

บทที่ 1

บทนำ

1.1 ความสำคัญของปัญหา

การศึกษาและวิจัยในสาขาต่างๆของวิศวกรรมศาสตร์มีความจำเป็นต้องทำการคำนวณด้วยข้อมูลจำนวนมากและใช้สมการทางคณิตศาสตร์ที่ซับซ้อน การคำนวณต่างๆเหล่านี้แม้ว่าจะสามารถคำนวณด้วยมือหรือใช้เครื่องคิดเลขก็จะมีข้อจำกัดในแง่ของจำนวนตัวแปรที่สามารถคำนวณได้ ทำให้นักศึกษาเรียนรู้ได้เฉพาะตัวอย่างที่ง่าย ๆซึ่งมักจะไม่น่าพบกันทางปฏิบัติ วิธีการอย่างหนึ่งในการแก้ไขข้อจำกัดนี้ซึ่งค่อนข้างได้ผลเป็นที่น่าพอใจคือ การอาศัยคอมพิวเตอร์ในการคำนวณสมการและข้อมูลต่างๆแทน ดังนั้นจึงได้มีการทำวิจัยทางด้านการพัฒนาซอฟต์แวร์สำหรับช่วยสอน (computer-aided instruction) ทางด้านวิศวกรรมศาสตร์สาขาต่างๆในช่วงที่ผ่านมา อย่างไรก็ตามมีปัญหาหลายประการที่จำกัดการเผยแพร่ซอฟต์แวร์ประเภทนี้ไปใช้งานในสถานศึกษาต่างๆ ได้แก่

- ไม่มีการพัฒนาซอฟต์แวร์อย่างต่อเนื่อง ดังนั้นข้อผิดพลาดของซอฟต์แวร์จึงไม่ได้ถูกแก้ไข รวมทั้งไม่มีการปรับให้ใช้กับระบบปฏิบัติการที่เปลี่ยนไปเรื่อยๆเช่น Microsoft Windows
- ซอฟต์แวร์ถูกพัฒนาขึ้นโดยอิงอยู่บนรายวิชาที่เปิดสอนในมหาวิทยาลัยนั้นๆ รวมทั้งเจาะจงเฉพาะรายวิชาที่เกี่ยวข้องจึงอาจไม่เหมาะสมกับรายวิชาที่เปิดสอนในมหาวิทยาลัยอื่นๆ นอกจากนี้เมื่อต้องการซอฟต์แวร์สำหรับช่วยในการสอนรายวิชาอื่นๆก็จำเป็นต้องทำการพัฒนาใหม่จากเริ่มต้น แม้ว่า การคำนวณส่วนใหญ่จะคล้ายกับซอฟต์แวร์ที่พัฒนาขึ้นแล้ว
- ซอฟต์แวร์ถูกพัฒนาโดยใช้ภาษาคอมพิวเตอร์ทั่วไปเช่น ภาษา C Pascal หรือ Basic โดยมีการเผยแพร่เฉพาะซอฟต์แวร์ที่ถูกคอมไพล์เรียบร้อยแล้ว ทำให้ไม่สามารถแก้ไขส่วนของซอฟต์แวร์ให้เหมาะสมกับผู้ใช้ของหน่วยงานอื่นๆ ได้ นอกจากนี้การศึกษาค้นฉบับของโปรแกรมที่เขียนด้วยภาษาดังกล่าวก็ทำได้ยาก

ในปัจจุบันซอฟต์แวร์สำหรับการเรียนการสอนและการวิจัยทางด้านวิศวกรรมไฟฟ้าที่นิยมใช้ในมหาวิทยาลัยต่างๆทั้งในประเทศไทยและระดับสากลได้แก่ โปรแกรม MATLAB [Mathworks, Online] โปรแกรม MATLAB มีข้อดีหลายอย่าง [Chapman, 2002] เช่น มีฟังก์ชันสำหรับการคำนวณทางคณิตศาสตร์จำนวนมาก เป็นซอฟต์แวร์ที่ได้ตอบทันทีต่อคำสั่งจึงง่ายต่อการใช้งาน มีฟังก์ชันสำหรับสร้างรูปภาพแบบต่างๆ สามารถพัฒนาฟังก์ชันสำหรับการคำนวณเองได้ มีโมดูลจำนวนมากสำหรับการคำนวณเฉพาะทาง มีหนังสือประกอบการใช้งานจำนวนมาก และสามารถใช้งานบนเครื่องคอมพิวเตอร์เกือบทุกประเภท อย่างไรก็ตามการนำโปรแกรม MATLAB มาใช้ในการเรียนการสอน [Etter, 1997] มีอุปสรรคหลักคือ ค่าใช้จ่ายที่สูงมาก เพราะโปรแกรม MATLAB มีราคาต่อผู้ใช้ประมาณ 40,000 บาทและสำหรับแต่ละโมดูลย่อยอีก 20,000 บาท เมื่อคิดค่าใช้จ่ายสำหรับการสอนในห้องเรียนครั้งละ 30 คน ก็จะเป็นจำนวนเงินในหลักหลายล้านบาท ทางเลือกที่เป็นไปได้ในการลดค่าใช้จ่ายทางด้านซอฟต์แวร์คือ การใช้โปรแกรม GNU Octave [Eaton, 2001] ซึ่ง

เป็นซอฟต์แวร์สำหรับคำนวณทางคณิตศาสตร์ที่มีรูปแบบการใช้งานคล้ายกับโปรแกรม MATLAB แต่ใช้ลิขสิทธิ์แบบเปิดเผยต้นฉบับโปรแกรม

การประยุกต์ใช้ซอฟต์แวร์คอมพิวเตอร์เป็นสื่อช่วยในการเรียนรู้เนื้อหาวิชาทางวิศวกรรมศาสตร์ เป็นรูปแบบของการสอนที่พบทั่วไปในปัจจุบัน โดยเฉพาะรายวิชาที่เกี่ยวข้องกับการคำนวณที่ซับซ้อน เช่น CAD/CAM ทฤษฎีระบบควบคุม เป็นต้น ปัญหาที่พบในการสอนซึ่งอาศัยการประยุกต์ใช้งานซอฟต์แวร์คอมพิวเตอร์ ในการอธิบายประกอบทฤษฎีหรือประมวลผลเพื่อหาคำตอบ คือ

- ผู้สอนไม่สามารถรับรู้สถานะของการเรียนรู้ หรือปัญหาที่เกิดขึ้นกับผู้เรียนได้จากบริเวณที่ผู้สอนกำลังดำเนินการสอนอยู่ เมื่อผู้เรียนบางคนต้องการความช่วยเหลือในการแก้ปัญหา ผู้สอนจำเป็นต้องเดินไปที่เครื่องคอมพิวเตอร์ซึ่งผู้เรียนคนนั้นใช้งานอยู่ ทำให้การเรียนการสอนดำเนินไปได้อย่างช้า
- การส่งผ่านเนื้อหาจากผู้สอนถึงผู้เรียนก็ทำได้เฉพาะผ่านทางบรรยายและการใช้สื่อประเภทเครื่องฉายภาพ
- การที่ผู้เรียนจะส่งผลลัพธ์จากการเรียนรู้หรือตอบโจทย์ที่ผู้สอนกำหนดให้ ก็จำเป็นต้องอาศัยการบันทึกลงในสื่อประเภทแผ่นดิสก์ ซึ่งผู้สอนจำเป็นต้องไปเปิดอ่านแยกแต่ละผู้เรียนในภายหลัง

1.2 หลักการและแนวความคิด

จากปัญหาต่างๆข้างต้นนี้ การวิจัยครั้งนี้จึงเน้นการพัฒนาซอฟต์แวร์บนหลักการพื้นฐาน 3 ประการคือ การพัฒนาแบบยั่งยืน การเอื้อให้เกิดการพัฒนาอย่างต่อเนื่อง และการจัดเตรียมเอกสารประกอบ ซึ่งมีรายละเอียดดังนี้

- การพัฒนาซอฟต์แวร์แบบยั่งยืน หมายถึงการพัฒนาซอฟต์แวร์โดยใช้รูปแบบลิขสิทธิ์แบบเปิดเผยต้นฉบับโปรแกรม (open source license) [Open Source Initiative, Online] กล่าวโดยย่อคือ อนุญาตให้เผยแพร่ซอฟต์แวร์ผ่านสื่อต่างๆเช่น อินเทอร์เน็ตอย่างเสรี เปิดเผยถึงต้นฉบับของซอฟต์แวร์ อนุญาตให้ผู้ผู้คัดแปลงบางส่วนของซอฟต์แวร์ให้เหมาะสมกับงานของตนเอง รวมทั้งไม่เสียค่าใช้จ่ายในการใช้งาน ลิขสิทธิ์แบบเปิดเผยต้นฉบับโปรแกรมจะเอื้อให้เกิดการพัฒนาต่อยอดจากส่วนที่ได้พัฒนาไปแล้ว ทำให้เมื่อสิ้นสุดโครงการวิจัยแล้วซอฟต์แวร์ก็ยังคงได้รับการร่วมพัฒนาอย่างต่อเนื่องจากผู้ใช้งานหลายหน่วยงานได้อีก ทางศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติได้ระบุลิขสิทธิ์แบบเปิดเผยต้นฉบับโปรแกรมนี้สำหรับระบบปฏิบัติการลินุกซ์ทะเลและโปรแกรมออฟฟิศทะเล ซึ่งมีการเผยแพร่และนำไปใช้งานอย่างเสรีในปัจจุบัน
- การเอื้อให้เกิดการพัฒนาอย่างต่อเนื่อง หมายถึงรูปแบบของภาษาคอมพิวเตอร์ระดับสูงที่ใช้ในการพัฒนาซอฟต์แวร์จะต้องเอื้อให้ผู้ใช้งานสามารถศึกษาและปรับปรุงได้อย่างง่ายดาย รวมทั้งมีปัจจัยที่จำเป็นสำหรับการศึกษาและวิจัยทางวิศวกรรมศาสตร์ได้แก่ การมีฟังก์ชันทางคณิตศาสตร์ที่หลากหลาย ความสามารถในการสร้างรูปกราฟทั้งแบบ 2 มิติและ 3 มิติ และการรองรับข้อมูลประเภทต่างๆที่ใช้ในการคำนวณทางคณิตศาสตร์
- การจัดเตรียมเอกสารประกอบ หมายถึงการจัดเตรียมเอกสารประกอบการใช้งานซอฟต์แวร์ที่เหมาะสมและพอเพียงได้แก่ คู่มือการใช้งานซอฟต์แวร์ คู่มือแนะนำการเขียนฟังก์ชันใหม่ และเอกสารประกอบการสอน ซึ่งแสดงถึงการประยุกต์ใช้ซอฟต์แวร์ เอกสารทั้งหมดจะอยู่ในรูปแบบลิขสิทธิ์แบบเปิดเผยเนื้อหาเอกสาร (open content license) [Opencontent, Online] กล่าวคือ มีไฟล์ต้นฉบับของเอกสารทั้งหมดรวมไปกับ

ซอฟต์แวร์ด้วยและอนุญาตให้ผู้ใช้ปรับปรุงแก้ไขเนื้อหาของเอกสารได้ ดังนั้นผู้ใช้ทั่วไปจะสามารถปรับปรุงและเพิ่มเติมเนื้อหาของเอกสารให้เหมาะสมกับการใช้งานภายในหน่วยงานตนเองได้

เนื่องจากจากเป้าหมายที่จะลดการจัดซื้อซอฟต์แวร์จากต่างประเทศ โครงการวิจัยนี้จะพัฒนาซอฟต์แวร์ในรูปของโมดูลเพิ่มเติมทางด้านวิศวกรรมไฟฟ้าสำหรับใช้งานร่วมกับโปรแกรม GNU Octave [Eaton, 2001] ซึ่งเป็นซอฟต์แวร์สำหรับคำนวณทางคณิตศาสตร์ที่มีรูปแบบการใช้งานคล้ายกับโปรแกรม MATLAB แต่ใช้ลิขสิทธิ์แบบเปิดเผยต้นฉบับโปรแกรม การใช้งานโปรแกรม GNU Octave ในปัจจุบันยังอยู่ในวงจำกัดเพราะมีโมดูลสำหรับการคำนวณเฉพาะทางที่น้อยมากเมื่อเทียบกับโปรแกรม MATLAB การพัฒนาโมดูลเพิ่มเติมจะใช้รูปแบบของการโปรแกรมแบบ m-file ซึ่งเป็นวิธีการสร้างคำสั่งใหม่โดยการเรียกใช้ชุดคำสั่งที่มีอยู่แล้วในโปรแกรม GNU Octave ชุดคำสั่งใหม่ในโมดูลที่เขียนขึ้นจะอยู่ในรูปของไฟล์แบบตัวอักษรซึ่งสามารถศึกษาและแก้ไขได้ง่าย ไม่จำเป็นต้องทำการคอมไพล์ใหม่ และสามารถนำไปใช้งานในโปรแกรม MATLAB ได้ เนื่องจากโปรแกรม GNU Octave มีลิขสิทธิ์แบบเปิดเผยต้นฉบับโปรแกรม คณะผู้วิจัยจะสามารถเผยแพร่โมดูลที่พัฒนาขึ้นพร้อมกับโปรแกรม GNU Octave เมื่อสิ้นสุดโครงการวิจัย ซึ่งผู้ใช้ทั่วไปสามารถใช้งานได้ทันทีโดยไม่เสียค่าใช้จ่ายและเผยแพร่ต่อโดยไม่ละเมิดลิขสิทธิ์

1.3 วัตถุประสงค์

1. เพื่อเผยแพร่และชักจูงให้มีการใช้งานซอฟต์แวร์แบบเปิดเผยต้นฉบับโปรแกรม ซึ่งจะช่วยประหยัดงบประมาณของประเทศในการซื้อซอฟต์แวร์จากต่างประเทศ
2. เพื่อพัฒนาโมดูลเพิ่มเติมให้กับโปรแกรม GNU Octave เพื่อใช้สำหรับการเรียนการสอนและการวิจัยทางด้านวิศวกรรมไฟฟ้า
3. เพื่อจัดทำเอกสารภาษาไทยประกอบการใช้งานโปรแกรม GNU Octave

1.4 ผลงานวิจัยที่เกี่ยวข้อง

ข้อกำหนด Learning Technology Systems Architecture (LTSA) [LTSA, 2003] ซึ่งถูกระบุในมาตรฐาน IEEE 1484 เป็นการแจกแจงถึงโครงสร้างในระดับบนและการจัดลำดับชั้นต่างๆ ที่เกี่ยวข้องกับระบบสารสนเทศที่สนับสนุนการเรียน การสอน และการอบรมต่างๆ เป้าหมายของข้อกำหนด LTSA ได้แก่ การให้ภาพที่ชัดเจนของโครงสร้างระบบสารสนเทศเพื่อการศึกษา ทั้งในปัจจุบันและอนาคต โดยไม่ขึ้นกับรูปแบบเนื้อหา วัฒนธรรม และแพลตฟอร์ม การสนับสนุนความเข้ากันได้และการทำงานร่วมกันด้วยการระบุถึงกลไกการเชื่อมต่อที่มีผลกระทบอย่างชัดเจน และการคาดการณ์แนวทางของเทคโนโลยี สำหรับช่วง 5-10 ปีในอนาคต

ข้อกำหนด Public and Private Information (PAPI) [PAPI, Online] เป็นความพยายามในการสร้างมาตรฐานของข้อมูลของผู้เรียนที่สามารถใช้งานได้ เพื่อใช้เป็นข้อกำหนดในการแลกเปลี่ยนข้อมูลระหว่างระบบเข้าด้วยกัน ข้อกำหนดของ PAPI ระบุถึงรายละเอียดในการบันทึกข้อมูลต่างๆ ในขั้นตอนการเรียนรู้ เช่น องค์ความรู้ ความถนัด รูปแบบการเรียน เป็นต้น โดยสามารถเลือกบันทึกด้วยความละเอียดที่แตกต่างกัน มาตรฐาน PAPI สนับสนุนมุมมองที่หลากหลายของแบบจำลองการเรียนรู้ เช่น Learner, Teacher, Parent, School, Employer เป็นต้น พร้อมทั้งระบุความเป็นส่วนตัวและความปลอดภัยของข้อมูล

Virtual Classroom Manager [Sen, 2000] เป็นระบบซอฟต์แวร์ซึ่งทำหน้าที่จัดการหลักสูตรทั้งแบบในและนอกพื้นที่ของสถานศึกษา แนวความคิดหลักของระบบซอฟต์แวร์ได้แยกกระบวนการทำงานออกเป็น

หลายลำดับชั้น โดยลำดับชั้นในส่วนหน้าได้อาศัยเทคโนโลยีอินเทอร์เน็ตเพื่อเป็นเครื่องมือในการเข้าถึงเนื้อหาที่เป็นแบบมัลติมีเดีย ลำดับชั้นกลางเป็นส่วนองบริการที่พัฒนาขึ้นตามนิยามขององค์ประกอบต่างๆในโครงสร้างของ LTSA ด้วยรูปแบบของวัตถุในเว็บ (Web Object) และเชื่อมต่อเข้าสู่ลำดับชั้นในส่วนหลังซึ่งใช้ระบบฐานข้อมูลหรือระบบไฟล์เซิร์ฟเวอร์ในการจัดเก็บข้อมูลต่างๆ งานวิจัยนี้เน้นที่การศึกษาโครงสร้างของวัตถุในเว็บที่จำเป็นในระบบซอฟต์แวร์ โดยอิงตามโครงสร้างของ LTSA พร้อมทั้งรองรับรูปแบบข้อมูลที่กำหนดใน PAPI

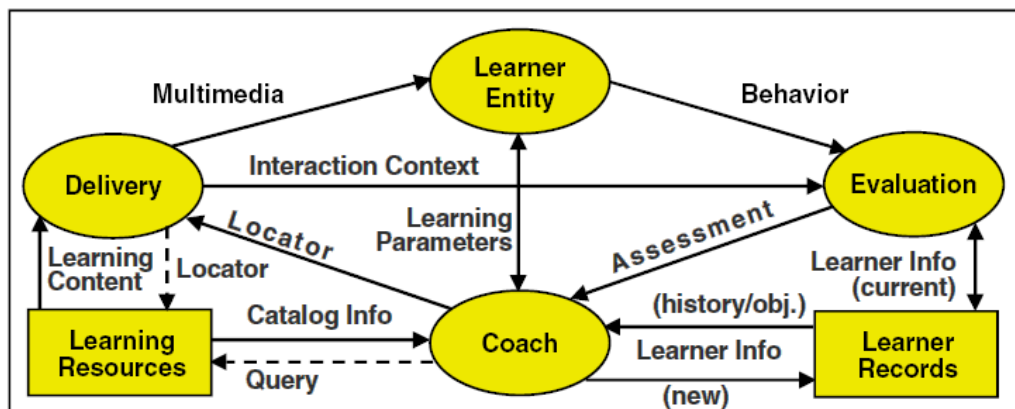
บทที่ 2

ทฤษฎีและความรู้พื้นฐาน

2.1 โครงสร้างของเทคโนโลยีการเรียนรู้

ปัจจุบัน วงการศึกษาได้มีการประยุกต์ใช้เทคโนโลยีสารสนเทศเพื่อเพิ่มประสิทธิภาพและขยายโอกาสในการเรียนรู้ผ่านเครือข่ายอินเทอร์เน็ต ทำให้ผู้เรียนสามารถเข้าถึงเนื้อหาได้จากทุกที่และทุกเวลา เนื่องจากการเรียนการสอนผ่านทางเครือข่ายคอมพิวเตอร์ได้กลายเป็นหนึ่งในประเด็นที่สำคัญของการพัฒนาระบบสารสนเทศ ทางสมาคม IEEE จึงได้ดำเนินการศึกษารูปแบบของการประยุกต์ใช้เทคโนโลยีสารสนเทศในการเรียนการสอน จากนั้นทางสมาคม IEEE จึงกำหนดโครงสร้างและองค์ประกอบต่างๆของเทคโนโลยีสารสนเทศที่เป็นพื้นฐานสำหรับสนับสนุนการเรียนรู้ในรูปแบบของมาตรฐานขึ้น เพื่อให้การพัฒนาเทคโนโลยีทางด้านนี้มีแนวทางที่ชัดเจน โครงสร้างของระบบสารสนเทศเพื่อการเรียนรู้ได้ถูกกำหนดไว้ในมาตรฐาน IEEE 1484 ในหัวข้อ IEEE Standard for Learning Technology — Learning Technology Systems Architecture (LTSA)

