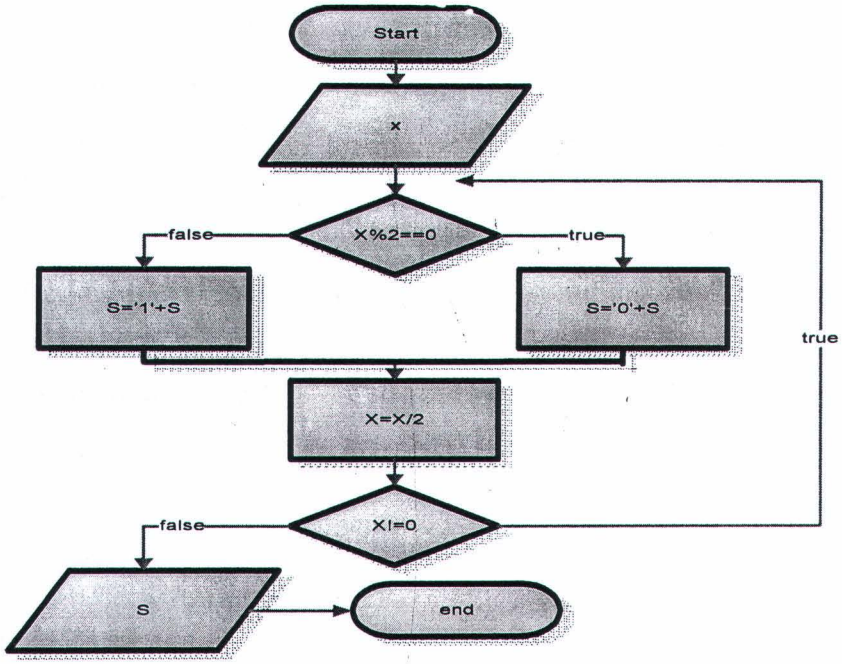
เริ่มพิจารณาในส่วนของฟังก์ชันคือ compute เพราะบอกถึงวัตถุประสงค์ของโปรแกรม โปรแกรมนี้มีแค่ฟังก์ชันเดียว ในส่วนของ main ไม่มีอะไรมีแต่รับข้อมูล เรียกฟังก์ชันและแสดงผลเท่านั้นไม่มีอะไรซับซ้อนแต่ส่วนที่ต้องวิเคราะห์หรือทำความเข้าใจ คือ ฟังก์ชัน compute สำคัญที่สุด

ผู้เรียนจะนำข้อความที่อยู่ภายในฟังก์ชัน computeนำมาจับคู่(Match)กับความรู้รูปแบบคำสั่ง(Syntactic knowledge)ที่ผู้เรียนมีอยู่ สร้างเป็นสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างเล็ก (Microstructure) ต่อจากนั้นผู้เรียนจะทำความเข้าใจรูปแบบคำสั่งโดยมีการรวมกลุ่ม (Chunk)ของรูปแบบคำสั่งมากขึ้นสร้างสิ่งแทนความเข้าใจคำสั่งเป็นโครงสร้างขนาดใหญ่ (Macrostructure) เช่นโครงสร้างของคำสั่งเงื่อนไข if ดังการสัมภาษณ์ที่ว่า

เข้าใจรูปแบบและการทำงานของคำสั่งที่อยู่ภายในฟังก์ชัน compute คำสั่ง if นี้จะเป็นทางเลือกการทำงาน เป็นการเขียนโปรแกรมแบบเรียกตัวเองตามรูปแบบของการเขียนฟังก์ชันเรียกตัวเอง ทั้งสามบรรทัดเป็นเพียง 1 คำสั่ง รูปแบบทั่วไปคือ if(เงื่อนไข) คำสั่งที่กระทำเมื่อเงื่อนไขเป็นจริง else คำสั่งที่กระทำเมื่อเงื่อนไขเป็นเท็จ

การสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model) เริ่มเกิดขึ้นเมื่อผู้เรียนกระตุ้นความรู้แผนงาน (Plan knowledge)หรือสคริปต์การทำงาน(Script) ที่เป็นความรู้เดิมที่ผู้เรียนมีอยู่มาจับคู่(Match) กับสิ่งแทนความเข้าใจรูปแบบคำสั่งที่สร้างขึ้น เพื่อสร้างแผนงานในระดับที่สูงขึ้น(Higher order plans) ก่อรูปเป็นสิ่งแทนความเข้าใจย่อย(Subgoal) กลายเป็นความเข้าใจกระบวนการทำงานของคำสั่ง (Procedure) ในรูปของกระแสควบคุม(Control flow) หรือผังโปรแกรม ดังการสัมภาษณ์ที่ว่า

การเขียนโปรแกรมแบบเรียกตัวเอง ในที่นี้คือการทำงานของฟังก์ชัน compute การส่งผ่านค่าเรียกตัวเอง compute(y/2,s) นอกจากนี้ต้องรู้การคำนวณ div และ mod ที่ส่งผลให้การส่งผ่านค่าในรอบต่อไปที่มีการเรียกตัวเองเปลี่ยนแปลงไปเรื่อย ๆ จนกระทั่งส่งผลให้การทำงานในส่วนของเงื่อนไขหยุดการเรียกตัวเอง ตามผังโปรแกรมดังนี้