มาตรฐาน IEEE 1484 เป็นข้อกำหนดที่มุ่งเป้าหมายในการระบุนามธรรมในระดับสูง ซึ่งอธิบายถึงองค์ประกอบต่างๆภายในระบบของการเรียนรู้ การศึกษา และการฝึกอบรม โดยครอบคลุมถึงการอบรมด้วยคอมพิวเตอร์ (computer-based training) การสนับสนุนการสอน (computer assisted instruction) ระบบช่วยแนะนำแบบฉลาด (intelligent tutoring) เป็นต้น ข้อดีของการใช้โครงสร้าง LTSA ในการพัฒนาระบบสารสนเทศเพื่อการเรียนรู้ คือ การมีแนวทางและภาพรวมที่ชัดเจนเพื่อความเข้าใจในระบบทั้งปัจจุบันและในอนาคตอย่างน้อย 5-10 ปี ซึ่งนำไปสู่การเข้ากันได้และการใช้งานร่วมกันของเทคโนโลยีที่พัฒนาขึ้น รวมทั้งการปรับเข้าสู่ระบบสารสนเทศอื่นๆที่พัฒนาขึ้นมาในภายหลัง โครงสร้างขององค์ประกอบต่างๆของ LTSA ได้แสดงไว้ในรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของ LTSA

รายละเอียดตามที่ระบุไว้ในโครงสร้างของ LTSA เป็นการนิยามโครงสร้างของระบบสารสนเทศในระดับสูง โดยแบ่งองค์ประกอบเป็น 3 กลุ่มหลักได้แก่

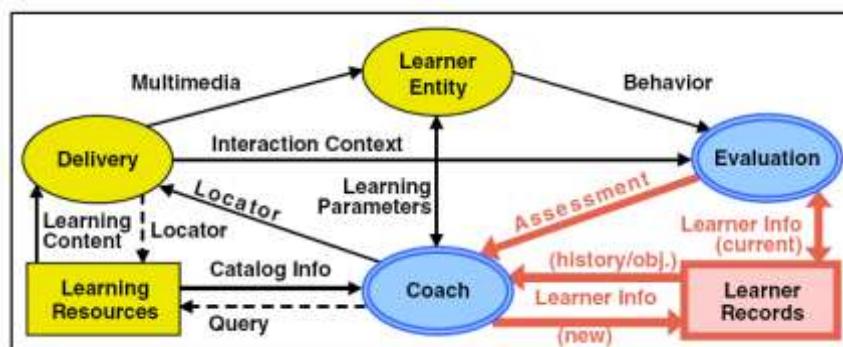
1. กระบวนการ (process) หมายถึง องค์ประกอบที่มีการเปลี่ยนแปลงด้วยตนเอง (active component) ของระบบสารสนเทศ ซึ่งทำหน้าที่เปลี่ยนหรือประมวลผลสิ่งที่รับเข้าให้เป็นผลตอบ ได้แก่ learner entity, coach, evaluation และ delivery
2. ที่จัดเก็บ (stores) หมายถึง องค์ประกอบที่ไม่มีการเปลี่ยนแปลงด้วยตนเอง (passive component) ทำหน้าที่เป็นแหล่งจัดเก็บข้อมูลสารสนเทศ ได้แก่ learner records และ learning resources
3. การไหล (flow) หมายถึง การส่งผ่านของข้อมูลระหว่างองค์ประกอบต่างๆของระบบทั้งในแบบทางเดียวและสองทาง ได้แก่ learning parameters, behavior, assessment information, learner information ทั้งสามส่วน, query, catalog info, locator ทั้งสองส่วน, learning content, multimedia, interaction context

การนิยามองค์ประกอบต่างๆภายในโครงสร้างของ LTSA มีรายละเอียดดังนี้

- Learner Entity หมายถึง กระบวนการที่เป็นนามธรรมของคนที่เป็นผู้เรียน ทั้งในแบบที่เป็นบุคคลเดียวและกลุ่มของบุคคลที่เรียนรู้แยกกัน ที่เรียนรู้ร่วมกัน ที่เรียนรู้ด้วยเป้าหมายที่ต่างกัน
- Learning Parameters หมายถึง การไหลของข้อมูลที่แลกเปลี่ยนระหว่างกระบวนการ learner entity และ coach
- Behavior หมายถึง การไหลของข้อมูลที่แสดงถึงกิจกรรมและการกระทำต่างๆขององค์ประกอบ learner entity ไปสู่กระบวนการ evaluation เพื่อใช้ในช่วงขั้นตอนการประเมิน
- Evaluation หมายถึง กลไกแบบนามธรรมที่แสดงถึงการประเมินองค์ประกอบ learner entity
- Learner Information หมายถึง การไหลของข้อมูลต่างๆที่เกี่ยวข้องกับองค์ประกอบ learner entity
- Learner Records หมายถึง แหล่งจัดเก็บและบันทึกข้อมูลต่างๆที่เกี่ยวข้องกับองค์ประกอบ learner entity ซึ่งมาจากกระบวนการ evaluation, coach และ learner entity เอง โดยเป็นข้อมูลในอดีต ปัจจุบัน และอนาคต
- Assessment หมายถึง การไหลของข้อมูลจากกระบวนการ evaluation ซึ่งแสดงถึงสถานะขององค์ประกอบ learner entity ไปสู่กระบวนการ coach เพื่อใช้ในหาแนวทางที่เหมาะสมในการให้ความรู้
- Coach หมายถึง กระบวนการที่เป็นนามธรรมของการให้ความรู้แก่องค์ประกอบ learner entity
- Query หมายถึง การไหลของข้อมูลในรูปของเงื่อนไขจากองค์ประกอบ coach ไปสู่ learning resources เพื่อค้นหาเนื้อหาสำหรับการเรียนรู้
- Learning Resources หมายถึง แหล่งจัดเก็บและรวบรวมเนื้อหาต่างๆ เช่น บทเรียน ตัวอย่าง การวัดผล เป็นต้น ซึ่งองค์ประกอบ coach สามารถเลือกมาใช้ในการระหว่างการเรียนการสอน
- Catalog Information หมายถึง การไหลของข้อมูลที่แสดงถึงผลจากการค้นหาตามเงื่อนไขที่ระบุไว้ในองค์ประกอบ query
- Locator หมายถึง การไหลของข้อมูลที่แสดงถึงตำแหน่งของเนื้อหาที่ใช้ในการระหว่างการเรียนการสอนจากองค์ประกอบ coach ไปสู่กระบวนการ delivery

- Learning Content หมายถึง การไหลของเนื้อหาที่องค์ประกอบ coach เลือกมาสำหรับสร้างประสบการณ์ในการเรียนรู้
- Delivery หมายถึง กระบวนการและกลไกที่เป็นนามธรรม ซึ่งทำหน้าที่เป็นสื่อในการส่งผ่านเนื้อหาจากองค์ประกอบ learning resources ไปสู่ learner entity
- Interaction Context หมายถึง การไหลของข้อมูลเกี่ยวกับเนื้อหาของ learning content จากองค์ประกอบ delivery ไปสู่กระบวนการ evaluation เพื่อใช้ในการตีความ behavior
- Multimedia หมายถึง การไหลของเนื้อหาที่ใช้ในการเรียนการสอนจากองค์ประกอบ delivery ไปสู่กระบวนการ learner entity ในรูปแบบของสื่อที่ต่าง ๆ กัน เช่น ภาพเคลื่อนไหว เสียง ภาพกราฟฟิก เป็นต้น

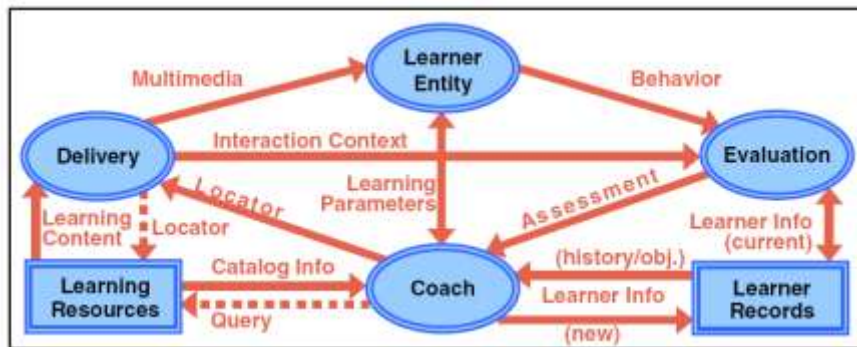
การพัฒนาองค์ประกอบต่างๆของเทคโนโลยี LTSA จะขึ้นอยู่กับวัตถุประสงค์ของระบบสารสนเทศเพื่อการเรียนการสอน ซึ่งในหลายระบบอาจมีการสร้างองค์ประกอบเพียงบางส่วนเท่านั้น เช่น ระบบการเรียนการสอนสำหรับการให้ประกาศนียบัตรจะมีโครงสร้างดังในรูปที่ 2.2 โดยมีเฉพาะองค์ประกอบของ learner records, evaluation และ coach ในการทำหน้าที่เลือกแบบทดสอบที่เหมาะสม และส่งให้ผู้เรียนทำการประเมินและบันทึกลงในฐานข้อมูล



รูปที่ 2.2 โครงสร้าง LTSA สำหรับการประเมินเพื่อประกาศนียบัตร

การพัฒนาระบบดังกล่าวนี้ มีเป้าหมายหลักในการระบุข้อกำหนดที่ชัดเจนในการตรวจประเมิน และระบบโครงสร้าง การอ้างอิง การเข้าถึงฐานข้อมูลที่ทำหน้าที่เก็บบันทึกข้อมูลของผู้เรียน ในขณะที่รูปแบบและรายละเอียดของการตรวจประเมินจะเป็นเป้าหมายที่รองลงมา

สำหรับระบบการเรียนการสอนทางไกล จะเป็นระบบสารสนเทศที่จำเป็นต้องใช้องค์ประกอบต่างๆของเทคโนโลยี LTSA อย่างครบถ้วนที่สุด ดังแสดงในรูปที่ 2.3

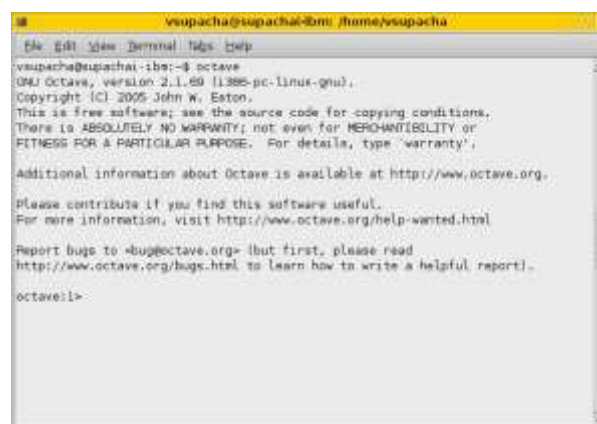


รูปที่ 2.3 โครงสร้าง LTSA สำหรับระบบการเรียนการสอนทางไกล

เป้าหมายหลักในการพัฒนาระบบจะมุ่งเน้นที่การระบุข้อกำหนด วิธีการ และการตีความสำหรับการตรวจสอบพฤติกรรมของผู้เรียน ซึ่งจะต้องสอดคล้องกับกระบวนการเลือกเนื้อหาที่เหมาะสมจากเนื้อหาที่รวบรวมไว้ใน learning resources และส่งผ่านกระบวนการ delivery ไปสู่ผู้เรียน ในขณะที่ ข้อกำหนดและกระบวนการประเมินผลจะเป็นเป้าหมายที่รองลงมา

2.2 การพัฒนาซอฟต์แวร์บน GNU Octave

โปรแกรม GNU Octave เป็นซอฟต์แวร์ที่ใช้ในการคำนวณทางด้านเมตริกซ์ ซึ่งมีการเพิ่มเติมคำสั่งสำหรับการคำนวณเวกเตอร์และโพลิโนเมียล การหาคำตอบของสมการเชิงเส้น การหาคำตอบของสมการอนุพันธ์ การควบคุมการไหลของคำสั่งต่างๆ รวมทั้งสามารถสร้างรูปภาพได้ทั้งแบบ 2 มิติและ 3 มิติ การทำงานโดยพื้นฐานของโปรแกรม GNU Octave จะเป็นแบบโต้ตอบทันที (interactive) กล่าวคือเมื่อพิมพ์คำสั่งเสร็จก็จะประมวลผลแล้วแสดงคำตอบทันที โปรแกรม GNU Octave สามารถสร้างคำสั่งเพิ่มเติมในรูปของไฟล์ที่มีนามสกุล .m ด้วยการใส่ชุดคำสั่งพื้นฐานร่วมกับคำสั่งควบคุมการไหลของโปรแกรมเช่น if-else while และ for เป็นต้น ระบบติดต่อผู้ใช้ของโปรแกรม GNU Octave ซึ่งทำงานบนระบบปฏิบัติการลินุกซ์ และ Microsoft Windows ถูกแสดงไว้ในรูปที่ 2.4 และ 2.5 ตามลำดับ สำหรับรายละเอียดในการใช้งานโปรแกรม GNU Octave สามารถศึกษาได้จากคู่มือของโปรแกรม GNU Octave ฉบับภาษาไทยในสื่อบันทึกข้อมูลที่แนบมา กับรายงานฉบับนี้



รูปที่ 2.4 ระบบติดต่อผู้ใช้ของโปรแกรม GNU Octave บนลินุกซ์



รูปที่ 2.5 ระบบติดต่อผู้ใช้ของโปรแกรม GNU Octave บน Microsoft Windows

2.2 การพัฒนาซอฟต์แวร์ด้วยภาษา Python

Python [Martelli, 2003] เป็นภาษาคอมพิวเตอร์ระดับสูงมากที่ทำงานแบบแปลทีละคำสั่ง (interpreted language) โดยมีจุดเด่นจากการมีชนิดข้อมูลระดับสูง เช่น list และ dictionary เป็นต้น การใช้ย่อหน้าในการจำแนกระดับและขอบเขตของการประมวลผล และการที่ไม่จำเป็นต้องประกาศชนิดของตัวแปร ซอฟต์แวร์ทำหน้าที่เป็นตัวแปลภาษา Python ที่ใช้อย่างแพร่หลายในปัจจุบัน ได้แก่ Classic Python หรือ Python ซึ่งพัฒนาขึ้นด้วยภาษาซี และ Jython ซึ่งพัฒนาขึ้นด้วยภาษาจาวา การติดตั้งซอฟต์แวร์ตัวแปลภาษา Python สามารถทำได้ง่ายมากสำหรับระบบปฏิบัติการลินุกซ์ เนื่องจากมักถูกติดตั้งเป็นซอฟต์แวร์พื้นฐานอยู่แล้ว ในขณะที่การใช้งานบนระบบปฏิบัติการ Microsoft Windows สามารถดาวน์โหลดโปรแกรม ActivePython จากเว็บเพจ <http://www.activestate.com/Products/ActivePython/> ที่รวมโปรแกรมอำนวยความสะดวกในการติดตั้งมาด้วย



รูปที่ 2.6 การติดตั้งและใช้งานโปรแกรม ActivePython

การพัฒนาซอฟต์แวร์ด้วยภาษา Python จำเป็นต้องใช้โปรแกรมสำหรับแก้ไขไฟล์ตัวอักษร เพื่อเขียนสคริปต์ในการประมวลผลลงในไฟล์ จากนั้นจึงใช้โปรแกรมตัวแปลภาษาก็คือ python ในลินุกซ์ หรือ python.exe ในวินโดวส์ เพื่อประมวลผลแต่ละคำสั่งในไฟล์นั้น การเรียกใช้โปรแกรมตัวแปลภาษาโดยไม่ระบุไฟล์สคริปต์จะทำให้เข้าสู่สถานะการแปลแบบตอบโต้ ซึ่งโปรแกรมจะแสดงเครื่องหมายรอการสั่งการจากผู้ใช้ เมื่อผู้ใช้ป้อนนิพจน์หรือคำสั่ง โปรแกรมก็จะประมวลผลเฉพาะนิพจน์หรือคำสั่งนั้นและแสดงผลออกมาทางหน้าจอ ดังในรูปที่ 2.7

```

Python Shell
Python 2.4.1 (805, Mar 30 2005, 09:13:57) [MSC v.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

=====
Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface.  This connection is not visible on any external
interface and no data is sent to or received from the Internet.
=====

IDLE 1.1.1
>>> for i in range(3):
>>>     print 'i is %d' % i

i is 0
i is 1
i is 2
>>> if i > 1:
>>>     print 'i is greater than 1'
>>> else:
>>>     print 'i is less or equal to 1'

i is less or equal to 1
>>>

```

รูปที่ 2.7 การป้อนคำสั่งให้โปรแกรมตัวแปลภาษา Python ในโหมดตอบโต้

โครงสร้างของโปรแกรมในภาษา Python ประกอบขึ้นจากกลุ่มของบรรทัดแบบตรรกะ ซึ่งอาจแต่ละบรรทัดแบบตรรกะประกอบขึ้นจากบรรทัดจริงหรือกลุ่มของบรรทัด สัญลักษณ์ # ซึ่งไม่อยู่ในขอบเขตของเครื่องหมายข้อความ แสดงถึงการเริ่มต้นของคำอธิบาย (comment) ซึ่งโปรแกรมตัวแปลภาษาจะละเลยเช่นเดียวกับบรรทัดที่มีเฉพาะช่องว่าง (whitespace) โดยทั่วไป การจบบรรทัดจะแสดงถึงการสิ้นสุดขอบเขตของนิพจน์ โดยสัญลักษณ์ \ (backspace) สามารถใช้ในการเชื่อมบรรทัดเข้าด้วยกัน ทั้งนี้ หากขอบเขตของเครื่องหมายล้อมแบบต่างๆ ได้แก่ ([และ { ยังไม่ครบ โปรแกรมจะเชื่อมบรรทัดเข้าด้วยกันโดยอัตโนมัติ และถือเสมือนเป็นหนึ่งบรรทัดในแบบตรรกะ ภาษา Python ใช้การย่อหน้าแสดงถึงกลุ่มของคำสั่งในระดับเดียวกัน ทำให้นิพจน์หรือคำสั่งที่จะถูกประมวลผลอย่างต่อเนื่อง ต้องมีการย่อหน้าด้วยระดับที่เท่ากัน ซึ่งโดยทั่วไป 4 ช่องว่างเท่ากับ 1 ย่อหน้า

โปรแกรมตัวแปลภาษาจะจำแนกแต่ละบรรทัดแบบตรรกะออกเป็น token ซึ่งสามารถจำแนกได้เป็น identifier, keyword, operator, delimiter และ literal แบบขึ้นอยู่กับการพิมพ์ (case sensitive) Identifier คือ token ที่ใช้แทนชื่อในการจำแนกตัวแปร ฟังก์ชัน คลาส หรือวัตถุ โดยเริ่มต้นด้วยตัวอักษรหรือสัญลักษณ์ _ (underscore) และตามด้วยตัวอักษร สัญลักษณ์ underscore หรือตัวเลข การขึ้นต้นชื่อของวัตถุด้วย _ (1 underscore) หมายถึงการเป็นวัตถุแบบ private และ __ (2 underscore) หมายถึงการเป็นวัตถุแบบ private แบบแข็ง การขึ้นต้นและจบ identifier ด้วย __ (2 underscore) หมายถึงชื่อเฉพาะที่นิยามเป็นพิเศษในภาษา Keyword เป็น identifier เฉพาะที่ภาษา Python จำกัดไว้ โดยประกอบด้วย 28 คำ ได้แก่ and assert break class continue def del elif else except exec finally for from global if import in is lambda not or pass print raise return try while และ yield โดยใช้เฉพาะตัวอักษรตัวพิมพ์เล็ก Operator หมายถึง token ของตัวกระทำ ซึ่งแทนด้วยสัญลักษณ์ที่ไม่ใช่ตัวอักษรหรือตัวเลข เช่น + - * / เป็นต้น Delimiter หมายถึงสัญลักษณ์ที่ใช้บอกขอบเขตของพจน์ต่างๆ เช่น ([{ += -= เป็นต้น Literal คือข้อมูลที่ปรากฏในนิพจน์ เช่น 120 2.0 'text' เป็นต้น โดยการนำ token แบบ literal ร่วมกับ delimiter จะกำหนดข้อมูลในรูปแบบอื่นได้ เช่น [1,2] ซึ่งเป็นข้อมูลแบบ list

ข้อมูลในภาษา Python ถูกกำหนดในรูปของวัตถุ โดยแบ่งได้เป็น 2 กลุ่มหลัก คือ mutable ซึ่งสามารถแก้ไขได้ และ immutable ซึ่งไม่สามารถแก้ไขได้ โดยการกระทำใดๆต่อวัตถุแบบ immutable จะทำให้เกิดวัตถุใหม่ขึ้น ข้อมูลพื้นฐานใน Python ได้แก่

1. ตัวเลข ที่สามารถอยู่ในรูปของจำนวนเต็ม จำนวนทศนิยม และจำนวนเชิงซ้อนซึ่งประกอบจากตัวเลข 2 จำนวน และใช้ j หรือ J ตามหลัง ในการระบุว่าเป็นตัวเลขของส่วนจินตภาพ เช่น $12.2.33$ และ $1 + 2i$ เป็นต้น
2. Sequence ในภาษา Python หมายถึงชนิดของข้อมูลที่เกิดจากการรวมข้อมูลแบบต่างๆกันในแบบเรียงลำดับ โดยอ้างอิงด้วยจำนวนที่ไม่เป็นลบ ข้อมูลแบบ sequence ได้แก่ string ซึ่งแทนข้อมูลแบบข้อความ และ tuple, list ซึ่งรวมข้อมูลแบบต่างๆ โดย tuple เป็นวัตถุแบบ immutable และ list เป็นวัตถุแบบ mutable ข้อมูลแบบ dictionary เป็นการรวมวัตถุที่อ้างอิง เช่น 'test' ซึ่งเป็นข้อมูลแบบ string $(1, 1.1, 'a')$ ซึ่งเป็นข้อมูลแบบ tuple และ $[2, 'a', 2.1]$ ซึ่งเป็นข้อมูลแบบ list การอ้างอิงสมาชิกใน sequence ใดๆใช้การระบุตัวเลขอ้างอิงภายในวงเล็บ $[]$ โดยเริ่มต้นด้วย 0 สำหรับสมาชิกตัวแรก เช่น $A = [1, 2, 3, 4]$ และ $A[1]$ เท่ากับ 2
3. Dictionary เป็นชนิดของข้อมูลที่รวมข้อมูลแบบต่างๆ โดยอนุญาตให้ใช้ตัวอ้างอิงใดๆในการระบุถึงสมาชิก เช่น $d = \{1: 'AAA', 2: 'BBB', 'A': 1.1, 'B': 2.2\}$ ซึ่งเป็นการระบุถึง dictionary ที่มีสมาชิก 4 ตัว และใช้ 1, 2, 'A' และ 'B' เป็นตัวอ้างอิงถึงสมาชิก การอ้างอิงสมาชิกใน dictionary ใดๆใช้การระบุตัวอ้างอิงภายในวงเล็บ $[]$ เช่น $d[1]$ เท่ากับ 'AAA' และ $d['B']$ เท่ากับ 2.2 ตามลำดับ
4. ชนิดข้อมูลแบบพิเศษ ได้แก่ None ซึ่งหมายถึงวัตถุแบบไม่มีตัวตน สำหรับการระบุว่าไม่มีวัตถุดังกล่าว ชนิดข้อมูลแบบตรรกะ เช่น True และ False และชนิดข้อมูลแบบ callable ซึ่งหมายถึงวัตถุที่เป็นโค้ด เช่น ฟังก์ชัน และคลาส เป็นต้น

โครงสร้างในของภาษา Python ที่เกี่ยวข้องกับลำดับการประมวลผลในการทำงาน ประกอบด้วยคำสั่ง if สำหรับการตัดสินใจ คำสั่ง for และ while สำหรับการซ้ำ และคำสั่ง try สำหรับการดักสถานการณ์ของการประมวลผลที่ผิดปกติ โครงสร้างของภาษาจะใช้การย่อหน้าเพื่อบอกขอบเขตในการประมวลผลตามลำดับ ดังตัวอย่างการป้อนคำสั่งในรายการที่ 2.1

```
for i in range(10):
    # print the value of i in each loop using format string
    print 'i is %d' %i

if i > 3:
    print 'i is greater than 3'
else:
    print 'i is less or equal to 3'
```

รายการที่ 2.1 การประมวลผลด้วยคำสั่ง for และ if ในภาษา Python

การประมวลผลแบบซ้ำด้วยคำสั่ง for จะอาศัยข้อมูลประเภท sequence หรือการใช้ iterator ซึ่งเป็นกลไกในการสร้างตัวอ้างอิง เพื่อสร้างตัวอ้างอิงในการประมวลผลแต่ละวงรอบ

2.3 การพัฒนาส่วนติดต่อผู้ใช้ด้วย wxPython

wxPython [wxPython, Online] เป็นเครื่องมือซอฟต์แวร์สำหรับสนับสนุนการพัฒนาแบบติดต่อผู้ใช้แบบกราฟิกด้วยภาษา Python ในแบบการโปรแกรมเชิงวัตถุ ซอฟต์แวร์ wxPython พัฒนาต่อยอดจากโค้ดภาษา C++ ของเครื่องมือซอฟต์แวร์ wxWidgets [wxWidgets, Online] ซึ่งใช้สำหรับการพัฒนาแบบติดต่อผู้ใช้

แบบกราฟฟิกด้วยภาษา C++ การประยุกต์ใช้ wxPython ในการพัฒนาซอฟต์แวร์ที่มีส่วนติดต่อผู้ใช้แบบกราฟฟิกจะมีขั้นตอนในการทำงานดังนี้

1. ร้องขอให้โปรแกรมตัวแปลภาษา Python ดึงข้อมูลเกี่ยวกับคลาสและวิธีการต่างๆที่รวมไว้ในแพ็คเกจ wx ซึ่งเป็นแพ็คเกจหลักของ wxPython
2. สร้างวัตถุที่สืบทอดจากคลาส wx.App
3. สร้างวัตถุของส่วนติดต่อผู้ใช้แบบกราฟฟิกต่างๆ เช่น หน้าต่าง เมนู เป็นต้น
4. เขียนโค้ดเพื่อรองรับเหตุการณ์ต่างๆ เช่น การทำงานเมื่อผู้ใช้เลือกเมนู เป็นต้น
5. เริ่มต้นการทำงานของวงรอบรอเหตุการณ์ (event loop) เพื่อตอบสนองต่อเหตุการณ์ที่ผู้ใช้กระทำต่อส่วนติดต่อผู้ใช้ต่างๆ เช่น เหตุการณ์ที่ผู้ใช้กดปุ่มบนหน้าต่าง เป็นต้น

รายการที่ 2.2 และรูปที่ 2.8 แสดงโครงสร้างของซอฟต์แวร์ที่ประยุกต์ใช้ฟังก์ชันการทำงานของ wxPython ในการสร้างหน้าต่างย่อยที่มีข้อความ “Hello from wxPython” และผลที่ได้รับ

```
import wx

class MyApp(wxApp):
    def OnInit(self):
        frame = wxFrame(NULL, -1, "Hello from wxPython")
        frame.Show(true)
        self.SetTopWindow(frame)
        return true

app = MyApp(0)
app.MainLoop()
```

รายการที่ 2.2 การสร้างหน้าต่างของโปรแกรมด้วย wxPython



รูปที่ 2.8 หน้าต่างที่สร้างจาก wxPython

2.4 การพัฒนาความสามารถในการสื่อสารผ่านเครือข่ายคอมพิวเตอร์ด้วย Twisted

Twisted [Shtull-Trauring, Online] เป็นเครื่องมือซอฟต์แวร์สำหรับพัฒนาความสามารถในการติดต่อสื่อสารผ่านทางเครือข่ายคอมพิวเตอร์ด้วยภาษา Python จุดเด่นของ Twisted คือ การเป็นซอฟต์แวร์แบบเปิดเผยต้นฉบับ การสนับสนุนการทำงานทั้งแบบไคลเอนต์และเซิร์ฟเวอร์ และการทำงานบนฮาร์ดแวร์และระบบปฏิบัติการที่หลากหลาย เป้าหมายของ Twisted คือการเป็นกรอบของการพัฒนาที่สามารถรองรับความต้องการของซอฟต์แวร์สมัยใหม่ ซึ่งต้องการสื่อสารข้อมูลผ่านเครือข่าย ด้วยโปรโตคอลที่หลากหลาย การทำงานของ Twisted อาศัยการนิยามการสื่อสารผ่านเครือข่ายคอมพิวเตอร์ ด้วยนามธรรมแบบหลายลำดับชั้น โดยลำดับชั้นล่างสุดที่ขึ้นอยู่กับแพลตฟอร์ม เป็นวงรอบรอรับเหตุการณ์ (event loop) ต่างๆ และเป็นโค้ดที่ไม่

จำเพาะกับแพลตฟอร์มคอมพิวเตอร์ในลำดับขั้นที่สูงขึ้น สำหรับการสื่อสาร โปรโตคอล และกรอบของเครือข่าย Twisted แบ่งเป็นหลายแพ็คเกจ ซึ่งแต่ละแพ็คเกจจะให้ทำงานกับบริการของเครือข่ายที่แตกต่างกัน และการทำงานของแพ็คเกจในระดับสูงจะสร้างขึ้นจากแพ็คเกจต่างๆในระดับล่าง

แพ็คเกจ `twisted.internet` เป็นแกนหลักในการทำงานของ Twisted ด้วยการสร้างวงรอบเพื่อรอรับเหตุการณ์ ซึ่งเมื่อเกิดเหตุการณ์ของเครือข่าย (network event) เช่น การสื่อสารผ่านซ็อกเก็ต ก็จะมีการสร้างเชรดสำหรับทำหน้าที่ประมวลผลหรือส่งผ่านเหตุการณ์นั้น ทั้งนี้ แพ็คเกจ `twisted.internet` ได้นิยามเหตุการณ์ของเครือข่ายต่างๆ ด้วยชุดคำสั่งหรือข้อมูลที่ไม่จำเพาะ ทำให้สามารถเรียกใช้งานบนแพลตฟอร์มคอมพิวเตอร์ที่ต่างกันได้ การทำงานของชุดคำสั่งในระดับบนสำหรับการสื่อสารด้วยโปรโตคอลที่เป็นพื้นฐานของเครือข่ายอินเทอร์เน็ต เช่น TCP, SSL และ UDP จะทำงานอยู่บนวงรอบรอเหตุการณ์ของแพ็คเกจ `twisted.internet` อีกทีหนึ่ง การสร้างโปรโตคอลในระดับบนจะแยกออกจากเลเยอร์ transport ทำให้โปรโตคอลสามารถทำงานได้บนองค์ประกอบ transport ที่เป็นประเภทเดียวกันได้ เช่น การสร้างโปรโตคอล TCP จะสามารถรองรับการเข้ารหัส SSL ได้โดยไม่ต้องแก้ไข การประยุกต์ Twisted สำหรับสร้างซอฟต์แวร์แบบไคลเอนต์ จะได้ความสามารถในระดับสูง เช่น การสื่อสารใหม่เมื่อเกิดความผิดพลาด การหมดเวลาระหว่างรอการติดต่อ เป็นต้น

แพ็คเกจ `twisted.protocols` เป็นการรวบรวมโปรโตคอลในการสื่อสารบนเครือข่ายคอมพิวเตอร์ที่ใช้กันทั่วไป ซึ่งทำงานได้ทั้งแบบเป็นไคลเอนต์และเซิร์ฟเวอร์ เช่น HTTP, FTP, DNS เป็นต้น ในขณะที่แพ็คเกจ `twisted.web` และ `twisted.names` ให้ชุดคำสั่งสำหรับการทำงานในระดับที่สูงขึ้น กล่าวคือ การสร้างเว็บเซิร์ฟเวอร์ในแบบวัตถุ และการสร้างเซิร์ฟเวอร์ DNS ตามลำดับ

แพ็คเกจ `twisted.cred` เป็นแพ็คเกจที่สนับสนุนการสร้างซอฟต์แวร์บนเซิร์ฟเวอร์ ด้วยการลดความซ้ำซ้อนของโค้ดที่ใช้ในส่วนต่างๆของกระบวนการทำงานของเซิร์ฟเวอร์ เช่น การร้องขอการอนุญาต ซึ่งจำเป็นต้องใช้ในขั้นตอนที่ต่างกันไป เป็นต้น แพ็คเกจ `twisted.cred` รวบรวมคลาสและส่วนเชื่อมต่อแบบต่างๆ เพื่อให้สามารถแยกหน้าที่รับผิดชอบในการพัฒนาระบบซอฟต์แวร์สำหรับทำงานบนเซิร์ฟเวอร์ออกจากกันได้

1. Protocol developer สร้างส่วนของโปรโตคอล สำหรับติดต่อกับองค์ประกอบด้านหลังที่ถูกมองเป็นนามธรรม
2. Application developer สร้างองค์ประกอบด้านหลังสำหรับรองรับโปรโตคอล ซึ่งอาจเป็นการทำงานของฐานข้อมูล หรือระบบอีเมล เป็นต้น
3. Authorization developer สร้างส่วนตรวจสอบการอนุญาตสำหรับการทำงานในขั้นตอนต่างๆ
4. Deployment เชื่อมต่อองค์ประกอบข้างต้นเข้าด้วยกัน

แพ็คเกจ `twisted.application` สนับสนุนการสร้างเซิร์ฟเวอร์ในระดับที่ซับซ้อน ด้วยการให้นิยามบริการต่างๆของเซิร์ฟเวอร์ในรูปแบบ tree นอกจากนี้ การติดตั้งเซิร์ฟเวอร์ยังสามารถทำได้ง่ายดาย ด้วยการสร้างสคริปต์ด้วยภาษา Python และเรียกให้ทำงานผ่านโปรแกรม `twistd` ซึ่งมีความสามารถในการบันทึกสถานะ ทำงานในแบบเบื้องหลัง และการประเมินประสิทธิภาพ

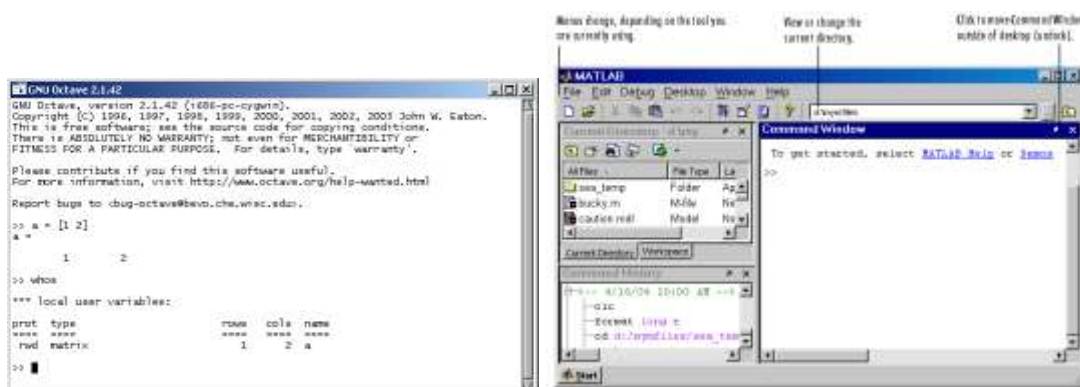
บทที่ 3

การออกแบบซอฟต์แวร์

3.1 การวิเคราะห์ปัญหา

เป้าหมายหลักของงานวิจัยนี้ที่ต้องการประยุกต์ใช้โปรแกรม GNU Octave เพื่อสนับสนุนการเรียนการสอนทางด้านวิศวกรรมไฟฟ้า คือการใช้เป็นเครื่องมือช่วยในการศึกษาเนื้อหาของทฤษฎีต่างๆ ที่มีการคำนวณที่ซับซ้อน โดยเฉพาะการคำนวณที่เกี่ยวข้องกับเวกเตอร์ เมทริกซ์ และการหาคำตอบของสมการอนุพันธ์ ซึ่งเป็นรูปแบบของการคำนวณทางคณิตศาสตร์ที่พบมากในสาขาวิศวกรรมไฟฟ้า รูปแบบของการคำนวณดังกล่าวไม่เหมาะสมกับการใช้เครื่องคำนวณ หรือแม้แต่การใช้งานซอฟต์แวร์ซึ่งมีความสามารถทางด้านการคำนวณทั่วไป เช่น โปรแกรม Microsoft Excel เนื่องจากข้อมูลที่มีตัวเลขจำนวนมากและความซับซ้อนในการตั้งคำนวณสมการต่างๆ ในระหว่างการแก้ปัญหา หากพิจารณาถึงรูปแบบการคำนวณดังกล่าว โปรแกรม GNU Octave มีข้อได้เปรียบเช่นเดียวกับโปรแกรม MATLAB ในแง่ที่เริ่มต้นจากการเป็นซอฟต์แวร์สำหรับคำนวณเวกเตอร์และเมทริกซ์ และรองรับการพัฒนาฟังก์ชันต่างๆ เพิ่มเติมด้วยภาษาคอมพิวเตอร์ในแบบภาษาแปล (interpreted language) สิ่งที่เป็นข้อได้เปรียบอย่างชัดเจนของโปรแกรม GNU Octave เมื่อเทียบกับโปรแกรม MATLAB คือ การเป็นซอฟต์แวร์แบบเปิดเผยต้นฉบับที่ไม่มีค่าใช้จ่ายในแง่ของราคาซอฟต์แวร์ จึงเหมาะกับการเป็นซอฟต์แวร์สำหรับแจกจ่ายให้นักศึกษาหรือผู้ที่สนใจได้นำไปใช้ประกอบการเรียนรู้ทางด้านวิศวกรรมไฟฟ้า

โปรแกรม GNU Octave ยังมีข้อจำกัดบางประการที่มักพบเมื่อนำไปใช้ในกลุ่มของนักศึกษา ได้แก่ เวลาที่ใช้ในการเรียนรู้วิธีใช้งานในช่วงแรกที่ยาวนาน เนื่องจากการใช้รูปแบบของการป้อนคำสั่งในการประมวลผล ดังนั้น ผู้ใช้โปรแกรมจะต้องมีความเข้าใจในวิธีการป้อนคำสั่งและไวยากรณ์ของภาษารวมทั้งจดจำรูปแบบของคำสั่งได้ บริษัท Mathworks ซึ่งเป็นผู้พัฒนาโปรแกรม MATLAB เองได้พยายามแก้ไขข้อจำกัดนี้ ด้วยการสร้างส่วนติดต่อผู้ใช้แบบกราฟิกตั้งแต่โปรแกรมในรุ่น 6.0 เป็นต้นมา โดยมีหน้าต่างสำหรับอำนวยความสะดวกในระหว่างการทำงาน



รูปที่ 3.1 การเปรียบเทียบส่วนติดต่อผู้ใช้ของโปรแกรม GNU Octave และ MATLAB

ในขณะที่กลุ่มของผู้พัฒนาอิสระซึ่งเป็นผู้รับผิดชอบต่อการปรับปรุงโปรแกรม GNU Octave ในแง่ที่เป็นซอฟต์แวร์แบบเปิดเผยต้นฉบับ ได้เน้นเป้าหมายหลักในการพัฒนาอัลกอริทึมสำหรับคำนวณและประมวลผลที่เป็นส่วนแกนของโปรแกรม GNU Octave โดยยังไม่มีมีการปรับปรุงส่วนติดต่อผู้ใช้เพิ่มเติม แม้แต่การสั่งการทางเมนู ทำให้ใช้งานได้ยากสำหรับผู้เริ่มต้น

โดยทั่วไป การสอนวิธีใช้งานโปรแกรมคอมพิวเตอร์หรือการประยุกต์ใช้ซอฟต์แวร์เพื่อแก้ปัญหาต่างๆ มักจะกระทำในห้องคอมพิวเตอร์ โดยผู้สอนจะใช้เครื่องฉายภาพเพื่อแสดงถึงวิธีการใช้งานแต่ละขั้นตอน แล้วให้ผู้เรียนทำตาม การสอนด้วยวิธีดังกล่าวมีความเหมาะสมกับซอฟต์แวร์ที่มีส่วนติดต่อผู้ใช้แบบกราฟฟิก ซึ่งใช้การสั่งการผ่านทางเมนูและการใช้เมาส์เพื่อคลิกเลือกที่องค์ประกอบต่างๆ บนหน้าต่าง การสอนวิธีการใช้งานโปรแกรม GNU Octave ร่วมกับเครื่องฉายภาพในห้องคอมพิวเตอร์ มักพบปัญหาจากรูปแบบการสั่งงานแบบตอบโต้ ซึ่งนักศึกษาจะใช้เวลาค่อนข้างมาก เพราะไม่คุ้นเคยกับวิธีสั่งการแบบพิมพ์คำสั่ง รวมทั้งผู้สอนเองอาจต้องเดินมาที่นักศึกษา เพื่อมาชี้ถึงข้อผิดพลาดในการพิมพ์คำสั่ง ซึ่งบางครั้งอาจเป็นปัญหาเล็กน้อยแค่การพิมพ์เครื่องหมายวงเล็บ ทำให้การเรียนการสอนดำเนินไปได้อย่างช้า โดยเฉพาะการสอนช่วงเริ่มต้น นอกจากนี้ การส่งผ่านเนื้อหาจากผู้สอนถึงผู้เรียนก็ทำได้เฉพาะผ่านทางบรรยายและการใช้สื่อประเภทเครื่องฉายภาพ และการที่ผู้เรียนจะส่งผลลัพธ์จากการเรียนรู้หรือตอบโจทย์ที่ผู้สอนกำหนดให้ ก็จำเป็นต้องอาศัยสื่อบันทึกข้อมูลที่ผู้สอนจำเป็นต้องไปเปิดอ่านแยกแต่ละผู้เรียนในภายหลัง ข้อจำกัดดังกล่าวเป็นสิ่งที่มักพบในซอฟต์แวร์ ซึ่งไม่ได้พัฒนาขึ้นเพื่อวัตถุประสงค์ของการเป็นซอฟต์แวร์สนับสนุนการเรียนรู้