ภาพที่ 5.7 แสดงการสร้างสิ่งแทนความรู้ในรูปผังโปรแกรม

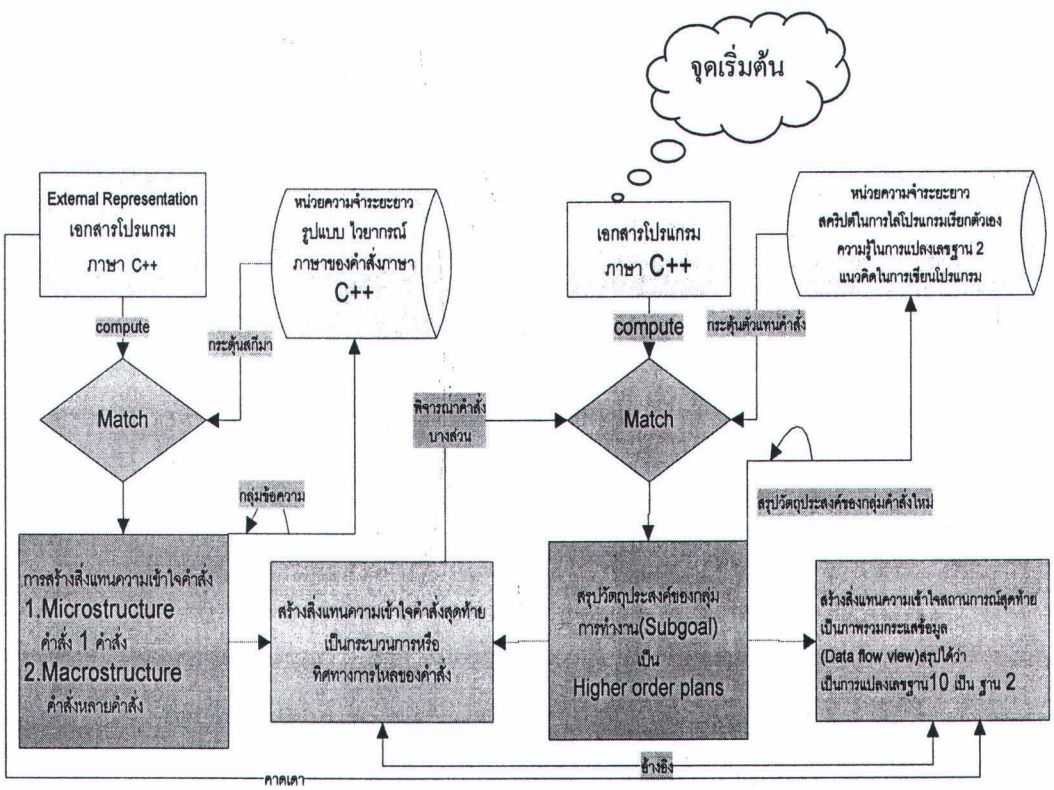
กระบวนการสร้างความเข้าใจโปรแกรมของผู้เรียนจะกระทำอย่างต่อเนื่องจากกระบวนการทำงานของคำสั่งต่าง ๆที่มีการเปลี่ยนแปลงไป ผู้เรียนจะใช้ความคิดอย่างมากในการติดตามการทำงานของกระแสข้อมูล(Dataflow view) ดังคำสัมภาษณ์ที่ว่า

ส่งค่า x เข้าไปในฟังก์ชัน จะนำค่า x ไปหาร 2 ถ้าหารลงตัวจะนำค่า 0 ไปต่อหน้าสตริงค์ s แต่ถ้าหารไม่ลงตัวจะนำค่า 1 ต่อหน้าสตริง s แล้วนำค่า x/2 เป็นตัวตั้งต้นในการเรียกตัวเอง การเรียกตัวเองเป็นการทำฟังก์ชัน compute ซ้ำ ทำจนกว่าค่า x==0 จึงหยุดการทำงานเรียกตัวเอง

ผู้เรียนสร้างสิ่งแทนความเข้าสถานการณ์ของกระบวนการทำงานของรูปแบบคำสั่งต่าง ๆโดยสรุปเป็นวัตถุประสงค์การทำงานย่อย(Subgoal) และสามารถสร้างสิ่งแทนความเข้าใจสถานการณ์สุดท้าย (Final mental representation) จากภาพกระแสข้อมูล(Dataflow view)เพียงบางส่วนของโปรแกรม และสรุปหน้าที่ของโปรแกรมได้(Functional) ผู้เรียนเพียงแต่ติดตามกระบวนการทำงานของกระแสข้อมูลเฉพาะภายในกลุ่มคำสั่งที่สนใจเท่านั้น ดังการสัมภาษณ์ที่ว่า

จากการติดตามการทำงานของคำสั่งในโปรแกรมและติดตามการทำงานของตัวแปร x ในการทำงานเริ่มต้น และกระทำซ้ำเมื่อมีการเรียกตัวเอง มีการลดค่าลงเรื่อย ๆ จนกระทั่งค่าของ x มีค่าเท่ากับ 0 จึงหยุดการทำงานทำให้สรุปได้ว่า ฟังก์ชัน compute เป็นฟังก์ชันในการแปลงเลขฐานสิบเป็นเลขฐานสอง จากการวิเคราะห์ผลของตัวแปร s ที่มีการเปลี่ยนแปลงในแต่ละรอบจนกระทั่งได้ผลลัพธ์เป็นค่าสุดท้าย ซึ่งผลที่ได้ตรง กับสมมุติฐานที่ได้คาดเดาไว้ตั้งแต่แรก เป็นอันแน่ใจได้ว่าโปรแกรมนี้เป็นโปรแกรมที่มีวัตถุประสงค์ในการแปลงเลขฐานสิบเป็นเลขฐานสองอย่างแน่นอน