สำหรับงานวิจัยนี้ จะเน้นการเพิ่มความเหมาะสมให้กับโปรแกรม GNU Octave ในแง่ของการเป็นเครื่องมือช่วยสำหรับการเรียนการสอนทางด้านวิศวกรรมไฟฟ้า การวิเคราะห์ข้อจำกัดของโปรแกรม GNU Octave ข้างต้นนี้ พบว่าความจำเป็นเริ่มต้นคือการเพิ่มความสามารถในแง่ของการเป็นซอฟต์แวร์เพื่อสนับสนุนการเรียนการสอน โดยเน้นการเรียนการสอนในห้องคอมพิวเตอร์ ซึ่งสามารถแบ่งเป็นประเด็นต่างๆ ดังนี้

1. การสื่อสารระหว่างผู้เรียนและผู้สอนผ่านทางเครือข่ายคอมพิวเตอร์ เกี่ยวกับสถานะของการเรียนรู้ของผู้เรียน เนื้อหาที่ใช้ประกอบ และงานที่ผู้สอนกำหนด
2. ส่วนติดต่อผู้ใช้แบบกราฟฟิก ซึ่งอำนวยความสะดวกในการป้อนคำสั่ง
3. การจัดเก็บไฟล์ต่างๆ ที่เกี่ยวข้องกับการเรียนการสอน

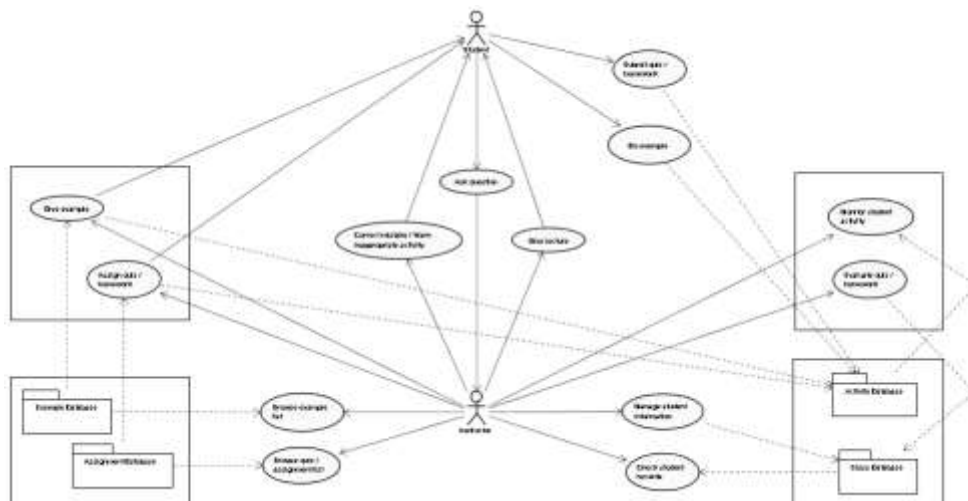
การพัฒนาความสามารถเพิ่มเติมสำหรับโปรแกรม GNU Octave จะอิงอยู่บนโครงสร้างของ LTSA งานวิจัยนี้ได้นิยามองค์ประกอบต่างๆ ของการสอนเนื้อหาทางทฤษฎีร่วมกับการใช้โปรแกรม GNU Octave ภายในโครงสร้างของ LTSA ดังนี้

- Learner Entity หมายถึง นักศึกษาหรือผู้เรียน โดยเป็นการเรียนรู้แบบกลุ่มภายในพื้นที่บริเวณห้องคอมพิวเตอร์
- Coach และ Evaluation หมายถึง อาจารย์ผู้สอน ซึ่งบรรยายเกี่ยวกับเนื้อหาทางทฤษฎี วิธีการใช้งานโปรแกรม GNU Octave และการประยุกต์ใช้โปรแกรม GNU Octave ในการแก้ปัญหา
- Learner Records หมายถึง ระบบฐานข้อมูลที่บันทึกข้อมูลต่างๆ ที่เกี่ยวข้องกับ learner entity ได้แก่ กิจกรรมต่างๆ ที่ learner entity ได้กระทำในระหว่างการเรียนการสอน ไฟล์ต่างๆ ที่ผู้เรียนได้ส่งให้ผู้สอน และเนื้อหาต่างๆ ที่ผู้สอนได้กำหนดให้ learner entity
- Learning Resources หมายถึง ระบบฐานข้อมูลที่รวบรวมเนื้อหาต่างๆ เช่น บทเรียน ตัวอย่าง การวัดผล เป็นต้น ซึ่งผู้สอนสามารถเลือกมาใช้ในระหว่างการเรียนการสอน

- เมื่อพิจารณาจากนิยามขององค์ประกอบข้างต้นนี้ ส่วนของกระบวนการและแหล่งจัดเก็บซึ่งสร้างขึ้นด้วยเทคโนโลยีสารสนเทศคือ learner records, learning resources และ delivery ในขณะที่ องค์ประกอบที่เป็นการไหลของข้อมูลระหว่างกระบวนการและแหล่งจัดเก็บ ซึ่งเป็นข้อมูลแบบอิเล็กทรอนิกส์ที่สื่อสารภายในระบบสารสนเทศ ประกอบด้วย

- ### 3.2 การออกแบบซอฟต์แวร์ในภาพรวม

ขั้นตอนเริ่มต้นในการพัฒนาซอฟต์แวร์ใดๆ คือ การวิเคราะห์ถึงข้อกำหนดและรายละเอียดต่างๆของซอฟต์แวร์นั้น โดยพิจารณาจากการตอบโต้ระหว่างซอฟต์แวร์และผู้ใช้โปรแกรม และหน้าที่ของซอฟต์แวร์ในแต่ละขั้นตอนของการประมวลผล ภาพรวมของกิจกรรมต่างๆระหว่างผู้สอน ผู้เรียน และระบบสารสนเทศสนับสนุนการเรียนการสอนภายในห้องคอมพิวเตอร์ เมื่อวิเคราะห์โดยอ้างอิงตามโครงสร้างของ LTSA สามารถเขียนได้ในรูปของแผนภูมิ Use Case ดังแสดงในรูปที่ 3.2

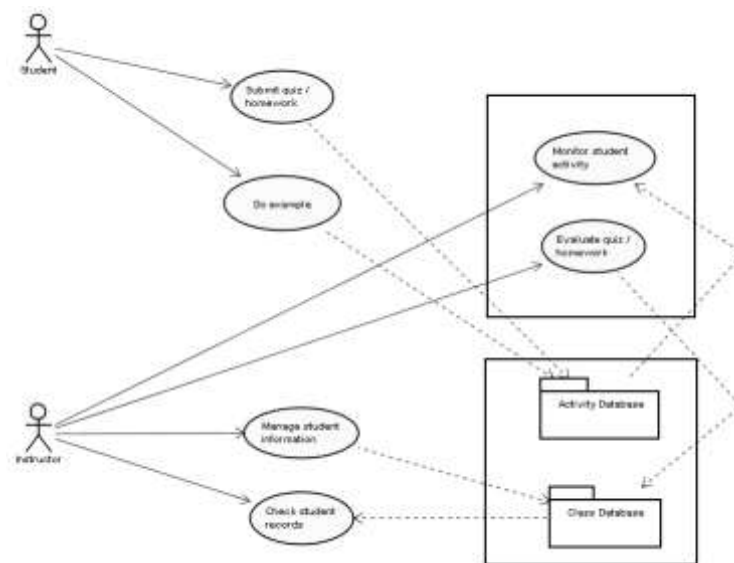


รูปที่ 3.2 การวิเคราะห์ Use Case ของการเรียนการสอนภายในห้อง

16

หรือข้อสงสัยเกี่ยวกับเนื้อหาที่ผู้สอนบรรยาย กิจกรรมนี้อาจกล่าวได้ว่าเป็นกิจกรรมหลักของการให้ความรู้ภายในห้องบรรยายในปัจจุบัน โดยยังไม่ได้พึ่งพาระบบสารสนเทศในการดำเนินกิจกรรม การสอนเนื้อหาทางทฤษฎีรวมกับการใช้งานโปรแกรม GNU Octave จะเป็นการสื่อสารระหว่างผู้สอนและผู้เรียนโดยตรง เนื่องจากขอบเขตของสถานที่จำกัดเฉพาะภายในห้องคอมพิวเตอร์ ทำให้ไม่จำเป็นต้องพัฒนาซอฟต์แวร์ในส่วนนี้

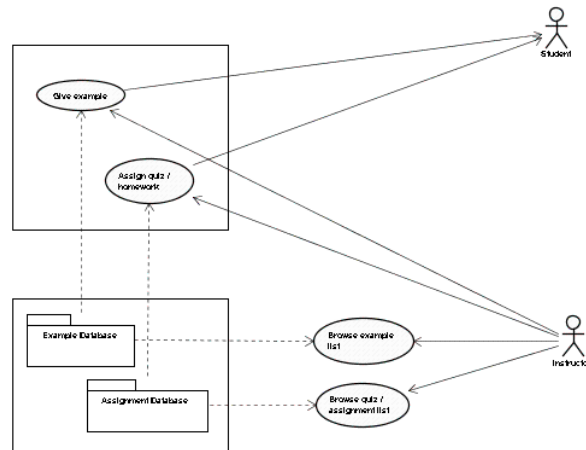
องค์ประกอบ learner records ตามที่ระบุไว้ในแผนภูมิ Use Case เป็นส่วนหนึ่งของระบบสารสนเทศซึ่งเก็บบันทึกข้อมูลต่างๆ ที่เกี่ยวกับผู้เรียน ได้แก่ ไฟล์ต่างๆ ที่เป็นผลของงานที่ผู้สอนกำหนดให้ผู้เรียน นอกจากนี้ การที่ผู้สอนทำหน้าที่ประเมินสถานะการเรียนรู้ของผู้เรียนตามนิยามของกระบวนการ evaluation ทำให้ข้อมูลเกี่ยวกับกิจกรรมและการกระทำต่างๆ ตามนิยามขององค์ประกอบ behavior เช่น คำสั่งต่างๆ ที่ผู้เรียนพิมพ์ป้อนให้กับโปรแกรม GNU Octave จะถูกส่งโดยตรงเข้าสู่องค์ประกอบ learner records เพื่อให้ผู้สอนสามารถนำไปใช้ประกอบการประเมินสถานะของผู้เรียนได้ การออกแบบโครงสร้างของ learner records เพื่อบันทึก learner information มีประเด็นที่ต้องให้ความสำคัญ คือ ผลสัมฤทธิ์ของการสอนแต่ละครั้งจะอยู่ในรูปของไฟล์แบบ .m ซึ่งแสดงถึงการเรียนรู้วิธีพัฒนาคำสั่งด้วยภาษาของโปรแกรม GNU Octave และอาจเป็นไฟล์ที่จำเป็นต้องใช้ในการเรียนการสอนต่อมา ดังนั้น การจัดเก็บข้อมูลขององค์ประกอบ learner records จะต้องมีความเหมาะสมในการจัดเก็บไฟล์แบบตัวอักษร และรองรับการร้องขอไฟล์ที่จัดเก็บไว้จากตัวของผู้เรียนเอง ในขณะที่สามารถควบคุมการเข้าถึงข้อมูลบางส่วนที่จำกัดไว้เฉพาะผู้สอนได้



รูปที่ 3.3 การตอบโต้ระหว่างผู้สอนและผู้เรียนกับองค์ประกอบ learner records

องค์ประกอบ learning resources ภายในแผนภูมิ Use Case คือ ส่วนหนึ่งของระบบสารสนเทศที่ทำหน้าที่เป็นแหล่งจัดเก็บเนื้อหาที่ผู้สอนได้จัดเตรียมไว้สำหรับการสอน เช่น เนื้อหาเกี่ยวกับทฤษฎี ตัวอย่าง โจทย์ และข้อสอบ สิ่งที่เป็นประเด็นสำคัญในการออกแบบโครงสร้างฐานข้อมูลของ learning resources คือ เนื้อหาที่ใช้ประกอบการสอนวิธีใช้งานโปรแกรม GNU Octave มีความเกี่ยวข้องกับปัญหาทางวิศวกรรมและการคำนวณทางคณิตศาสตร์ ทำให้เนื้อหาต้องอยู่ในรูปแบบที่เหมาะสมกับการแสดงผลรูปภาพและสมการคณิตศาสตร์ นอกจากนี้ รูปแบบของเนื้อหาจะต้องเอื้อให้ผู้สอนสามารถเตรียมขึ้นได้อย่างง่าย ไม่จำเป็นต้องอาศัยซอฟต์แวร์

เฉพาะทางในการสร้างและจัดเก็บคลัง learning resources ในระหว่างการสอน เมื่อผู้สอนทำการเลือกเนื้อหาที่เหมาะสมในการสอนครั้งนั้น ดังที่นิยามไว้ในองค์ประกอบ query และ catalog information ก็จะได้รับตำแหน่งอ้างอิงของเนื้อหาที่ถูกเลือกภายใน learning resources ซึ่งจะส่งผ่านให้กับกลไกของ delivery ในรูปของ locator เพื่อส่งผ่านให้กับผู้เรียน



รูปที่ 3.4 การตอบโต้ระหว่างผู้สอนและผู้เรียนกับองค์ประกอบ learning resources และ delivery

การที่ผู้สอนในฐานะขององค์ประกอบ coach และ evaluation และผู้เรียนในฐานะขององค์ประกอบ learner entity ตามโครงสร้างของ LTSA จะเข้ามาเป็นส่วนหนึ่งของระบบสารสนเทศเพื่อสนับสนุนการเรียนการสอน จำเป็นต้องมีซอฟต์แวร์ที่ทำหน้าที่เชื่อมโยงข้อมูลและการสั่งการเข้ากับองค์ประกอบอื่นๆ การเชื่อมโยงระหว่างผู้สอนและผู้เรียนกับองค์ประกอบอื่นๆ สามารถแยกแยะตามองค์ประกอบที่เกี่ยวข้อง ประเภทของการสั่งการ และรายละเอียดของข้อมูลทำการส่งผ่าน ดังนี้

- ผู้เรียนในฐานะ learner entity ส่งรายละเอียดของข้อมูลที่เป็น behavior ได้แก่ กิจกรรมต่างๆ ในการติดต่อกับระบบสารสนเทศ คำสั่งต่างๆ ที่พิมพ์ป้อนให้กับโปรแกรม GNU Octave และไฟล์ต่างๆ ที่เกิดขึ้นระหว่างการเรียนการสอน ไปทำการจัดเก็บลงใน learner records
- ผู้สอนในฐานะ coach และ evaluation ร้องขอข้อมูลที่ถูกบันทึกไว้ใน learner records เพื่อใช้ประกอบการประเมินสถานะการเรียนรู้ของผู้เรียน
- ผู้สอนในฐานะ coach ทำการเลือกเนื้อหาที่เหมาะสมที่ถูกจัดเก็บไว้ใน learning resources และกำหนดให้กลไก delivery กระจายไปสู่กลุ่มของผู้เรียน
- ผู้เรียนในฐานะ learner entity ได้รับเนื้อหาจากกลไก delivery

การสื่อสารระหว่างซอฟต์แวร์ต่างๆ ภายในระบบสารสนเทศของห้องคอมพิวเตอร์ สามารถอาศัยเครือข่าย LAN แบบอีเทอร์เน็ตภายในห้องสอน ซึ่งถือได้ว่าเป็นความต้องการที่สมฤทธิ์ได้เสมอแล้วในปัจจุบัน สิ่งที่สำคัญสำหรับการเชื่อมโยงระหว่างองค์ประกอบของโครงสร้าง LTSA ดังที่จำแนกไว้ คือ การมีซอฟต์แวร์ที่ทำงานร่วมกับโปรแกรม GNU Octave ในการสื่อสารข้อมูลต่างๆ ที่ได้นิยามไว้ผ่านเครือข่ายคอมพิวเตอร์ และแสดงผลข้อมูลในรูปแบบที่แตกต่างกัน ภาษาคอมพิวเตอร์ของโปรแกรม GNU Octave ไม่สามารถรองรับการสื่อสารข้อมูลและการแสดงผลด้วยรูปแบบที่หลากหลาย ทำให้จำเป็นต้องพัฒนาซอฟต์แวร์ดังกล่าวด้วยภาษาคอมพิวเตอร์อื่น แล้วจึงประสานการทำงานร่วมกับโปรแกรม GNU Octave ซึ่งทำหน้าที่ในด้านการ

ประมวลผลทางคณิตศาสตร์ เพื่อให้กลายเป็นระบบสารสนเทศเพื่อสนับสนุนการเรียนการสอนทางด้านวิศวกรรมไฟฟ้าต่อไป

การพัฒนาซอฟต์แวร์สำหรับทำหน้าที่เป็นระบบสารสนเทศภายในห้องสอนคอมพิวเตอร์ เพื่อประยุกต์ใช้โปรแกรม GNU Octave ทำหน้าที่สนับสนุนการเรียนการสอนทางวิศวกรรมไฟฟ้า ตามที่ได้วิเคราะห์ด้วยการอ้างอิงโครงสร้างของ LTSA และแผนภูมิ Use Case แบ่งออกเป็น 3 ส่วนได้แก่

1. ซอฟต์แวร์บนเครื่องเซิร์ฟเวอร์ ทำหน้าที่จัดเก็บข้อมูลต่างๆ ในระหว่างการสอนลงในองค์ประกอบ learner records และเป็น learning resources ที่รวบรวมเนื้อหาต่างๆ ที่จำเป็นสำหรับการสอน รวมทั้งเป็นกลไกของกระบวนการ delivery ซึ่งส่งผ่านเนื้อหาประกอบการสอนที่ผู้สอนกำหนดไปสู่ผู้เรียน
2. ซอฟต์แวร์ส่วนติดต่อผู้ใช้ของโปรแกรม GNU Octave บนเครื่องคอมพิวเตอร์ของผู้สอน ทำหน้าที่เชื่อมต่อกับองค์ประกอบ learner records และ learning resources เพื่อดึงข้อมูลที่เกี่ยวข้องมาให้ผู้สอนสามารถประเมินสถานะของการเรียนการสอนในห้องสอน และเชื่อมต่อกับองค์ประกอบ delivery เพื่อควบคุมการส่งผ่านเนื้อหาที่เลือกไปสู่ผู้เรียนในห้อง
3. ซอฟต์แวร์ส่วนติดต่อผู้ใช้ของโปรแกรม GNU Octave บนเครื่องคอมพิวเตอร์ของผู้เรียน ทำหน้าที่รายงานสถานะและกิจกรรมต่างๆ ของผู้เรียนไปจัดเก็บลงในองค์ประกอบ learner records รวมทั้งแสดงเนื้อหาของ learning content ที่ถูกส่งจาก delivery ให้กับผู้เรียน

งานวิจัยนี้ได้เลือกภาษาคอมพิวเตอร์สำหรับการพัฒนาซอฟต์แวร์ทั้ง 3 ส่วน คือ ภาษา Python เพราะจุดเด่นหลายประการ ที่สอดคล้องกับวัตถุประสงค์ในการพัฒนาซอฟต์แวร์บนหลักการของลิขสิทธิ์แบบเปิดเผยต้นฉบับ และความต้องการของซอฟต์แวร์ที่วิเคราะห์ในข้างต้น ได้แก่

- เป็นภาษาคอมพิวเตอร์ที่ทำงานด้วยรูปแบบของการแปลภาษา (interpreted language) ทำให้สามารถแจกจ่ายต้นฉบับของซอฟต์แวร์ที่พัฒนาขึ้นให้กับผู้ใช้งาน โดยไม่จำเป็นต้องทำการคอมไพล์เป็นโปรแกรมคอมพิวเตอร์ ดังนั้น ผู้ใช้โปรแกรมสามารถที่จะศึกษาการทำงานของซอฟต์แวร์จากส่วนของต้นฉบับ และปรับปรุงแก้ไขให้เหมาะสมกับการใช้งานจริง
- เป็นภาษาคอมพิวเตอร์ที่รองรับความสามารถในการโปรแกรมเชิงวัตถุ (object-oriented programming) ทำให้ผู้ที่สนใจจะพัฒนาซอฟต์แวร์ต่อเนื่อง นำซอฟต์แวร์ไปทำการเพิ่มคุณลักษณะต่างๆ โดยไม่จำเป็นต้องเขียนต้นฉบับโปรแกรมใหม่ (code reusable)
- ซอฟต์แวร์ที่ใช้ในการพัฒนามีรูปแบบของลิขสิทธิ์แบบเปิดเผยต้นฉบับ เช่นเดียวกับซอฟต์แวร์เครื่องมือช่วยในการพัฒนา ได้แก่ wxPython สำหรับการพัฒนาสchnittstelleผู้ใช้แบบกราฟฟิก และ Twisted สำหรับการพัฒนาความสามารถในการสื่อสารผ่านเครือข่ายด้วยโปรโตคอลที่หลากหลาย

สำหรับรายละเอียดของการพัฒนาซอฟต์แวร์ในแต่ละส่วนถูกแสดงไว้ในหัวข้อ 3.3-3.4

3.3 การพัฒนาซอฟต์แวร์สำหรับเซิร์ฟเวอร์

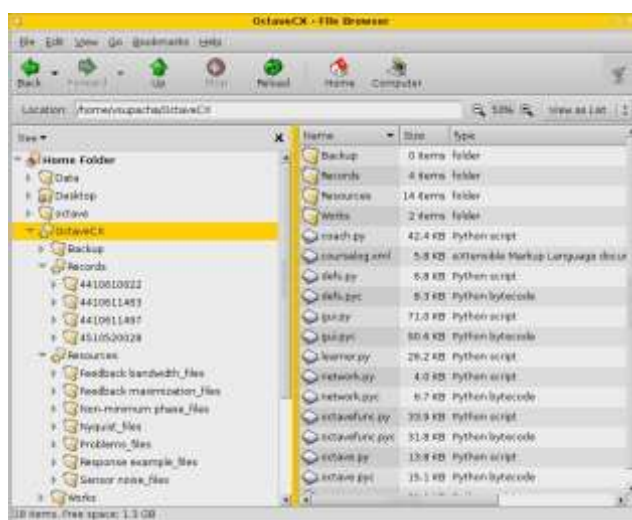
หน้าที่หลักของซอฟต์แวร์บนเครื่องเซิร์ฟเวอร์ คือ การทำงานตามหน้าที่ของแหล่งจัดเก็บ learner records และ learning resources ซึ่งเกี่ยวข้องกับข้อมูลและเนื้อหาต่างๆ ในการเรียนการสอน และการทำงานตามหน้าที่ของกระบวนการ delivery ในการส่งผ่านเนื้อหาที่ผู้สอนเลือกไปสู่คอมพิวเตอร์ของผู้เรียน ซอฟต์แวร์ที่ครอบคลุมหน้าที่การทำงานดังกล่าวจำเป็นต้องมีความสามารถในการจัดการฐานข้อมูลและการสื่อสารผ่านทางเครือข่ายคอมพิวเตอร์

3.3.1 การพัฒนาระบบฐานข้อมูลบนเซิร์ฟเวอร์

โครงสร้างของระบบฐานข้อมูลในเซิร์ฟเวอร์จะแบ่งออกเป็น 2 ส่วน เพื่อรองรับการทำงานของ learner records กล่าวคือ การบันทึกข้อมูลเกี่ยวกับกิจกรรมของผู้เรียนและไฟล์ต่างๆที่เป็นผลลัพธ์จากการสอน และการทำงานของ learning resources กล่าวคือ การจัดเก็บไฟล์เนื้อหาสำหรับประกอบการสอน เมื่อพิจารณาถึงรายละเอียดของข้อมูลที่เกี่ยวข้องในแต่ละแหล่งจัดเก็บ พบว่า

- สถานะของผู้เรียน คือ คำสั่งที่ผู้เรียนพิมพ์ป้อนให้กับโปรแกรม GNU Octave รวมทั้งคำตอบที่โปรแกรม GNU Octave แจ้งออกมา ต่างก็เป็นข้อความแบบตัวอักษรธรรมดา จึงสามารถบันทึกด้วยการเขียนเพิ่มเติมลงในไฟล์ตัวอักษรทั่วไปได้
- ไฟล์ที่เกิดขึ้นจากการเรียนการสอนแบ่งออกเป็น 2 ประเภทคือ ไฟล์ตัวอักษรที่มีนามสกุล .m ซึ่งบันทึกชุดคำสั่งที่ผู้เรียนได้เขียนขึ้น และไฟล์ไบนารีสำหรับข้อมูลตัวเลขที่ผู้เรียนสั่งให้โปรแกรม GNU Octave บันทึกในระหว่างการประมวลผล
- ไฟล์ของเนื้อหาสำหรับประกอบการสอน จะอยู่ในรูปของไฟล์ตัวอักษรแบบ html พร้อมทั้งไฟล์ไบนารีของรูปภาพที่เกี่ยวข้อง เพื่อให้สามารถแสดงรูปภาพและสมการทางคณิตศาสตร์

การที่ข้อมูลต่างๆที่บันทึกหรือจัดเก็บใน learner records และ learning resources อยู่ในรูปของไฟล์คอมพิวเตอร์ ทำให้การจัดเก็บในรูปแบบเรคอร์ดของระบบฐานข้อมูลทั่วไป เช่น MySQL อาจจะไม่เหมาะสม เพราะการจัดการใดๆจำเป็นต้องมีซอฟต์แวร์ในการดึงข้อมูลออกจากฐานข้อมูลและบันทึกเป็นไฟล์คอมพิวเตอร์อีกครั้ง ดังนั้น การพัฒนาโครงสร้างของการบันทึกและจัดเก็บข้อมูลใน learner records และ learning resources จะใช้โครงสร้างของระบบไฟล์และไดเรกทอรีทั่วไป ซึ่งผู้สอนสามารถทำสำเนาออกจากเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ เพื่อนำไปจัดเก็บหรือประเมินนอกห้องสอนคอมพิวเตอร์ได้อย่างสะดวก



รูปที่ 3.5 โครงสร้างของระบบไฟล์และไดเรกทอรีบนเครื่องเซิร์ฟเวอร์

การจัดการในส่วน learner records จะใช้การสร้างไฟล์เดอรัย่อยที่ตั้งชื่อตามหมายเลขทะเบียนหรือชื่อของผู้เรียน เพื่อจัดเก็บไฟล์ต่างๆของผู้เรียนที่เกิดขึ้นจากการเรียนการสอน และบันทึกกิจกรรมต่างๆในระหว่างการเรียนการสอนลงในไฟล์สำหรับการบันทึก การจัดการในส่วนของ learning resources จะใช้ไฟล์เดอรัย่อยที่มี Resources สำหรับเก็บไฟล์แบบ html โดยรูปภาพต่างๆที่ใช้ประกอบไฟล์ html จะถูกจัดเก็บในไฟล์เดอรัย่อยที่มี

ข้อสอดคล้องกับไฟล์ การสร้างเนื้อหาสำหรับประกอบการสอนในรูปแบบของไฟล์ html สามารถใช้โปรแกรม Microsoft Office หรือ OpenOffice ในการจัดพิมพ์เนื้อหาที่มีรูปภาพและสมการทางคณิตศาสตร์ประกอบ จากนั้นจึงเลือกให้บันทึกเป็นไฟล์แบบ html เมื่อสร้างเนื้อหาประกอบการสอนในรูปแบบไฟล์แบบ html แล้ว ผู้สอนสามารถใช้การทำสำเนาไฟล์ต่างๆที่สร้างขึ้นลงในโพลเดอร์ Resources ซึ่งอยู่ภายใต้โพลเดอร์ที่จะเรียกใช้งานซอฟต์แวร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ โครงสร้างของระบบไฟล์และไดเรกทอรีเมื่อผู้สอนจะเรียกใช้งานโปรแกรมเซิร์ฟเวอร์ภายใต้โพลเดอร์ /home/vsupacha/OctaveCX/ ถูกแสดงไว้ในรูปที่ 3.5

3.3.2 การพัฒนาเว็บเซอร์วิสบนเซิร์ฟเวอร์

การเชื่อมต่อระหว่างแหล่งจัดเก็บ learner records และ learning resources ในโครงสร้าง LTSA กับองค์ประกอบอื่นๆ จำเป็นต้องมีกลไกในการสื่อสารทั้งคำสั่งและข้อมูลผ่านทางเครือข่ายคอมพิวเตอร์ ซึ่งในงานวิจัยนี้เป็นเครือข่ายแบบฮิเตอร์เน็ตที่อาศัยโปรโตคอล TCP/IP เป็นพื้นฐาน เทคโนโลยีการสื่อสารผ่านเครือข่ายคอมพิวเตอร์ที่เหมาะสมทั้งการส่งคำสั่งเพื่อให้เกิดการประมวลผลและการร้องขอข้อมูล คือ เทคโนโลยีเว็บเซอร์วิส ซึ่งใช้การส่งคำร้องขอที่มีโครงสร้างระบุด้วยไวยากรณ์แบบ XML ในการขอใช้บริการจากคอมพิวเตอร์ที่เปิดให้ส่งงานผ่านทางเครือข่าย ปัจจุบัน เทคโนโลยีเว็บเซอร์วิสมีอยู่หลายรูปแบบ เช่น XML-RPC, SOAP, DCOM เป็นต้น โดยเทคโนโลยีที่เลือกมาใช้ในงานวิจัยนี้คือ XML-RPC ซึ่งเป็นการปรับปรุงกลไกของการส่งงานระยะไกล (remote procedure call) ด้วยการใช้โครงสร้างของไวยากรณ์ XML ทำให้ข้อมูลที่ถูกส่งผ่านเครือข่ายมีรูปแบบที่ไม่ขึ้นอยู่กับประเภทของคอมพิวเตอร์ และรองรับประเภทของข้อมูลที่หลากหลาย การออกแบบบริการสำหรับรองรับการส่งงานระยะไกลบนเครื่องเซิร์ฟเวอร์ จะอิงอยู่บนการวิเคราะห์นิยามของวิธีการ (method) ขององค์ประกอบ learner records, learning resources และ delivery ตามที่ระบุไว้ในมาตรฐาน IEEE 1484 ดังนี้

- วิธีการขององค์ประกอบ learner records ได้แก่ การจัดเก็บ (Store) และการเรียกขอ (Retrieve) ซึ่งทำการรับส่งข้อมูลผ่านทางองค์ประกอบ learner information ดังนี้
 - ผู้เรียนส่งข้อมูลของการใช้งานโปรแกรม GNU Octave มาบันทึก ได้แก่ คำสั่งที่ผู้เรียนพิมพ์ป้อน และคำตอบที่แจ้งกลับมา
 - ผู้เรียนส่งไฟล์ที่เกิดขึ้นระหว่างการเรียนการสอนมาจัดเก็บ ได้แก่ ไฟล์ของชุดคำสั่งที่เป็นไฟล์แบบตัวอักษร และไฟล์ของข้อมูลตัวเลขที่บันทึกจากโปรแกรม GNU Octave
 - ผู้เรียนทำการเรียกขอไฟล์ที่ทำการจัดเก็บไว้
 - ผู้สอนทำการเรียกขอข้อมูลเกี่ยวกับผู้เรียน
 - ผู้สอนทำการเรียกขอไฟล์ที่บันทึกคำสั่งและคำตอบที่ผู้เรียนได้ทำไป เพื่อประเมินสถานะของการเรียนรู้
 - ผู้สอนทำการเรียกขอไฟล์ที่ผู้เรียนได้ทำการจัดเก็บไว้
- วิธีการขององค์ประกอบ learning resources ได้แก่ การเรียกขอ (Retrieve) ที่ทำงานแตกต่างกันตามองค์ประกอบ query, catalog information และ locator ดังนี้
 - ผู้สอนเรียกขอดูรายการของไฟล์เนื้อหาทั้งหมด ที่มีอยู่ใน learning resources
 - ผู้สอนระบุไฟล์เนื้อหาใน learning resources แล้ว เพื่อขอรายละเอียดตำแหน่งอ้างอิงในรูปแบบ URL ของไฟล์นั้น

- องค์ประกอบ delivery ไม่มีวิธีการโดยเฉพาะ โดยมีการรับส่งข้อมูลดังนี้
 - ผู้สอนแจ้งตำแหน่ง URL ของเนื้อหาสำหรับกระจายให้ผู้เรียน โดยอาจเป็นการส่งให้แบบรายบุคคล หรือส่งให้ทั้งห้องสอน

การวิเคราะห์เกี่ยวกับวิธีการและข้อมูลขององค์ประกอบ learner records, learning resources และ delivery ทำให้สามารถกำหนดบริการของเซิร์ฟเวอร์ที่เป็รับการสั่งงานระยะไกลผ่านทางโปรโตคอล XML-RPC ดังนี้

1. บริการขององค์ประกอบ learner records
 - a. StoreBehavior ผู้เรียนส่งคำสั่งที่พิมพ์ป้อนให้โปรแกรม GNU Octave พร้อมทั้งการตอบสนองต่อคำสั่งนั้น และรายละเอียดของกิจกรรมต่างๆ มาบันทึกในไฟล์บันทึกสถานะของการเรียนรู้
 - b. StoreFile ผู้เรียนทำการส่งไฟล์มาเก็บในโฟลเดอร์ของผู้เรียน
 - c. RetrieveLearnerInfo ผู้สอนเรียกขอข้อมูลเกี่ยวกับผู้เรียน
 - d. RetrieveLearnerDict ผู้สอนเรียกขอข้อมูลและสถานะของผู้เรียนทั้งหมดที่ลงทะเบียนไว้
 - e. RetrieveBehavior ผู้สอนเรียกขอเนื้อหาของไฟล์ที่บันทึกสถานะของผู้เรียนที่ถูกระบุ
 - f. QueryFiles ผู้เรียนเรียกขอรายการของไฟล์ทั้งหมดที่จัดเก็บอยู่ในโฟลเดอร์ของผู้เรียนเอง โดยผู้สอนก็สามารถเรียกขอรายการของไฟล์ได้เช่นเดียวกัน
 - g. RetrieveFile ผู้เรียนเรียกขอไฟล์ที่จัดเก็บอยู่ในโฟลเดอร์ของผู้เรียนเอง โดยผู้สอนก็สามารถเรียกขอไฟล์ได้เช่นเดียวกัน
2. บริการขององค์ประกอบ learning resources
 - a. QueryContents ผู้สอนดึงรายการของไฟล์ทั้งหมดที่จัดเก็บในโฟลเดอร์ Resources
 - b. RetrieveContentURL ผู้สอนร้องขอตำแหน่งอ้างอิงในรูปของ URL ของไฟล์ในโฟลเดอร์ Resources ที่ระบุ
3. บริการขององค์ประกอบ delivery
 - a. DeliveryLearner ผู้สอนกำหนดไฟล์แบบ html และตัวผู้เรียน ซึ่งองค์ประกอบ delivery จะส่งตำแหน่งของไฟล์ในรูปของ URL ไปให้ผู้เรียนนั้นร้องขอผ่านทางโปรโตคอล HTTP และแสดงผลบนหน้าจอของผู้เรียน
 - b. Delivery ผู้สอนกำหนดไฟล์แบบ html ซึ่งองค์ประกอบ delivery จะส่งตำแหน่งของไฟล์ในรูปของ URL ไปให้ผู้เรียนทุกคนในห้องสอนทำการร้องขอผ่านทางโปรโตคอล HTTP และแสดงผลบนหน้าจอของผู้เรียน

นอกจากบริการที่อิงตามนิยามขององค์ประกอบในโครงสร้าง LTSA แล้ว เซิร์ฟเวอร์ยังต้องเปิดบริการพื้นฐานของระบบสารสนเทศได้แก่

4. การจัดการห้องสอน
 - a. CoachLogin ผู้สอนทำการร้องขอเพื่อเข้าสู่ระบบ
 - b. CoachLogout ผู้สอนทำการร้องขอเพื่อออกจากระบบ
 - c. LearnerLogin ผู้เรียนทำการร้องขอเพื่อเข้าสู่ระบบ
 - d. LearnerLogout ผู้เรียนทำการร้องขอเพื่อออกจากระบบ
 - e. AddLearner ผู้สอนทำการเพิ่มผู้เรียนลงในระบบ พร้อมทั้งป้อนข้อมูลพื้นฐาน

3.3.3 การพัฒนาส่วนประมวลผลโปรโตคอล

การทำงานของซอฟต์แวร์เพื่อให้บริการสำหรับการส่งงานระยะไกลผ่านเครือข่ายคอมพิวเตอร์ตามที่ได้ออกมาข้างต้นนี้ มีข้อจำกัดที่ต้องคำนึงถึงในหลายประเด็น โดยเฉพาะการที่คอมพิวเตอร์ของห้องสอนอาจต้องทำหน้าที่เป็นเซิร์ฟเวอร์ของหน่วยงานนั้น ดังนั้น การประมวลผลคำร้องขอ อาจจะไม่สามารถเรียกใช้บริการของซอฟต์แวร์ที่ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ของคอมพิวเตอร์นั้นเช่น โปรแกรม Microsoft IIS หรือโปรแกรม Apache และไม่สามารถใช้พอร์ตสื่อสาร ซึ่งนิยมใช้สำหรับการให้บริการเว็บเซิร์ฟเวอร์ ได้แก่ พอร์ต 80 และ 8080 เพื่อไม่ให้เกิดความขัดแย้งในการทำงานระหว่างซอฟต์แวร์ งานวิจัยนี้จึงได้พัฒนาซอฟต์แวร์ในส่วนของเซิร์ฟเวอร์ สำหรับทำหน้าที่ประมวลผลคำร้องขอที่ส่งผ่านเครือข่าย โดยอาศัยโมดูล Twisted ในการสร้างส่วนจัดการคำร้องขอ ส่วนโค้ดของซอฟต์แวร์ทำหน้าที่เป็นเซิร์ฟเวอร์ได้รวมไว้ในไฟล์ server.py โดยมีรายละเอียดดังนี้

1. เรียกใช้คลาส static ในโมดูล twisted.web เพื่อสร้างส่วนจัดการโปรโตคอล HTTP ที่พอร์ต 8000 ในแบบเว็บเซิร์ฟเวอร์ สำหรับให้ผู้เรียนเรียกขอไฟล์เนื้อหาแบบ html ภายใต้โฟลเดอร์ Resources ซึ่งเป็นตำแหน่งจัดเก็บไฟล์ทั้งหมดของ learning resources
2. เรียกใช้คลาส xmlrpc ในโมดูล twisted.web เพื่อสร้างส่วนจัดการโปรโตคอล XML-RPC ที่พอร์ต 8001 โดยอาศัยวิธีการที่กำหนดไว้ในคลาส OctaveServerService สำหรับเป็นบริการ
3. เรียกใช้โมดูล xmlrpclib ในการสร้างส่วนไคลเอนต์สำหรับร้องขอบริการบนเครื่องคอมพิวเตอร์ของผู้สอนและผู้เรียน ด้วยการส่งงานระยะไกลผ่านโปรโตคอล XML-RPC

```
import os, stat, sys, time, exceptions
from twisted.web import xmlrpc, static, server
from twisted.internet import reactor
import xmlrpclib

class OctaveCX_Server(xmlrpc.XMLRPC):
    """ The XML-RPC server class for the framework of GNU Octave Classroom
    eXtension """

    def __init__(self):
        xmlrpc.XMLRPC.__init__(self)
        self.coach = None
        self.learners = []

    """ Services for interfacing with learner records """
    def xmlrpc_StoreBehavior(self, behavior):

        ↓

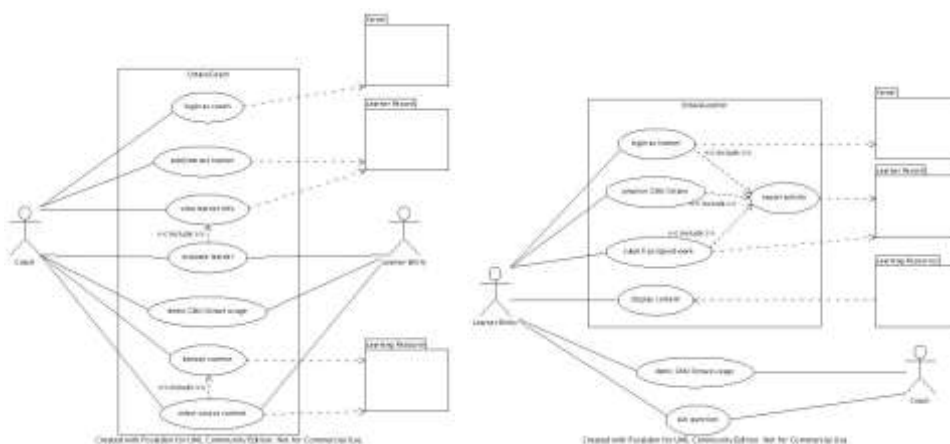
if __name__ == '__main__':
    """ Implement static web service for port 8000 """
    root = static.File('Resources')
    web_site = server.Site(root)
    reactor.listenTCP(8000, web_site)

    """ Implement XML-RPC service for port 8001 """
    classServer = OctaveCX_Server()
    xmlrpc.addIntrospection(classServer)
    xmlrpc_site = server.Site(classServer)
    reactor.listenTCP(8001, xmlrpc_site)

    reactor.run()
```

3.4 การพัฒนาซอฟต์แวร์สำหรับผู้สอนและผู้เรียน

เมื่ออ้างอิงตามโครงสร้าง LTSA ซอฟต์แวร์ที่ผู้สอนและผู้เรียนใช้งานในระหว่างการเรียนการสอน จำเป็นต้องมีความสามารถในการเชื่อมต่อกับซอฟต์แวร์บนเซิร์ฟเวอร์ เพื่อเรียกใช้บริการต่างๆในการทำงานตามหน้าที่ของ learner entity, coach และ evaluation การแยกแยะกิจกรรมต่างๆของผู้สอนและผู้เรียนภายในระบบสารสนเทศได้แสดงไว้ในรูปที่ 3.6