กระบวนการทำความเข้าใจโปรแกรมของผู้เรียน สรุปดังแสดงในภาพที่ 5.8



ภาพที่ 5.8 กระบวนการทำความเข้าใจโปรแกรมของผู้เรียน ลักษณะที่ 3

3. ผลการศึกษาความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรม

การศึกษาความคิดเห็นของผู้เรียนเกี่ยวกับโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายฯ จากแบบสำรวจความคิดเห็นที่ผู้วิจัยสร้างขึ้น ซึ่งเปิดโอกาสให้ผู้เรียนได้เขียนแสดงความคิดเห็น รวมทั้งผลการสัมภาษณ์ผู้เรียน ซึ่งการศึกษาความคิดเห็นดังกล่าวจะพิจารณาใน 2 ด้าน คือ ด้านการใช้โมเดลฯ และ ด้านประโยชน์ที่ได้รับ สรุปผลได้ดังนี้

3.1 ด้านการใช้โมเดลฯ พบว่า ผู้เรียนสามารถเลือกใช้องค์ประกอบต่าง ๆได้ง่าย ผู้เรียนสามารถควบคุมการเรียนรู้โดยใช้องค์ประกอบต่าง ๆของโมเดลฯได้อย่างอิสระ และช่วยส่งเสริมและสนับสนุนผู้เรียนให้เข้าใจโปรแกรมเรียกตัวเอง

3.2 ด้านประโยชน์ที่ได้รับ พบว่า โมเดลฯเปิดโอกาสให้เกิดการปฏิสัมพันธ์(Interactive) ระหว่างผู้เรียนกับผู้สอนและผู้เรียนกับผู้เรียนหรือผู้เรียนกับเนื้อหาบทเรียน โมเดลฯสามารถนำเสนอเนื้อหาในรูปแบบของสื่อหลายมิติ ช่วยทำให้ผู้เรียนเกิดความเข้าใจได้ดี มีแหล่งการเรียนรู้บนเครือข่ายที่หลากหลาย รวมทั้งผู้เรียนสามารถเรียนตามความพร้อมความถนัดและความสนใจของตน โดยไม่มีข้อจำกัดทางสถานที่และเวลาของการเรียนรู้

4. ผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรม

ผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรม ในระยะที่ 2 และระยะที่ 3 ผู้วิจัยได้ทำการทดสอบก่อนและหลังเรียน จากการเรียนด้วยสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม ในรายวิชา CT212 เรื่องการเขียนโปรแกรมเรียกตัวเอง โดยใช้ข้อสอบอัตนัยจำนวน 10 ข้อ ๆละ 3 คะแนน รวม 30 คะแนน ด้วยแบบทดสอบชุดเดียวกัน สรุปผลดังแสดงในตารางที่ 5.4

ตารางที่ 5.4 แสดงค่าคะแนนคะแนนเฉลี่ย และส่วนเบี่ยงเบนมาตรฐาน ในระยะที่ 2 และระยะที่ 3

ระยะที่	$\overline{X}$ ก่อนเรียน	$\overline{X}$ หลังเรียน	S.D.ก่อนเรียน	S.D.หลังเรียน
2	10.00	21.25	1.57	1.25
3	11.35	21.78	1.09	1.47

ผลการวิจัยพบว่า ผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรม ในระยะที่ 2 คะแนนเฉลี่ยการทดสอบหลังเรียนเท่ากับ 21.25 ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.25 สูงกว่าคะแนนเฉลี่ยการทดสอบก่อนเรียน ซึ่งค่าคะแนน

เฉลี่ยการทดสอบก่อนเรียนเท่ากับ 10.00 ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.57 ผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรมในระยะที่ 3 คะแนนเฉลี่ยการทดสอบหลังเรียนเท่ากับ 21.78 ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.47 สูงกว่าคะแนนเฉลี่ยการทดสอบก่อนเรียน ซึ่งค่าคะแนนเฉลี่ยการทดสอบก่อนเรียนเท่ากับ 11.35 ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.09 นอกจากนี้ยังพบว่าผู้เรียนมีผลสัมฤทธิ์ทางการเรียนคิดเป็นร้อยละ 70 สูงกว่าเกณฑ์ที่กำหนดทั้งในระยะที่ 2 และระยะที่ 3 ดังนั้นผู้เรียนมีผลสัมฤทธิ์ทางการเรียน เรื่องการเขียนโปรแกรมเรียกตัวเองสูงขึ้น

5. สรุปและอภิปรายผลการวิจัย

จากการวิจัยดังกล่าวข้างต้น สามารถสรุปและอภิปรายผลได้ดังนี้

5.1 การออกแบบและพัฒนาโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม

จากผลการศึกษาเกี่ยวกับการออกแบบและพัฒนาโมเดลสิ่งแวดล้อมทางการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม พบว่าโมเดลฯมีองค์ประกอบที่สำคัญ 7 องค์ประกอบคือ สถานการณ์ปัญหา(Problem base) ธนาคารความรู้(Data banks) กรณีใกล้เคียง(Related case) แลกเปลี่ยนเรียนรู้(Collaboration) ห้องปฏิบัติการความเข้าใจโปรแกรม ฐานการช่วยเหลือ(Scaffolding) และเครื่องมือส่งเสริมความเข้าใจโปรแกรม(Tools enhanced Program Comprehension)การออกแบบและพัฒนาโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายฯ ผลของการศึกษาครั้งนี้สอดคล้องกับงานวิจัยของอิศรา ก้านจักร(2547) ที่ศึกษาเรื่องผลการจัดการสิ่งแวดล้อมทางการเรียนรู้บนเครือข่ายที่พัฒนาตามแนว Constructivism: Open Learning Environment(OLEs) สำหรับนักศึกษาระดับบัณฑิตศึกษา สาขาวิชาเทคโนโลยีการศึกษา สอดคล้องกับการศึกษาของ สุมาลี ชัยเจริญ(2547) ที่ศึกษาเรื่องการพัฒนารูปแบบการสร้างความรู้โดยใช้เทคโนโลยีสารสนเทศ สอดคล้องกับ Nulden(1997) ที่ศึกษาเรื่อง Designing Environments for Reflection and Collaborative Learning ซึ่งพบว่าผู้เรียนส่วนใหญ่เห็นว่าสิ่งแวดล้อมทางการเรียนรู้มีความเหมาะสมในทุกด้าน นอกจากนี้ยังสอดคล้องกับการศึกษาของ Hazeyama(2005) ที่ศึกษาเรื่อง Cognitive Apprenticeship-based Object-oriented Software Engineering Education Support ที่พบว่าฐานความช่วยเหลือสนับสนุนในการช่วยเหลือผู้เรียนในการกระทำงานและการทำงานเป็นกลุ่มในลักษณะการร่วมมือ การศึกษาของ Faessler, et al. (2006) ศึกษาเรื่อง Evaluating student motivation in constructivistic, Problem-based introductory computer science courses ที่พบว่า การใช้สถานการณ์ปัญหาสามารถกระตุ้นให้ผู้เรียนเกิดแรงจูงใจให้ผู้เรียนสามารถสร้างความรู้ได้ด้วยตนเอง งานวิจัยของ อิศรา ก้านจักร (2547) และสุมาลี ชัยเจริญ(2547) จะใช้เฉพาะหลักการตามแนวทฤษฎีคอนสตรัคติวิสต์เป็นหลัก โดยศึกษารูปแบบการทำความเข้าใจของนักเรียนในขณะที่เรียนจากสิ่งแวดล้อมทางการเรียนบนเครือข่ายที่พัฒนาตามแนวคอนสตรัคติวิสต์ Hazeyama(2005) & Faessler, et al.(2006)

จะใช้หลักการตามแนวทฤษฎีคอนสตรัคติวิสต์โดยสร้างสิ่งแวดล้อมที่เหมาะสมสำหรับผู้เรียน ได้แก่แบบจำลอง ฐานความช่วยเหลือ สถานการณ์ปัญหา เพื่อให้ผู้เรียนสามารถแก้ปัญหาได้ และ Nulden(1997) มุ่งเน้นออกแบบสิ่งแวดล้อมสำหรับการเรียนรู้และไตร่ตรอง สำหรับความแตกต่างของการออกแบบและพัฒนาโมเดลฯในงานวิจัยนี้ พบว่า ผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมทางการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรม สามารถสร้างความเข้าใจโปรแกรมที่มีลักษณะสำคัญคือ 1) สิ่งแทนความเข้าใจโปรแกรม(Program model) 2) สิ่งแทนความเข้าใจสถานการณ์(Situation model) ที่ผลการวิจัยเป็นเช่นนี้อาจเนื่องมาจากโมเดลสิ่งแวดล้อมทางการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม มีการออกแบบที่ใช้ ID theory และได้นำหลักการของทฤษฎีความเข้าใจโปรแกรมมาเป็นพื้นฐานในการออกแบบองค์ประกอบที่ช่วยส่งเสริมความเข้าใจโปรแกรม ตั้งแต่การออกแบบสถานการณ์ปัญหา ดังหลักฐานเชิงประจักษ์ที่ได้จากสัมภาษณ์ผู้เรียนว่า “สถานการณ์ปัญหาทำให้เราได้ฝึกวิเคราะห์ปัญหาจากสถานการณ์ที่เกิดขึ้นจริงและตรวจสอบแก้ไขปรับเปลี่ยนโปรแกรม ซึ่งสามารถนำมาใช้ในชีวิตจริงได้” ทั้งยังมีห้องปฏิบัติการส่งเสริมความเข้าใจโปรแกรมช่วยการฝึกหัดทางปัญญากับความเข้าใจโปรแกรม โดยเสนอแนะวิธีการคิดเกี่ยวกับความเข้าใจโปรแกรมสำหรับผู้เรียนที่มีประสบการณ์และไม่มีประสบการณ์ ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า “ห้องปฏิบัติการส่งเสริมความเข้าใจโปรแกรมช่วยทดสอบความสามารถของตนเอง ช่วยให้เราได้ ฝึกกระบวนการทางความคิด ช่วยตรวจสอบความรู้ทางด้านภาษาโปรแกรม การคิดวิเคราะห์และแก้ปัญหาในหลาย ๆ ลักษณะ ทำให้มีความเข้าใจโปรแกรมมากขึ้นด้วย” กรณีใกล้เคียงที่ช่วยให้ผู้เรียนสามารถเข้าถึงประสบการณ์ที่เกี่ยวข้องกับปัญหา ที่ผู้เรียนสามารถนำมาอ้างอิงได้ เชื่อมโยงประสบการณ์ที่เกี่ยวข้องมาใช้ได้ ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า “องค์ประกอบกรณีใกล้เคียง เป็นการนำเสนอตัวอย่างสถานการณ์ที่ใกล้เคียงกับโจทย์ หนูสามารถนำมาประยุกต์ให้เข้ากับโจทย์ที่เราทำอยู่ นำมาเทียบเคียงกันทำให้ทราบว่าควรแก้ไขโปรแกรมให้ถูกต้องได้ตรงไหน ทำให้เกิดความเข้าใจมากขึ้น” ธนาคารความรู้จะเป็นแหล่งที่รวบรวมข้อมูลต่าง ๆเป็นแหล่งที่ผู้เรียนสามารถค้นคว้าสร้างความรู้ได้ด้วยตนเอง ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า