รูปที่ 3.6 แผนภาพ Use Case สำหรับผู้สอนในฐานะ coach/evaluation และผู้เรียนในฐานะ learner entity

ทั้งนี้ ซอฟต์แวร์สำหรับผู้สอนจะมีฟังก์ชันในการเรียกขอข้อมูลต่างๆของผู้เรียนทั้งหมดใน learner records และเนื้อหาที่จัดเตรียมใน learning resources จากเซิร์ฟเวอร์ รวมทั้งสามารถควบคุมกระบวนการ delivery ในการกระจายเนื้อหาไปสู่ผู้เรียน ในขณะที่ซอฟต์แวร์สำหรับผู้เรียนจะสามารถเรียกขอข้อมูลในส่วนที่เกี่ยวข้องกับผู้เรียนเท่านั้น โดยไม่รวมถึงข้อมูลเกี่ยวกับกิจกรรมต่างๆที่ถูกบันทึกระหว่างการเรียนการสอน งานวิจัยนี้จึงได้พัฒนาซอฟต์แวร์สำหรับผู้สอนและผู้เรียนแยกออกจากกัน โดยซอฟต์แวร์สำหรับผู้สอนคือโปรแกรม OctaveCoach และซอฟต์แวร์สำหรับผู้เรียนคือโปรแกรม OctaveLearner

3.4.1 การพัฒนาส่วนติดต่อผู้ใช้

สิ่งที่เป็นอุปสรรคหลักในการประยุกต์ใช้โปรแกรม GNU Octave ร่วมกับระบบสารสนเทศเพื่อสนับสนุนการเรียนการสอนตามโครงสร้างของ LTSA คือ การที่โปรแกรม GNU Octave จะต้องสามารถส่งข้อมูลเกี่ยวกับกิจกรรมของผู้เรียน ได้แก่ คำสั่งที่พิมพ์และคำตอบที่ได้จากการประมวลผลเข้าสู่ learner records บนเครื่องเซิร์ฟเวอร์ การเชื่อมต่อกับเซิร์ฟเวอร์เพื่อบันทึกข้อมูลตามนิยามของ behavior นี้ จำเป็นต้องมีการปรับปรุงกลไกในการดักคำสั่งที่ผู้เรียนพิมพ์และข้อความที่โปรแกรม GNU Octave ตอบสนองต่อคำสั่งนั้น อย่างไรก็ตาม กลไกรับคำสั่งจากผู้ใช้โปรแกรมมีความเกี่ยวข้องกับส่วนโค้ดที่รับและประมวลผลการพิมพ์คำสั่ง จึงไม่เหมาะสมในการเข้าไปดัดแปลงในส่วนต้นฉบับโปรแกรม เพราะอาจทำให้การทำงานผิดพลาดและมีผลกระทบต่อส่วนอื่นๆของโปรแกรมได้ งานวิจัยนี้ทำการพัฒนาซอฟต์แวร์ที่ทำหน้าที่เป็นส่วนติดต่อผู้ใช้แบบกราฟฟิกของโปรแกรม GNU Octave โดยรับคำสั่งจากผู้เรียนและส่งผ่านให้โปรแกรม GNU Octave พร้อม

ทั้งเรียกใช้บริการของเซิร์ฟเวอร์เพื่อบันทึกกิจกรรมของผู้เรียน การทำงานแบบ front-end นี้จะไม่มีผลกระทบต่อการทำงานของผลภายในโปรแกรม GNU Octave ในขณะเดียวกันก็เป็นการอำนวยความสะดวกต่อการใช้งานโปรแกรม GNU Octave ด้วยการเพิ่มคุณลักษณะต่างๆ เช่น การสั่งการด้วยเมนู การมีหน้าต่างสำหรับแสดงรายละเอียดต่างๆ และการมีหน้าต่างสำหรับเขียนชุดคำสั่ง เป็นต้น นอกจากนี้ ส่วนติดต่อผู้ใช้แบบกราฟิกยังทำหน้าที่แสดงเนื้อหาจาก learning resources ซึ่งอยู่ในรูปแบบไฟล์ html

การพัฒนาซอฟต์แวร์ในส่วนนี้ได้ใช้ภาษา Python ร่วมกับ wxPython ซึ่งเป็นโมดูลส่วนขยายสำหรับสร้างส่วนติดต่อผู้ใช้แบบกราฟิก การออกแบบรูปแบบของส่วนติดต่อผู้ใช้แบบกราฟิกจะแบ่งหน้าต่างของโปรแกรมออกเป็น 3 ส่วน ได้แก่

1. กลุ่มของหน้าต่างย่อยสำหรับสั่งงานโปรแกรม GNU Octave ประกอบด้วยหน้าต่างย่อยของโปรแกรม Octave และหน้าต่างย่อยสำหรับการแก้ไขไฟล์ชุดคำสั่ง
2. กลุ่มของหน้าต่างย่อยสำหรับแสดงรายละเอียดประกอบการป้อนคำสั่ง ประกอบด้วยหน้าต่างย่อยแสดงตัวแปร หน้าต่างย่อยแสดงคำสั่งต่างๆของโปรแกรม GNU Octave และหน้าต่างแสดงคำสั่งต่างๆที่ได้ถูกพิมพ์ไปแล้ว
3. กลุ่มของหน้าต่างย่อยสำหรับเชื่อมต่อระบบสารสนเทศเพื่อสนับสนุนการสอน ประกอบด้วยหน้าต่างย่อยที่แสดงข้อมูลต่างๆบนเซิร์ฟเวอร์

คลาสและวิธีการพื้นฐานของส่วนติดต่อผู้ใช้แบบกราฟิกได้ถูกรวมไว้ในไฟล์ gui.py ดังแสดงในโครงสร้างของคลาสในรูปที่ 3.7



รูปที่ 3.7 คลาสพื้นฐานต่างๆสำหรับการทำงานของส่วนติดต่อผู้ใช้แบบกราฟิก

โครงสร้างของส่วนติดต่อผู้ใช้แบบกราฟิกจะเริ่มจากคลาส OctaveFrame ซึ่งสืบทอดจากคลาส wx.Frame สำหรับแสดงผลและจัดการส่วนหน้าต่างหลักของโปรแกรม ขั้นตอนของการสร้างวัตถุของคลาสนี้จะมีการสร้างองค์ประกอบต่างๆ ของส่วนติดต่อผู้ใช้ ดังนี้

- คลาส OctaveMenuBar (สืบทอดจาก wx.MenuBar) ทำหน้าที่จัดการเมนูของหน้าต่างหลัก
- คลาส OctaveToolBar (สืบทอดจาก wx.Toolbar) ทำหน้าที่จัดการแถบเครื่องมือ

- คลาส OctaveStatusbar (สืบทอดจาก wx.StatusBar) ทำหน้าที่จัดการแถบแสดงสถานะของหน้าต่างหลัก
- คลาส ConsoleNotebook (สืบทอดจาก wx.Notebook) ทำหน้าที่จัดการกลุ่มของหน้าต่างย่อยที่เกี่ยวข้องกับการสั่งงาน โปรแกรม GNU Octave
- คลาส SupportNotebook (สืบทอดจาก wx.Notebook) ทำหน้าที่จัดการกลุ่มของหน้าต่างย่อยที่สนับสนุนการป้อนคำสั่ง
- คลาส ClassroomNotebook (สืบทอดจาก wx.Notebook) ทำหน้าที่จัดการกลุ่มของหน้าต่างย่อยที่แสดงผลข้อมูลที่เกี่ยวข้องกับระบบสารสนเทศบนเวิร์ฟเวอร์

คลาสอื่นๆในไฟล์ gui.py จะเกี่ยวข้องกับองค์ประกอบในแต่ละกลุ่มของหน้าต่างย่อย ได้แก่

- WorkspacePanel (สืบทอดจาก BasicPanel) จัดการหน้าต่างที่แสดงรายละเอียดของตัวแปรต่างๆที่โปรแกรม GNU Octave รับรู้ เป็นต้น
- FunctionPanel (สืบทอดจาก BasicPanel) จัดการหน้าต่างที่แสดงรายละเอียดของฟังก์ชันที่โปรแกรม GNU Octave สามารถใช้งานได้
- HistoryPanel (สืบทอดจาก BasicPanel) จัดการหน้าต่างที่แสดงรายการของคำสั่งที่ผู้ใช้ได้พิมพ์ไปแล้ว
- ResourceListPanel (สืบทอดจาก BasicPanel) จัดการหน้าต่างที่แสดงรายการของไฟล์เนื้อหาภายใน learning resources
- FileListPanel (สืบทอดจาก BasicPanel) จัดการหน้าต่างที่แสดงรายการของไฟล์ต่างๆที่เก็บอยู่ใน learner records
- ContentPanel (สืบทอดจาก BasicPanel) จัดการหน้าต่างที่แสดงเนื้อหาของไฟล์ทั้งแบบตัวอักษรและแบบ html

3.4.2 การเชื่อมต่อกับโปรแกรม GNU Octave

โปรแกรม OctaveCoach และ OctaveLearner จำเป็นต้องเชื่อมต่อกับโปรแกรม GNU Octave เพื่อส่งผ่านคำสั่งที่ผู้ใช้ป้อนไปทำการประมวลผล และนำคำตอบที่ได้จากการประมวลผลมาแสดงผล การเปิดช่องสื่อสารประเภท pipe เพื่อติดต่อสื่อสารกับโปรแกรม GNU Octave ที่ทำงานในเบื้องหลัง มีปัญหาหลักในกรณีที่โปรแกรม GNU Octave ใช้เวลาในการประมวลผลที่นาน ก็จะส่งผลถึงการตอบสนองทั้งหมดของโปรแกรม OctaveCoach และ OctaveLearner ได้ งานวิจัยนี้ใช้การสร้างเธรด (thread) ที่จะทำงานคู่ขนานกับส่วนติดต่อผู้ใช้ของโปรแกรม OctaveCoach และ OctaveLearner เพื่อสื่อสารกับโปรแกรม GNU Octave จากนั้นจึงส่งผลการทำงานไปที่หน้าต่างรับคำสั่งโดยตรง คลาสต่างๆที่เกี่ยวข้องกับการสั่งการโปรแกรม GNU Octave ซึ่งทำงานแบบเบื้องหลังได้รวบรวมไว้ในไฟล์ octave.py ซึ่งมีโครงสร้างดังในรูปที่ 3.8



รูปที่ 3.8 คลาสสำหรับการเชื่อมต่อกับโปรแกรม GNU Octave ที่ทำงานแบบเบื้องหลัง

คลาสที่สำคัญของไฟล์ octave.py คือ

- OctaveShell (สืบทอดจาก wx.py.shell.Shell) จัดการหน้าต่างย่อยที่ผู้ใช้โปรแกรมจะสั่งการ โปรแกรม GNU Octave
- OctaveEditor (สืบทอดจาก wx.py.editor.Editor) จัดการหน้าต่างย่อยที่ผู้ใช้โปรแกรมสามารถแก้ไขไฟล์ของชุดคำสั่ง
- OctaveConsole (สืบทอดจาก threading) เป็นคลาสของวัตถุที่ทำงานแบบเธรด สำหรับเชื่อมต่อกับโปรแกรม GNU Octave ซึ่งเรียกให้ทำงานผ่านช่องสื่อสารแบบ pipe

3.4.3 การพัฒนาเว็บเซอร์วิสบนไคลเอนต์

การศึกษากิจกรรมต่างๆที่เกิดขึ้นภายในระบบสารสนเทศดังในรูปที่ 3.6 ซึ่งให้เห็นว่าโปรแกรม OctaveCoach และ OctaveLearner จำเป็นต้องให้บริการสำหรับการติดต่อจากเซิร์ฟเวอร์เช่นเดียวกัน การติดต่อจากเซิร์ฟเวอร์มีอยู่ในหลายรูปแบบ ทั้งการยื่นยันสถานะในการสื่อสาร การแจ้งถึงสถานะของผู้เรียนที่เปลี่ยนแปลงไป และการแจ้งตำแหน่งของเนื้อหาที่กระจาย อย่างไรก็ตาม การพัฒนาความสามารถในการสื่อสารผ่านเครือข่ายคอมพิวเตอร์ด้วย Twisted มีข้อจำกัดในด้านการทำงานร่วมกับ wxPython เนื่องจากต่างก็ใช้โครงสร้างของวงรอบรอเหตุการณ์ (event loop) ทำให้ขัดแย้งกัน การพัฒนาโค้ดสำหรับส่วนให้บริการจึงได้เลือกใช้แพ็คเกจ SimpleXMLRPCServer ร่วมกับคลาส ThreadingMixIn ในแพ็คเกจ SocketServer เพื่อสร้างกลไกของบริการของโปรโตคอล XML-RPC ซึ่งจะสร้างเธรดสำหรับรองรับแต่ละการเชื่อมต่อแบบซ็อกเก็ตที่มีต่อโปรแกรม OctaveCoach และ OctaveLearner รายการที่ 3.2 แสดงโครงสร้างของโค้ดสำหรับการสร้างบริการสำหรับรองรับการเชื่อมต่อผ่านโปรโตคอล XML-RPC บนฝั่งไคลเอนต์

```
class AsyncXMLRPCServer(SocketServer.ThreadingMixIn,
SimpleXMLRPCServer.SimpleXMLRPCServer):
    """ Threaded mix-in server for client side service. """
    pass

class XMLRPCServer(threading.Thread):
    """ Thread class - provide XML-RPC services at client side """

    def __init__(self, port, service):
        """ Init thread, bind handler and start thread. """
        threading.Thread.__init__(self)
        self.IP = get_ip_address('eth0')
        self.setDaemon(True)
        self.server = AsyncXMLRPCServer(('', self.port),
SimpleXMLRPCServer.SimpleXMLRPCRequestHandler)

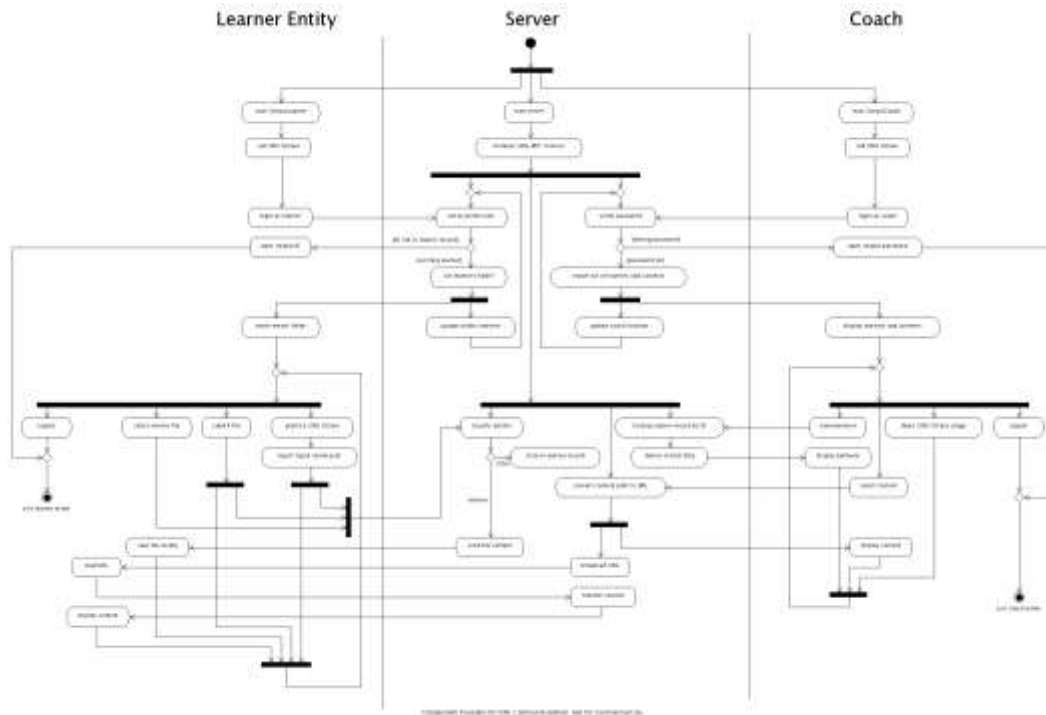
        # Register service
        self.server.register_instance(service)
        self.start()

    def run(self):
        """ Run State - register server. """
        self.server.serve_forever()
```

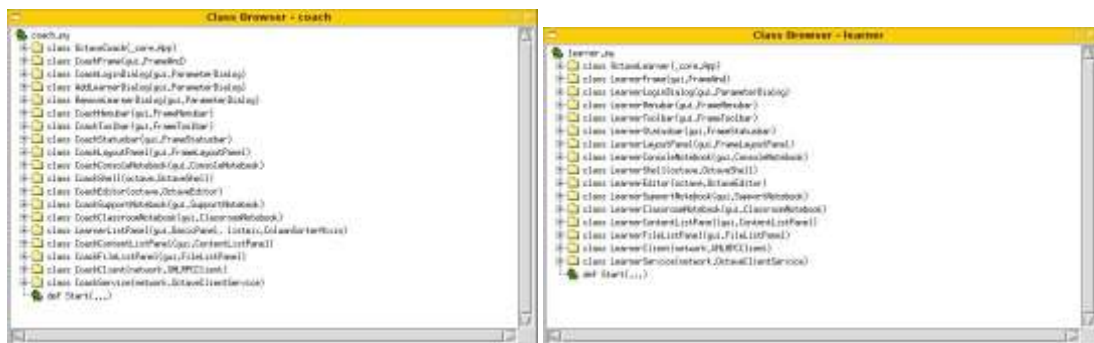
รายการที่ 3.2 การสร้างบริการสำหรับโปรโตคอล XML-RPCบนฝั่งไคลเอนต์

3.4.4 การทำงานของโปรแกรม OctaveCoach และ OctaveLearner

กระบวนการทำงานของโปรแกรม OctaveCoach และ OctaveLearner แสดงไว้ในแผนภาพลำดับในรูปที่ 3.9 โดยคลาสและวิธีการที่จำเพาะสำหรับผู้สอนและผู้เรียน ซึ่งสืบทอดจากคลาสต่างๆในไฟล์ gui.py octave.py และ network.py ได้ถูกรวมไว้ในไฟล์ coach.py และ learner.py ตามลำดับ ดังแสดงไว้ในรูปที่ 3.10



รูปที่ 3.9 การจำแนกกิจกรรมต่างๆของเซิร์ฟเวอร์ ผู้สอน และผู้เรียน



รูปที่ 3.10 คลาสต่างๆสำหรับโปรแกรม OctaveCoach และ OctaveLearner

บทที่ 4

การใช้งานซอฟต์แวร์

งานวิจัยนี้ได้พัฒนาซอฟต์แวร์ขึ้นทั้งหมด 3 ส่วน โดยแบ่งเป็นโปรแกรม OctaveServer สำหรับทำงานเป็นเซิร์ฟเวอร์ของการเรียนการสอน โปรแกรม OctaveCoach สำหรับให้ผู้สอนใช้ประกอบการสอนและจัดการข้อมูลต่างๆของการเรียนการสอนในรายวิชานั้นๆ และโปรแกรม OctaveLearner สำหรับให้นักศึกษาหรือผู้เรียนใช้ประกอบการเรียน การเป็นระบบสารสนเทศเพื่อสนับสนุนการเรียนการสอนในงานวิจัยนี้ จะนิยามโดยอิงอยู่บนสถานะการทำงานของโปรแกรม OctaveServer และการเชื่อมต่อจากผู้สอนและผู้เรียนเข้าสู่เซิร์ฟเวอร์ผ่านทางโปรแกรม OctaveCoach และ OctaveLearner ทั้งนี้ หากขาดองค์ประกอบใดก็ตาม ถือว่าไม่ใช่เป็นระบบสารสนเทศเพื่อสนับสนุนการเรียนการสอนตามโครงสร้างของ LTSA

เนื้อหาในบทนี้ จะเป็นการอธิบายถึงการใช้งานซอฟต์แวร์ต่างๆที่ได้พัฒนาขึ้น โดยหัวข้อ 4.1 จะอธิบายวิธีการใช้งานส่วนติดต่อผู้ใช้ต่างๆที่ได้พัฒนาขึ้นเพื่อความสะดวกในการใช้งาน โปรแกรม GNU Octave โดยครอบคลุมถึงกรณีซึ่งไม่ได้ใช้งานภายใต้ระบบสารสนเทศ ส่วนรายละเอียดของขั้นตอนที่ผู้สอนและผู้เรียนจะปฏิบัติเพื่อเรียกใช้ความสามารถของการเป็นระบบสารสนเทศ จะอธิบายในหัวข้อ 4.2 และ 4.3 ตามลำดับ