องค์ประกอบธนาคารความรู้เป็นแหล่งข้อมูลที่ใช้ค้นคว้าเกี่ยวกับรูปแบบไวยากรณ์ภาษา C++ ได้ ตั้งแต่โครงสร้างโปรแกรม รูปแบบคำสั่งต่าง ๆ การเขียนโปรแกรมเรียกตัวเอง สอนหลักการวิธีคิดแก้ปัญหาอย่างเป็นระบบ ทำให้เข้าใจและนำไปใช้ได้ถูกต้อง มีการอธิบายกระบวนการทำงานทดสอบโปรแกรมและวิธีการแก้ไขโปรแกรมช่วยให้ผมมีประสบการณ์ในการปฏิบัติได้มากขึ้น ซึ่งการอธิบายขั้นตอนการทำงานทีละขั้นตอนเป็นรูปของผังโปรแกรมทำให้เข้าใจการเขียนโปรแกรมเรียกตัวเองได้มากขึ้นครับ

แลกเปลี่ยนเรียนรู้ เน้นการร่วมมือกันแก้ปัญหา สนับสนุนให้ผู้เรียนได้แลกเปลี่ยนประสบการณ์กับผู้อื่นเพื่อขยายมุมมองของความคิดทำให้เกิดความเข้าใจโปรแกรม ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า “องค์ประกอบแลกเปลี่ยนเรียนรู้ เป็นองค์ประกอบที่

ผู้เรียนเข้าไปในกระดานสนทนาเพื่อระดมสมองช่วยกันแก้ปัญหา แบ่งปันความรู้ในโจทย์ที่ไม่สามารถแก้ปัญหาได้ด้วยตนเอง ทำให้เข้าใจโปรแกรมได้ในส่วนที่ไม่รู้" รวมทั้งการปรึกษาผู้เชี่ยวชาญเพื่อขอคำแนะนำถึงวิธีการในการแก้ปัญหา ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า "องค์ประกอบปรึกษาผู้เชี่ยวชาญ ใช้เมื่อต้องการปรึกษาปัญหา หรือสิ่งที่ติดขัดไม่เข้าใจกับอาจารย์ทำให้ได้รับคำตอบที่ถูกต้องและตรงประเด็น" และเครื่องมือส่งเสริมความเข้าใจโปรแกรม ผู้เรียนสามารถสืบเสาะแสวงหาความรู้(Inquiry) และดาวน์โหลดโปรแกรมที่ต้องการได้ด้วยตนเอง ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า "เครื่องมือส่งเสริมความเข้าใจโปรแกรม เป็นการเตรียมลิงค์ไปยังเครื่องมือต่าง ๆที่จำเป็นต้องใช้ในการเขียนโปรแกรม ซึ่งช่วยผมในการดาวน์โหลดตัวแปลภาษาได้สะดวก เครื่องมือเหล่านี้มีส่วนที่จำมากที่ช่วยผมในการเรียนรู้การเขียนโปรแกรมเรียกตัวเอง"

จากหลักฐานเชิงประจักษ์และข้อค้นพบของงานวิจัยที่ศึกษาครั้งนี้ ดังกล่าวข้างต้นอาจสรุปได้ว่าการเรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมช่วยส่งเสริมความเข้าใจโปรแกรมให้แก่ผู้เรียน

5.2 การศึกษาความเข้าใจโปรแกรมของผู้เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม

จากผลการศึกษาความเข้าใจโปรแกรมของผู้เรียนในครั้งนี้ผู้วิจัยทำการเก็บข้อมูลโดยวิธีการสัมภาษณ์ผู้เรียนซึ่งเป็นกลุ่มเป้าหมายในระยะที่ 2 จำนวน 12 กลุ่มและกลุ่มเป้าหมายในระยะที่ 3 จำนวน 20 กลุ่ม สรุปได้ว่าผู้เรียนมีความเข้าใจโปรแกรมตามกรอบ Pennington (1987) ผู้เรียนสามารถสร้างสิ่งแทนความรู้(Mental representation)ได้ทั้ง 2 โมเดล ได้แก่ Program model และ Situation model ซึ่งสอดคล้องกับงานวิจัยของ Shneiderman(1979); Pennington(1987); Crosby and Stelovsky(1990); Corritore and Wiedenbeck(1999); Burkhardt, Detienne and Wiedenbeck(2002); Aschwanden and Crosby(2006) โดยกระบวนการทำความเข้าใจโปรแกรมของผู้เรียนสามารถแบ่งได้เป็น 3 ลักษณะ คือ