4.1 การใช้งานทั่วไป

4.1.1 การติดตั้งซอฟต์แวร์

งานวิจัยนี้ได้พัฒนาซอฟต์แวร์ต่างๆโดยอิงอยู่บนการใช้ซอฟต์แวร์เครื่องมือสำหรับพัฒนาที่เป็นแบบเปิดเผยต้นฉบับ 3 ชิ้น ได้แก่ Python รุ่น 2.3, wxPython รุ่น 2.6.1 และ Twisted รุ่น 2.0 โดยทำการพัฒนาและทดสอบการทำงานบนคอมพิวเตอร์ที่มีหน่วยประมวลผลแบบ Pentium IV และติดตั้งระบบปฏิบัติการลินุกซ์แบบ Debian ความต้องการพื้นฐานของฮาร์ดแวร์และระบบปฏิบัติการของการใช้งานระบบสารสนเทศที่งานวิจัยนี้ได้พัฒนาขึ้น ได้แก่ คอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการลินุกซ์พร้อมทั้งระบบจัดการกราฟฟิก X Window System รุ่น X11R6 และ GNOME รุ่น 2.X ซอฟต์แวร์ของงานวิจัยนี้สามารถทำงานบนฮาร์ดแวร์ของคอมพิวเตอร์ที่ใช้หน่วยประมวลผลอื่นที่ไม่ใช่ตระกูล Pentium ได้ ในกรณีที่มีซอฟต์แวร์ Python สำหรับหน่วยประมวลผลแบบนั้น เช่น ตระกูล Sparc หรือ PowerPC เป็นต้น อย่างไรก็ตาม การใช้งานบนคอมพิวเตอร์ที่ติดตั้งระบบปฏิบัติการ Microsoft Windows นั้น ยังไม่สามารถทำได้ในซอฟต์แวร์รุ่นปัจจุบัน เนื่องจากข้อจำกัดหลายอย่างของโปรแกรม GNU Octave ที่พัฒนาสำหรับทำงานบนระบบปฏิบัติการ Microsoft Windows รวมถึงข้อจำกัดในการเรียกข้อมูลเกี่ยวกับระบบเครือข่ายจากระบบปฏิบัติการ Microsoft Windows

นอกจากระบบปฏิบัติการลินุกซ์และระบบจัดการกราฟฟิกแล้ว การใช้งานโปรแกรม OctaveServer จำเป็นต้องติดตั้งซอฟต์แวร์ Python ตั้งแต่รุ่น 2.3 และ Twisted ตั้งแต่รุ่น 1.3 หรือใหม่กว่า ส่วนการใช้งานโปรแกรม OctaveCoach และ OctaveLearner จำเป็นต้องติดตั้งโปรแกรม GNU Octave ตั้งแต่รุ่น 2.1.X

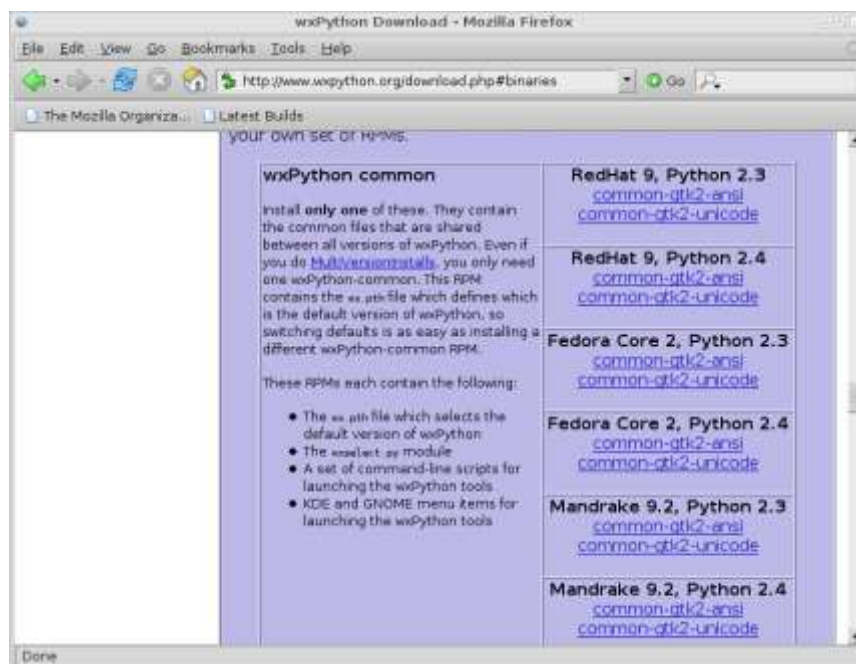
ซอฟต์แวร์ Python ตั้งแต่รุ่น 2.3 และ wxPython ตั้งแต่รุ่น 2.5.X หรือใหม่กว่า การติดตั้งซอฟต์แวร์ต่างๆในกรณีของระบบปฏิบัติการลินุกซ์แบบ Debian สามารถใช้การพิมพ์คำสั่งดังนี้บนเทอร์มินัลรับคำสั่ง

```
apt-get install octave python python-wxgtk2.6 python2.3-twisted
```

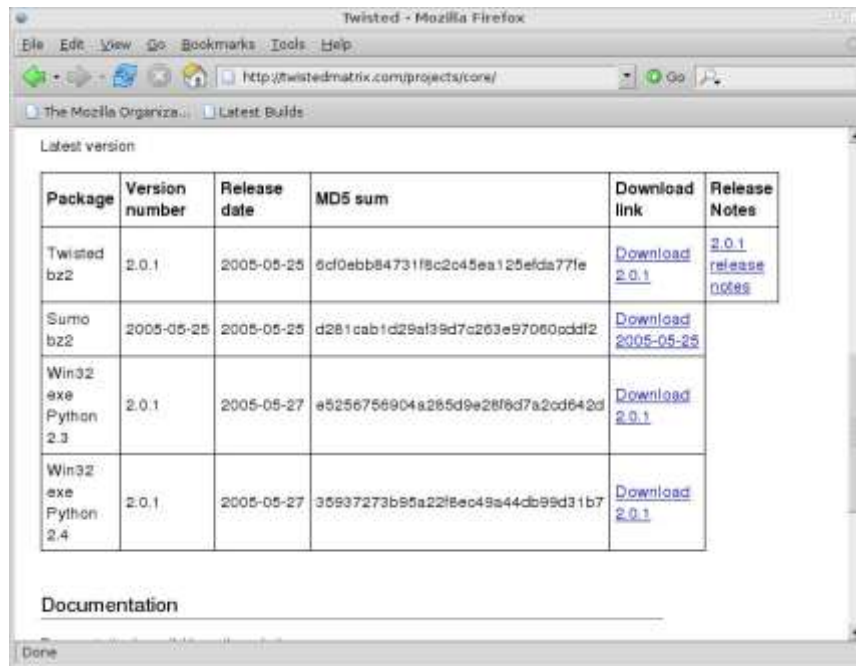
สำหรับระบบปฏิบัติการลินุกซ์อื่นๆ สามารถใช้การดาวน์โหลดไฟล์สำหรับการติดตั้งแบบ rpm จากโฮมเพจของผู้พัฒนาซอฟต์แวร์ดังในรูปที่ 4.1-4.3 ตามลำดับ



รูปที่ 4.1 การดาวน์โหลดไฟล์สำหรับติดตั้งซอฟต์แวร์ Python



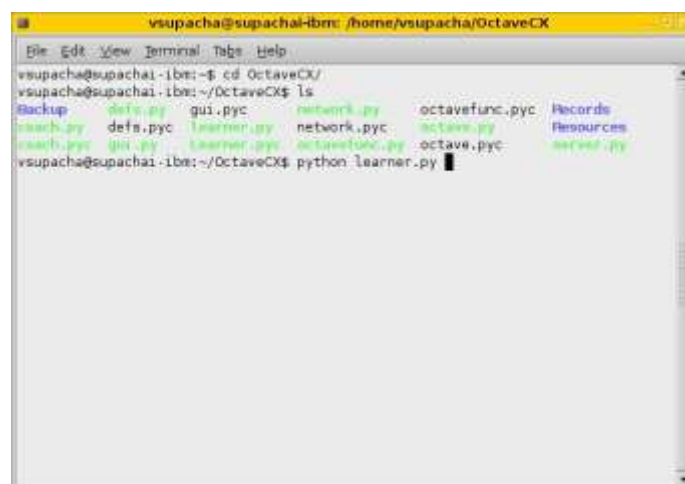
รูปที่ 4.2 การดาวน์โหลดไฟล์สำหรับติดตั้งซอฟต์แวร์ wxPython



รูปที่ 4.3 การดาวน์โหลดไฟล์สำหรับติดตั้งซอฟต์แวร์ Twisted

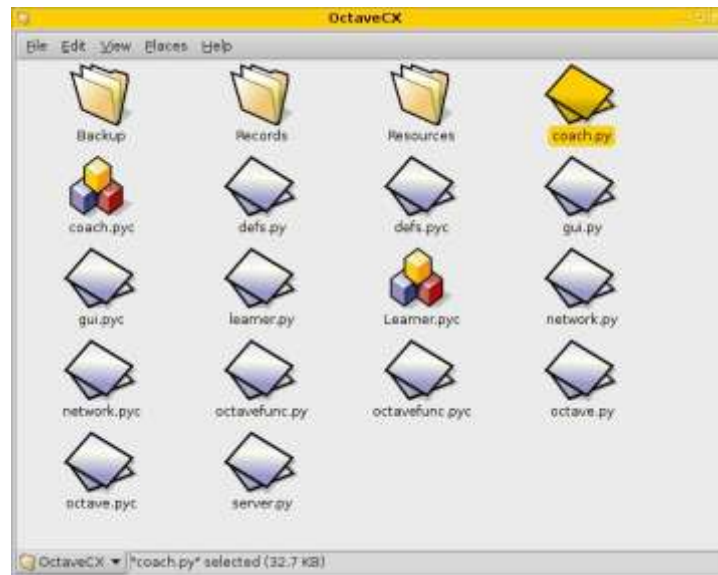
4.1.2 วิธีใช้งานพื้นฐาน

การเรียกใช้งานโปรแกรม OctaveServer OctaveCoach และ OctaveLearner บนระบบปฏิบัติการลินุกซ์สามารถทำได้ 2 วิธี วิธีแรกคือการเปิดหน้าต่างเทอร์มินัลรับคำสั่งแล้วทำการเปลี่ยนตำแหน่งไดเรกทอรีไปที่โฟลเดอร์ซึ่งมีโปรแกรมอยู่ จากนั้นจึงพิมพ์คำสั่ง **python xxxx** โดย XXXX คือ ชื่อไฟล์ server.py / coach.py / learner.py สำหรับโปรแกรม OctaveServer OctaveCoach และ OctaveLearner ตามลำดับ รูปที่ 4.4 แสดงถึงการพิมพ์คำสั่งเพื่อย้ายไปโฟลเดอร์ OctaveCX แล้วจึงเรียกใช้โปรแกรม OctaveLearner



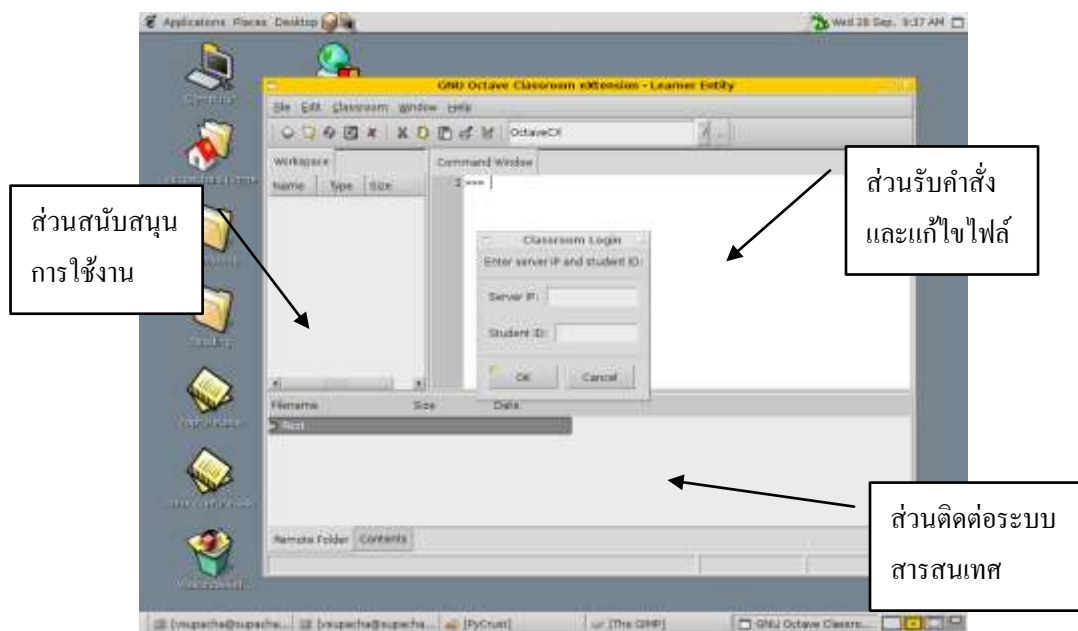
รูปที่ 4.4 การพิมพ์คำสั่งเพื่อเรียกใช้โปรแกรม OctaveLearner

นอกจากการเรียกใช้โปรแกรมด้วยการพิมพ์ในหน้าต่างเทอร์มินัลรับคำสั่งแล้ว ผู้ใช้งานอาจใช้คุณสมบัติของส่วนจัดการกราฟิก GNOME ในการแสดงหน้าต่างของโฟลเดอร์ที่จัดเก็บไฟล์ต่างๆของโปรแกรม แล้วจึงใช้เมาส์คลิกเลือกเพื่อเรียกใช้โปรแกรม ดังแสดงในรูปที่ 4.5



รูปที่ 4.5 การเรียกใช้งานโปรแกรม OctaveCoach ผ่าน GNOME

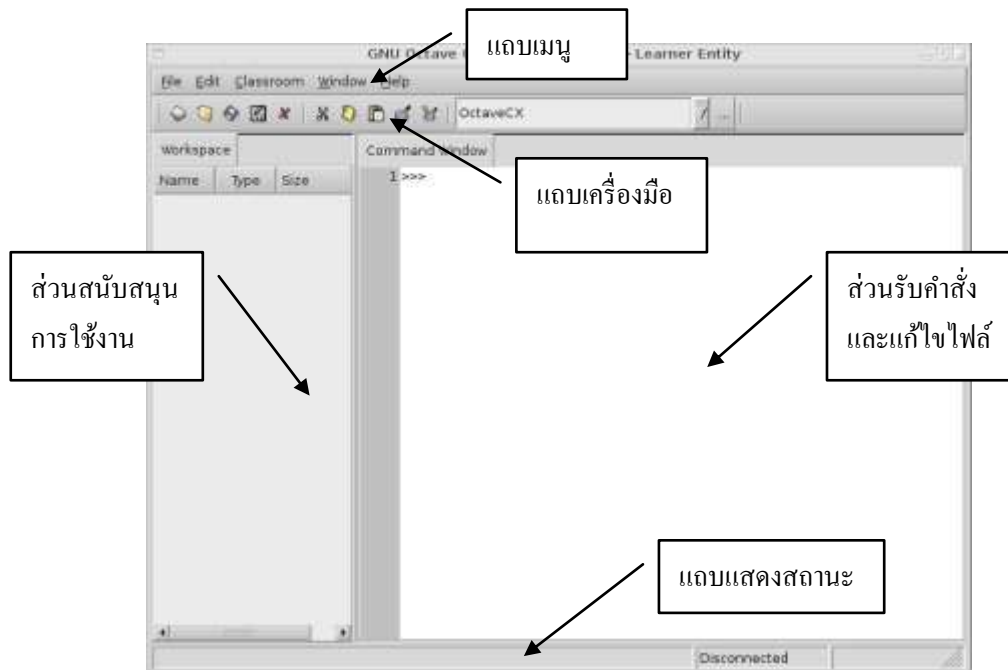
เมื่อผู้ใช้งานสั่งให้โปรแกรม OctaveCoach หรือ OctaveLearner ทำงาน โปรแกรมจะเริ่มต้นด้วยการแสดงหน้าต่างของโปรแกรม แล้วแสดงหน้าต่างย่อยสำหรับถามข้อมูลจากผู้ใช้งานเกี่ยวกับการเชื่อมต่อเข้าสู่เซิร์ฟเวอร์ เพื่อเข้าสู่การเป็นระบบสารสนเทศสำหรับสนับสนุนการเรียนการสอน ดังแสดงในรูปที่ 4.6



รูปที่ 4.6 การแสดงผลในขั้นเริ่มต้นของโปรแกรม OctaveLearner

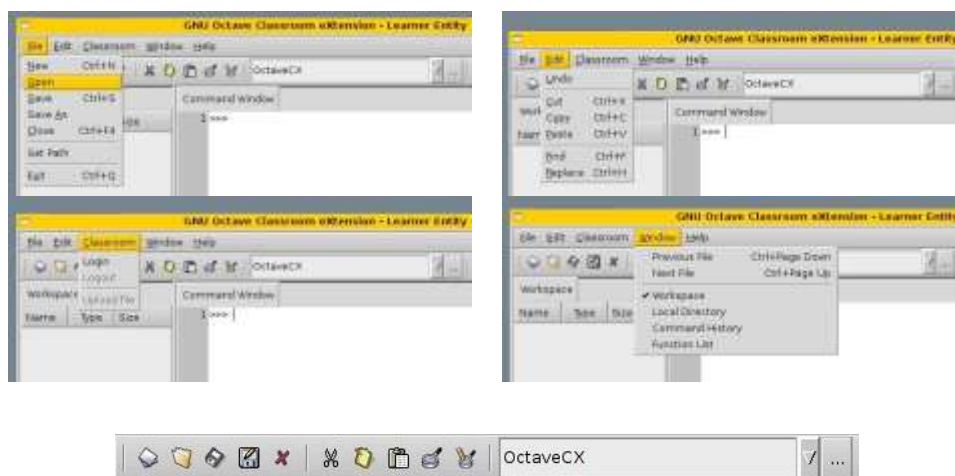
ในขั้นตอนนี้ โปรแกรม OctaveCoach จะถามถึงตำแหน่งไอพี (IP address) ของคอมพิวเตอร์ซึ่งโปรแกรม OctaveServer ทำงานอยู่ และรหัสผ่านสำหรับการเข้าสู่ระบบสารสนเทศในฐานะของผู้สอน ในขณะที่โปรแกรม OctaveLearner จะถามถึงตำแหน่งไอพี (IP address) ของคอมพิวเตอร์ซึ่งโปรแกรม OctaveServer ทำงานอยู่ และเลขประจำตัวหรือชื่อผู้ใช้ของผู้เรียนตามที่ผู้สอนได้กำหนดไว้ในฐานข้อมูลของ learner records ทั้งนี้ หากผู้ใช้งานทำการกดปุ่ม Esc หรือใช้เมาส์คลิกเลือกที่ปุ่ม Cancel ก็จะเป็นการยกเลิกการเข้าสู่ระบบ

สารสนเทศ ซึ่งส่วนติดต่อระบบสารสนเทศก็จะปิดลงโดยอัตโนมัติ พร้อมทั้งมีการแจ้งสถานะการติดต่อเซิร์ฟเวอร์ไว้ที่แถบแสดงสถานะ ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 หน้าต่างของโปรแกรม OctaveLearner ในกรณีไม่อยู่ในระบบสารสนเทศ

ผู้ที่สนใจเรียนรู้การใช้งานโปรแกรม GNU Octave หรือต้องการนำโปรแกรม GNU Octave ไปใช้แก้ปัญหาทางคณิตศาสตร์ ก็สามารถใช้โปรแกรม OctaveCoach และ OctaveLearner ทำหน้าที่เป็นส่วนติดต่อผู้ใช้แบบกราฟฟิกของโปรแกรม GNU Octave ถึงแม้จะไม่อยู่ในระบบสารสนเทศ หน้าต่างในรูปที่ 4.7 แสดงให้เห็นถึงองค์ประกอบต่างๆของส่วนติดต่อผู้ใช้แบบกราฟฟิก ได้แก่ ส่วนจัดการหน้าต่างย่อยที่เกี่ยวข้องกับการรับคำสั่งและการแก้ไขไฟล์ ส่วนจัดการหน้าต่างย่อยที่สนับสนุนการใช้งาน แถบเมนู แถบเครื่องมือ และแถบแสดงสถานะ การใช้งานโปรแกรม OctaveCoach และ OctaveLearner สามารถใช้เมาส์ในการเลือกการทำงานจากเมนูต่างๆ หรือแถบเครื่องมือ ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 หัวข้อต่างๆของแถบเมนูและแถบเครื่องมือ