ลักษณะที่ 1 ผู้เรียนจะสร้างสิ่งแทนความเข้าใจคำสั่งโปรแกรม (Program model) ก่อน สร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model) ผู้เรียนในกลุ่มนี้จะสนใจคำ ที่ประกอบเป็นคำสั่งเพื่อทำความเข้าใจรูปแบบของคำสั่ง สามารถแยกแยะได้ว่าคำสั่งที่เห็นเป็นคำสั่งรูปแบบใด ต่อจากนั้นจึงสร้างสิ่งแทนความเข้าใจถึงกระบวนการทำงานของคำสั่ง โดยค่อย ๆ ก่อรูปความเข้าใจจากคำสั่ง 1 คำสั่งจนกลายเป็นกลุ่มของคำสั่ง ลำดับการทำงานของคำสั่งและพิจารณาวิเคราะห์ความเปลี่ยนแปลงที่เกิดขึ้นของข้อมูลทีละคำสั่ง ๆจนกระทั่งเข้าใจกระบวนการทำงานทุกคำสั่งในโปรแกรม ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า "เมื่อพิจาณาโปรแกรมต่อไปนี้ หนูไม่ทราบว่าโปรแกรมนี้ทำอะไรต้องไล่โปรแกรมก่อน การทำความเข้าใจโปรแกรมจะเริ่มจากคำสั่งในบรรทัดแรกเลยคือ #include <iostream>" ผู้เรียนในกลุ่มนี้มีความรู้หรือประสบการณ์ที่เกี่ยวข้องกับปัญหา(Problem domain) มีความรู้ในแนวคิดของการ

เขียนโปรแกรมในระดับต่ำ กระบวนการทำความเข้าใจการทำงานเป็นลำดับตามกรอบของ Pennington

ลักษณะที่ 2 ผู้เรียนจะสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model) จากคำที่คุ้นเคย(Beacon)ทำให้สามารถคาดเดาถึงวัตถุประสงค์(Subgoal)การทำงานของกลุ่มคำสั่งนี้ได้ ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า "*จุดเริ่มต้นที่หนูดูคือ input(x,n) เพราะเป็นคำหรือชื่อฟังก์ชันที่ค่อนข้างคุ้นเคย อาจารย์จะสอนและใช้ชื่อ input อยู่เกือบทุกตัวอย่าง ฟังก์ชัน input ต้องเป็นการรับข้อมูลทางแป้นพิมพ์อย่างแน่นอน เพราะอาจารย์จะใช้ชื่อนี้สื่อความหมายว่าเป็นฟังก์ชันในการรับข้อมูลเกือบทุกโปรแกรม*" ผู้เรียนในกลุ่มนี้เป็นผู้เรียนที่มีประสบการณ์ในการพัฒนาโปรแกรมในระดับปานกลาง เพราะมีความรู้เดิมที่เกี่ยวข้องกับปัญหา (Problem domain) สามารถคาดเดาเหตุการณ์หรือวัตถุประสงค์ของการทำงานโปรแกรมได้ในบางส่วน การทำความเข้าใจโปรแกรมของผู้เรียนจะเป็นกระบวนการที่กระทำเหมือนผู้เรียนในลักษณะที่ 1 เพื่อทดสอบสมมุติฐานหรือสิ่งที่คาดเดาว่าถูกต้องหรือไม่ถูกต้อง ผู้เรียนสร้างสิ่งแทนความเข้าใจโปรแกรมและสิ่งแทนความเข้าใจสถานการณ์ของคำสั่งโปรแกรมทั้งหมดก่อนสรุปเป็นวัตถุประสงค์หลักของโปรแกรมได้

ลักษณะที่ 3 ผู้เรียนจะสร้างสิ่งแทนความเข้าใจสถานการณ์(Situation model) จากคำที่คุ้นเคย(Beacon)ทำให้สามารถคาดเดาถึงวัตถุประสงค์การทำงานของโปรแกรมได้(Functional) ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ผู้เรียนว่า

ดูจากโปรแกรมแล้วเป็นโปรแกรมที่มีขนาดเล็ก มีเพียงฟังก์ชัน main และ compute เท่านั้น ส่วนที่สำคัญคือการเข้าใจกระบวนการทำงานของคำสั่ง compute เท่านั้นก็สามารถทราบได้ว่าโปรแกรมทำอะไร เพราะกระบวนการทำงานของโปรแกรมไม่มีอะไรที่ซับซ้อน เป็นคำสั่งพื้นฐานธรรมดา เริ่มพิจารณาในส่วนของฟังก์ชันคือ compute เพราะบอกถึงวัตถุประสงค์ของโปรแกรม โปรแกรมนี้มีแค่ฟังก์ชันเดียว ในส่วนของ main ไม่มีอะไรมีแต่รับข้อมูล เรียกฟังก์ชันและแสดงผลเท่านั้นไม่มีอะไรซับซ้อนแต่ส่วนที่ต้องวิเคราะห์หรือทำความเข้าใจ คือฟังก์ชัน compute สำคัญที่สุด

ผู้เรียนในกลุ่มนี้เป็นผู้เรียนที่มีประสบการณ์ในการพัฒนาโปรแกรมในระดับสูง เพราะมีความรู้เดิมที่เกี่ยวข้องกับปัญหา(Problem domain) สามารถคาดเดาเหตุการณ์หรือวัตถุประสงค์ของการทำงานได้โดยไม่ต้องทำความเข้าใจโปรแกรมทั้งหมด การทำความเข้าใจโปรแกรมของผู้เรียนจะเป็นกระบวนการที่กระทำเหมือนผู้เรียนในลักษณะที่ 1 เพื่อทดสอบสมมุติฐานหรือสิ่งที่คาดเดาว่าถูกต้องหรือไม่เฉพาะกลุ่มของคำสั่งที่สนใจ โดยสร้างสิ่งแทนความเข้าใจรูปแบบคำสั่งและสิ่งแทนความเข้าใจสถานการณ์ของโปรแกรมเฉพาะส่วนที่สนใจก่อนสรุปเป็นวัตถุประสงค์หลักของโปรแกรมได้

จากผลของการศึกษากระบวนการทำความเข้าใจโปรแกรมของผู้เรียน สามารถวิเคราะห์ได้ว่า ความรู้เดิมในเรื่องของรูปแบบคำสั่ง แผนงานหรือสคริปต์ในการทำงาน เป็นลำดับขั้นตอนของผู้เรียนมีผลต่อกระบวนการทำความเข้าใจโปรแกรมอย่างมาก ผู้เรียนที่มีประสบการณ์มี

ความรู้เดิมที่สะสมไว้มากกว่าจะใช้เวลาในการทำความเข้าใจโปรแกรมน้อยกว่า จากหลักฐานเชิงประจักษ์ที่ว่า "เริ่มพิจารณาฟังก์ชันคือ compute เพราะบอกถึงวัตถุประสงค์ของโปรแกรม โปรแกรมนี้มีแค่ฟังก์ชันเดียว ในส่วนของ main. ไม่มีอะไรมีแต่รับข้อมูล เรียกฟังก์ชัน และแสดงผลเท่านั้นไม่มีอะไรซับซ้อนแต่ส่วนที่ต้องวิเคราะห์หรือทำความเข้าใจคือ ฟังก์ชัน compute สำคัญที่สุด" เห็นได้ว่าผู้เรียนสนใจเฉพาะกลุ่มของคำสั่งเท่านั้น ซึ่งผู้เรียนที่มีความรู้หรือมีประสบการณ์ในการเขียนโปรแกรมมาบ้างเคยปฏิบัติการทดสอบและรันโปรแกรมสามารถคาดเดาวัตถุประสงค์การทำงานของโปรแกรมในบางส่วนได้ แต่สำหรับผู้เรียนที่ไม่มีความรู้เกี่ยวข้องกับปัญหามากนัก ต้องเริ่มทำความเข้าใจโปรแกรมในทุก ๆคำสั่งจึงจะสามารถเข้าใจกระบวนการทำงานของโปรแกรมทั้งหมดได้ ดังหลักฐานเชิงประจักษ์ที่ว่า "อาจารย์สอนรูปแบบของคำสั่งแต่ละชนิดแล้ว การทำความเข้าใจนั้นต้องรู้ว่าคำสั่งแบบไหนทำอะไร เช่นการประกาศตัวแปร คำสั่งในการรับข้อมูล คำสั่งในการแสดงผล คำสั่งในการเรียกฟังก์ชัน โดยเฉพาะรูปแบบการเขียนโปรแกรมเรียกตัวเอง การทำความเข้าใจโปรแกรม ต้องรู้ทุกคำ ทุกคำสั่ง ทุกเงื่อนไข ทุกทางเลือกถ้าไม่รู้ก็ไล่โปรแกรมไม่ได้" และ "การแก้ไขภารกิจาต้องมีความรู้เรื่อง การคิดอัตราดอกเบี้ย คำสั่งในการเรียกตัวเอง การแก้ไขคำสั่งให้สั้นลงเพื่อให้การทำงานให้เร็วขึ้น"

จากผลการศึกษาความเข้าใจโปรแกรมของผู้เรียนที่เรียนด้วยโมเดลฯ ผู้วิจัยสามารถนำข้อค้นพบที่ได้จากการวิจัยนำมาออกแบบและพัฒนาโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมให้แก่ผู้เรียนได้หลากหลายลักษณะให้สอดคล้องและเหมาะสมกับความต้องการของผู้เรียนที่มีความรู้เดิมที่แตกต่างกัน โดยออกแบบองค์ประกอบของโมเดลฯเป็นแหล่งความรู้ที่ผู้เรียนเพื่อเติมเต็มความรู้ที่ขาดหายไปของผู้เรียนเพื่อสนับสนุนและส่งเสริมกระบวนการทำความเข้าใจของผู้เรียนได้อย่างมีประสิทธิภาพมากขึ้น

5.3 การศึกษาความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรม

การศึกษาความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลสิ่งแวดล้อมการเรียนบนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมจากการศึกษาในระยะที่ 2 และระยะที่ 3 ข้อค้นพบที่ได้จากการวิเคราะห์แบบสำรวจความคิดเห็นของผู้เรียนที่มีต่อการเรียนด้วยโมเดลฯ ผลสรุปได้ดังนี้

(1) ด้านการใช้โมเดลฯ พบว่า ผู้เรียนสามารถเลือกใช้องค์ประกอบต่าง ๆได้ง่าย ผู้เรียนสามารถควบคุมการเรียนรู้โดยใช้องค์ประกอบต่าง ๆของโมเดลฯได้อย่างอิสระ และช่วยส่งเสริมและสนับสนุนผู้เรียนให้เข้าใจโปรแกรมเรียกตัวเอง

(2) ด้านประโยชน์ที่ได้รับ พบว่า โมเดลฯเปิดโอกาสให้เกิดการปฏิสัมพันธ์ (Interactive)ระหว่างผู้เรียนกับผู้สอนและผู้เรียนกับผู้เรียนหรือผู้เรียนกับเนื้อหาบทเรียน โมเดลฯสามารถนำเสนอเนื้อหาในรูปแบบของสื่อหลายมิติ ช่วยทำให้ผู้เรียนเกิดความเข้าใจได้ดี มีแหล่งการเรียนรู้บนเครือข่ายที่หลากหลาย รวมทั้งผู้เรียนสามารถเรียนตามความพร้อมความถนัดและความสนใจของตน โดยไม่มีข้อจำกัดทางสถานที่และเวลาของการเรียนรู้

ซึ่งสอดคล้องกับผลการวิจัยของ สุมาลี ชัยเจริญ(2547); อิศรา ก้านจักร(2547); จารุณี ซามาตย์(2552) ที่พบว่า ผู้เรียนมีความคิดเห็นเกี่ยวกับสิ่งแวดล้อมทางการเรียนรู้ฯ และนวัตกรรมการเรียนรู้ที่ออกแบบและพัฒนาขึ้น ว่าสามารถส่งเสริมทักษะกระบวนการแสวงหาความรู้ด้วยตนเอง และกระบวนการคิดที่ผ่านการลงมือปฏิบัติ รวมทั้งช่วยกระตุ้นให้เกิดความสนใจในการเรียนรู้

ข้อค้นพบของการวิจัยในการศึกษาครั้งนี้อาจเนื่องมาจากการออกแบบโมเดลฯที่ประสานร่วมกันทั้งหลักการ ทฤษฎีและคุณลักษณะของสื่อ โดยเฉพาะการออกแบบที่อาศัยทฤษฎีการเรียนรู้คอนสตรัคติวิสต์ที่เน้นการสร้างความรู้ และสถานการณ์ปัญหาที่ส่งเสริมความเข้าใจโปรแกรมตามกรอบของ Pennington(1987) รวมทั้งองค์ประกอบทั้งหมดของโมเดลสิ่งแวดล้อมการเรียนรู้บนเครือข่ายฯ ตลอดจนการอาศัยคุณลักษณะของสื่อบนเครือข่ายที่สามารถส่งเสริมการสร้างความรู้ตลอดจนขยายความคิดของผู้เรียนได้เป็นอย่างดี ประกอบกับการออกแบบให้มีองค์ประกอบที่ส่งเสริมความเข้าใจโปรแกรม เช่น ห้องปฏิบัติการส่งเสริมความเข้าใจโปรแกรม ทั้งยังมีองค์ประกอบที่ช่วยในการเชื่อมโยงความรู้และประสบการณ์ให้สามารถนำไปแก้ปัญหาได้ เช่น กรณีใกล้เคียง เครื่องมือที่ส่งเสริมความเข้าใจโปรแกรม แลกเปลี่ยนเรียนรู้ระหว่างผู้เรียนด้วยกันหรือกับผู้เชี่ยวชาญได้ตลอดเวลา อันจะส่งผลต่อการส่งเสริมกระบวนการทำความเข้าใจโปรแกรมของผู้เรียนได้อย่างมีประสิทธิภาพต่อไป

5.4 การศึกษาผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลสิ่งแวดล้อมการเรียนรู้ที่ส่งเสริมความเข้าใจโปรแกรม

จากการศึกษาเกี่ยวกับผลสัมฤทธิ์ทางการเรียนของผู้เรียนที่เรียนด้วยโมเดลฯ เมื่อพิจารณาจากค่าสถิติพื้นฐานที่แสดงคะแนนเฉลี่ยของการทดสอบก่อนเรียนและหลังเรียนจากการศึกษาในระยะที่ 2 และระยะที่ 3 พบว่า ผู้เรียนมีผลสัมฤทธิ์ทางการเรียนสูงขึ้น ซึ่งสอดคล้องกับการศึกษาของ อิศรา ก้านจักร(2547); จารุณี ซามาตย์(2552); ชไมพร อินทร์แก้ว(2552); อิศรา ก้านจักร(2552) อาจเนื่องมาจากการเรียนจากโมเดลฯส่งเสริมให้ผู้เรียนได้ลงมือสร้างความรู้ด้วยตนเองในทุกขั้นตอนตามศักยภาพของผู้เรียนที่ต้องการเรียนรู้ตามความสนใจ อีกทั้งในการเรียนรู้ผู้เรียนมีโอกาสในการแลกเปลี่ยนความคิดของตนเอง การถกปัญหา การอภิปรายการสนทนาผ่านเครือข่าย ซึ่งช่วยให้ผู้เรียนขยายแนวคิดเกี่ยวกับการเรียนรู้ได้มากขึ้นจนกระทั่งสามารถแก้ไขภารกิจได้ด้วยตนเอง ดังหลักฐานเชิงประจักษ์ที่ได้จากการสัมภาษณ์ของผู้เรียนที่ว่า “สิ่งแวดล้อมการเรียนรู้ มีเครื่องมือช่วยคิด เมื่อไม่เข้าใจสามารถปฏิบัติได้ด้วยตนเองสามารถ Run หาคำตอบได้ สามารถเรียนรู้ได้ทุกที่ ทุกเวลา ไม่จำเป็นต้องมาเรียน คือสามารถศึกษาเวลาไหนก็ได้ ได้ดูตัวอย่างเพิ่มขึ้น ได้ความรู้มากขึ้น โมเดลฯมีแหล่งการเรียนรู้ที่ทำให้เข้าใจเนื้อหาง่าย สามารถเรียนรู้ได้ตลอดเวลา ใช้งานง่าย มีตัวอย่างครอบคลุมเนื้อหา”

จากเหตุผลดังกล่าวการเรียนด้วยโมเดลสิ่งแวดล้อมทางการเรียนรู้บนเครือข่ายที่ส่งเสริมความเข้าใจโปรแกรมจึงอาจมีผลทำให้ผู้เรียนมีผลสัมฤทธิ์ทางการเรียนสูงขึ้น