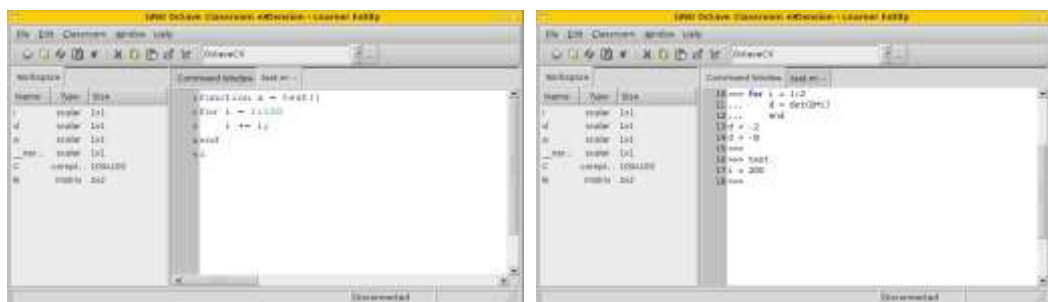
4.1.3 การสั่งงานโปรแกรม GNU Octave

การสั่งโปรแกรม GNU Octave ประมวลผลทางคณิตศาสตร์ ผ่านทางโปรแกรม OctaveCoach หรือ OctaveLearner จะใช้การพิมพ์คำสั่งในหน้าต่าง Command Window ด้วยรูปแบบเดียวกับการป้อนคำสั่งให้กับ การพิมพ์คำสั่งให้โปรแกรม GNU Octave โดยตรง ดังแสดงเปรียบเทียบไว้ในรูปที่ 4.9 สิ่งที่แตกต่างกันคือ โปรแกรม OctaveCoach และ OctaveLearner จะมีหน้าต่างย่อยที่อำนวยความสะดวกในการใช้งาน เช่น หน้าต่างย่อย Workspace ในรูปที่ 4.9 แสดงรายละเอียดของตัวแปรต่างๆที่โปรแกรม GNU Octave รับรู้ จากการป้อน คำสั่งต่างๆเข้าไป ในขณะที่ผู้ใช้โปรแกรม GNU Octave จะต้องพิมพ์คำสั่ง **who** เพื่อขอดูรายละเอียดของตัวแปรต่างๆเป็นครั้งๆไป



รูปที่ 4.9 การป้อนคำสั่งให้โปรแกรม OctaveLearner และ GNU Octave

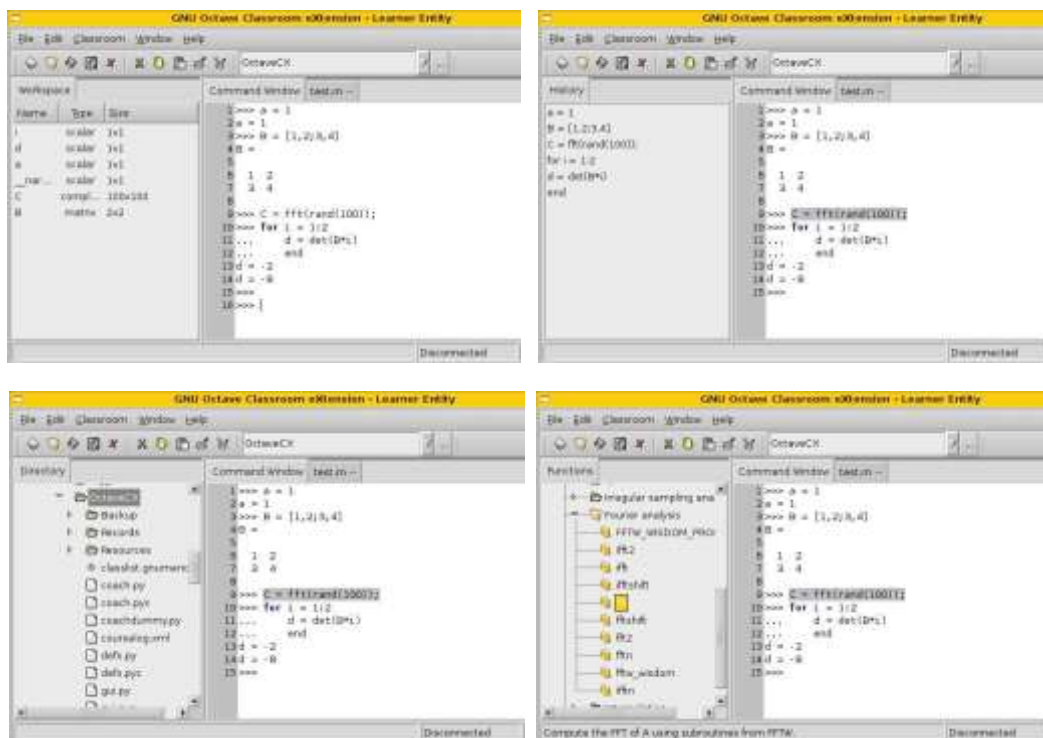
นอกจากนี้ โปรแกรม OctaveCoach และ OctaveLearner สามารถสร้างไฟล์ใหม่หรืออ่านไฟล์ชุดคำสั่งที่มีนามสกุล .m ในระหว่างการพัฒนาคำสั่งใหม่จากการใช้ชุดคำสั่งเพื่อประมวลผลตามสมการหรือลำดับการคำนวณที่ซับซ้อนได้ ไฟล์ของชุดคำสั่งที่ได้แก้ไขแล้ว เมื่อทำการบันทึก ก็จะสามารถเรียกใช้ด้วยการพิมพ์คำสั่งในหน้าต่างรับคำสั่ง เช่นในตัวอย่างของการสร้างและทดสอบไฟล์ test.m ในรูปที่ 4.10 ในขณะที่การสร้างหรือแก้ไขไฟล์ชุดคำสั่งสำหรับโปรแกรม GNU Octave จะต้องเรียกใช้โปรแกรมสำหรับแก้ไขไฟล์ตัวอักษรต่างหาก เช่น โปรแกรม Notepad โปรแกรม gedit เป็นต้น ซึ่งอาจพบความไม่สะดวกในกรณีที่โปรแกรมเหล่านั้นทำการบันทึกไฟล์ลงในโฟลเดอร์ต่างจากโปรแกรม GNU Octave ทำให้ไม่สามารถเรียกใช้คำสั่งที่สร้างขึ้นได้



รูปที่ 4.10 การแก้ไขและเรียกใช้ไฟล์ชุดคำสั่ง test.m

หน้าต่างย่อยที่อำนวยความสะดวกในการประมวลผลด้วยโปรแกรม GNU Octave มีอยู่ 4 ประเภท คือ หน้าต่าง Workspace สำหรับแสดงรายละเอียดของตัวแปร หน้าต่าง History สำหรับแสดงคำสั่งต่างๆที่ป้อนไปแล้ว หน้าต่าง Directory สำหรับแสดงรายละเอียดของไฟล์และโฟลเดอร์ต่างๆ และหน้าต่าง Functions สำหรับ

แสดงรายละเอียดของฟังก์ชันและคำสั่งต่างๆที่ใช้ได้ การเรียกให้หน้าต่างย่อยเหล่านี้ขึ้นมาแสดงผล ใช้การเลือกจากเมนู Window โดยอาจเลือกให้เปิดพร้อมๆกันได้ แล้วจึงใช้แถบในการเลือกหน้าต่างย่อยที่จะแสดงผล



รูปที่ 4.11 หน้าต่างย่อยต่างๆที่อำนวยความสะดวกในการป้อนคำสั่ง

4.2 ขั้นตอนสำหรับผู้สอน

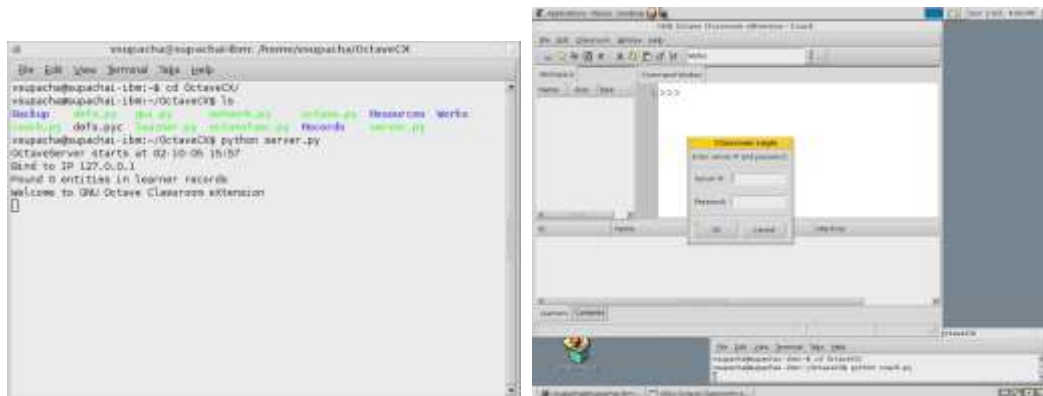
4.2.1 การเริ่มต้นระบบสารสนเทศ

ขั้นตอนแรกในการเริ่มต้นระบบสารสนเทศเพื่อสนับสนุนการเรียนการสอน คือ การสั่งให้โปรแกรม OctaveServer ทำงานบนเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ ด้วยการพิมพ์คำสั่ง

python server.py

ที่หน้าต่างของเทอร์มินัลรับคำสั่ง หรือใส่คำสั่งนี้ลงในสคริปต์ซึ่งถูกเรียกใช้เมื่อคอมพิวเตอร์เริ่มต้นทำงาน เช่น ในไฟล์ภายใต้ระบบไดเรกทอรี rc.d ของระบบปฏิบัติการลินุกซ์ โปรแกรม OctaveServer จะทำงานตรวจสอบไดเรกทอรี Records เพื่อรวบรวมข้อมูลเกี่ยวกับผู้เรียนที่ลงทะเบียนไว้ จากนั้นจึงสร้างบริการบนพอร์ตสื่อสาร TCP/IP หมายเลข 8000 และ 8010 ด้วยโปรโตคอล HTTP และ XML-RPC ตามลำดับ จากนั้นผู้สอนทำการเรียกใช้โปรแกรม OctaveCoach และป้อนหมายเลขไอพีและรหัสผ่าน ตามที่กำหนดไว้ในไฟล์ server.py เพื่อเข้าสู่ ความแตกต่างระหว่างโปรแกรม OctaveCoach และ OctaveLearner ที่เห็นได้อย่างชัดเจนคือ โปรแกรม OctaveCoach จะใช้ขนาดของตัวอักษรที่ใหญ่เป็นพิเศษในหน้าต่างรับคำสั่งและหน้าต่างย่อยสำหรับแก้ไขไฟล์ เพื่อให้สามารถเห็นได้ชัดเมื่อใช้งานร่วมกับเครื่องฉายภาพ นอกจากนี้ หน้าต่างย่อยสำหรับติดต่อระบบสารสนเทศของโปรแกรม OctaveCoach ที่จะเปิดไว้อย่างตลอดเวลาที่ยังติดต่อกับระบบสารสนเทศคือ หน้าต่างย่อย Learners ที่แสดงสถานะของผู้เรียนทั้งหมดที่ลงทะเบียนไว้ และหน้าต่างย่อย Contents ที่แสดงรายการของไฟล์ของเนื้อหาทั้งหมดที่ผู้สอนสามารถกระจายไปสู่ผู้เรียนได้ หน้าต่างของเทอร์มินัลที่เรียกใช้

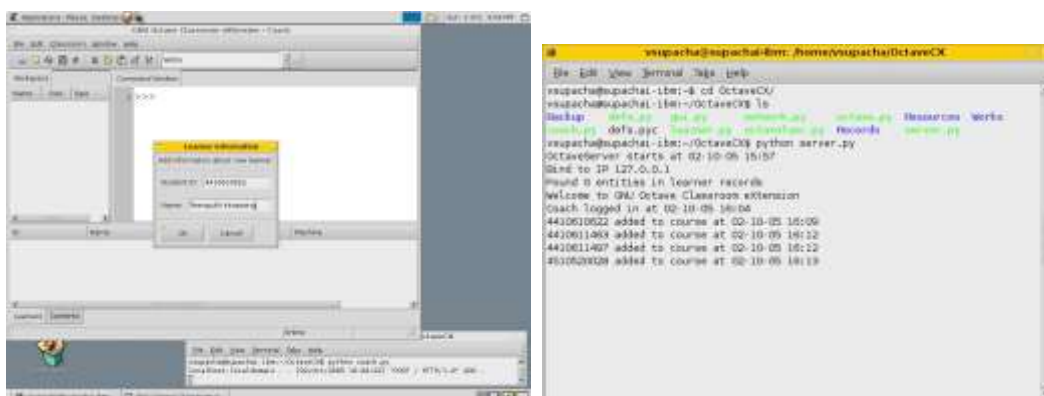
โปรแกรม OctaveServer และหน้าต่างของโปรแกรม OctaveCoach ซึ่งแสดงส่วนจัดการหน้าต่างย่อยสำหรับติดต่อระบบสารสนเทศ แสดงไว้ในรูปที่ 4.12



รูปที่ 4.12 การเรียกใช้โปรแกรม OctaveServer และ OctaveCoach เพื่อเริ่มต้นระบบสารสนเทศ

4.2.2 การจัดการข้อมูลใน Learning Records

เมื่อผู้สอนทำการเข้าสู่ระบบในฐานะ coach แล้ว ผู้สอนสามารถจัดการโครงสร้างของ learner records ด้วยการเพิ่มเติมหรือลบรายการของผู้เรียน การเพิ่มผู้เรียนเข้าไปในโครงสร้างของ learner records ใช้การสั่งผ่านเมนู Classroom -> Add Learner ซึ่งจะมีการแสดงหน้าต่างมารับข้อมูลพื้นฐาน ได้แก่ รหัสทะเบียน และชื่อของผู้เรียน โปรแกรม OctaveServer จะรายงานกิจกรรมที่สำคัญได้แก่ การเพิ่มและลบผู้เรียน การเข้าและออกระบบของทั้งผู้สอนและผู้เรียน พร้อมทั้งระยะเวลาที่เกิดกิจกรรมดังกล่าวในระบบสารสนเทศ ออกทางหน้าจอเทอร์มินัลที่เรียกใช้ รูปที่ 4.13 แสดงการป้อนข้อมูลของผู้เรียนคนใหม่ด้วยโปรแกรม OctaveCoach และการรายงานสถานะการเพิ่มผู้เรียนภายในเทอร์มินัลซึ่งเรียกใช้งานโปรแกรม OctaveServer

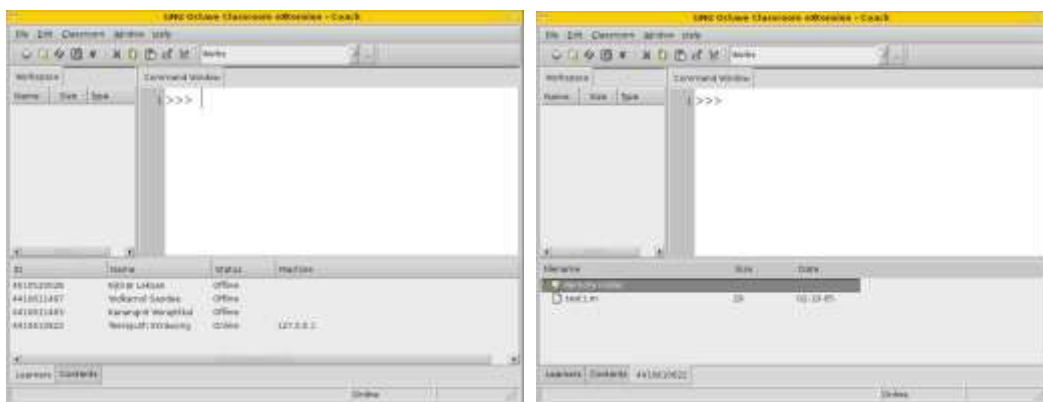


รูปที่ 4.13 การรับข้อมูลเกี่ยวกับผู้เรียนใหม่

ระบบสารสนเทศจะใช้รหัสทะเบียนในการจำแนกผู้เรียน และสร้างไคเร็กทอรีและไฟล์ที่เกี่ยวข้องได้แก่ info.xml ตามรหัสทะเบียนภายใต้โฟลเดอร์ Records สำหรับไฟล์ที่บันทึกสถานะการเรียนรู้ของผู้เรียนได้แก่ session.xml และ history.xml จะถูกสร้างขึ้นโดยอัตโนมัติ เมื่อผู้เรียนทำการเข้าสู่ระบบเป็นครั้งแรก

โปรแกรม OctaveCoach จะแสดงข้อมูลพื้นฐานและสถานะของผู้เรียนในการเชื่อมต่อกับระบบสารสนเทศ ในหน้าต่างย่อย Learners ที่เปิดขึ้นเมื่อผู้สอนเข้าสู่ระบบ รายการของผู้เรียนที่แสดงในหน้าต่าง

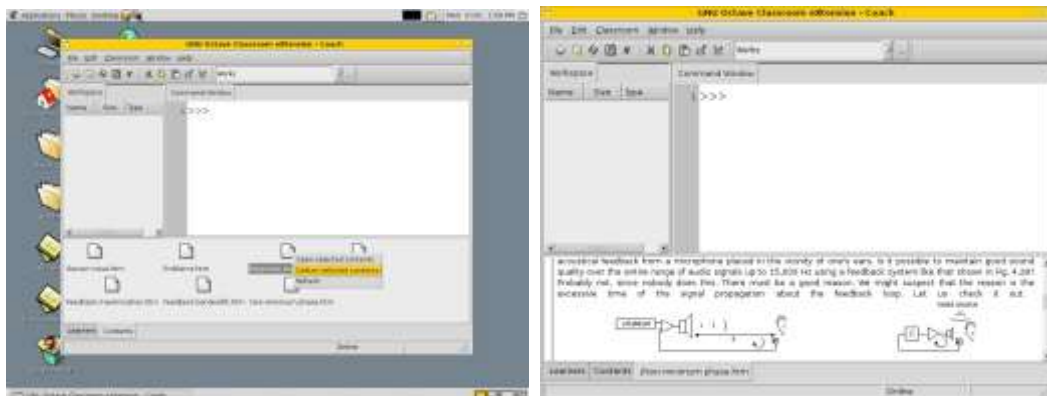
ข้อ Learners จะปรับปรุงใหม่โดยอัตโนมัติ เมื่อมีผู้เรียนเข้าหรือออกจากระบบ รวมไปถึงการเพิ่มชื่อและลบชื่อออกจากระบบ ภายใต้นหน้าต่างข้อ Learners ผู้สอนสามารถใช้การกดซ้ำในขณะที่ใช้เมาส์เลื่อนตัวชี้ไปที่รายการของผู้เรียน เพื่อทำการแสดงหน้าต่างข้อใหม่สำหรับแสดงรายการของไฟล์ทั้งหมดภายใต้โฟลเดอร์ของผู้เรียนคนนั้นบนเครื่องเซิร์ฟเวอร์ หน้าต่างข้อที่เปิดขึ้นใหม่จะใช้รหัสทะเบียนของผู้เรียนเป็นชื่อของหน้าต่าง โดยผู้สอนสามารถดาวน์โหลดไฟล์จากโฟลเดอร์ของผู้เรียนมาที่โฟลเดอร์ทำงานบนเครื่องคอมพิวเตอร์ของผู้สอน ด้วยการเลื่อนตัวชี้ไปที่ชื่อของไฟล์และกดปุ่มซ้ายของเมาส์ซ้ำ อนึ่งผู้สอนไม่สามารถส่งไฟล์จากเครื่องคอมพิวเตอร์ของผู้สอนไปที่โฟลเดอร์ของผู้เรียนผ่านทางโปรแกรม OctaveCoach ได้ โดยผู้สอนจะต้องไปที่เครื่องเซิร์ฟเวอร์และใช้การคัดลอกไฟล์โดยตรงไปที่โฟลเดอร์ของผู้เรียนภายใต้โครงสร้างไดเรกทอรี Records รูปที่ 4.14 แสดงหน้าต่างข้อ Learners และหน้าต่างข้อที่แสดงไฟล์ของผู้เรียน ภายใต้นหน้าต่างของโปรแกรม OctaveCoach



รูปที่ 4.14 การแสดงรายละเอียดของผู้เรียนใน โปรแกรม OctaveCoach

4.2.3 การกระจายเนื้อหาใน Learning Resources

ผู้สอนสามารถจัดการเนื้อหาสำหรับใช้ประกอบการสอนผ่านทางหน้าต่างข้อ Contents ซึ่งเปิดขึ้นพร้อมกับหน้าต่างข้อ Learners หน้าต่างข้อ Contents จะแสดงชื่อของไฟล์ทั้งหมดภายใต้โครงสร้างไดเรกทอรี Resources บนเครื่องเซิร์ฟเวอร์ โดยไม่รวมถึงไดเรกทอรีข้อซึ่งส่วนใหญ่จะใช้เก็บไฟล์รูปภาพต่างๆ ประกอบไฟล์ HTML



รูปที่ 4.15 การจัดการเนื้อหาประกอบการสอนด้วย โปรแกรม OctaveCoach

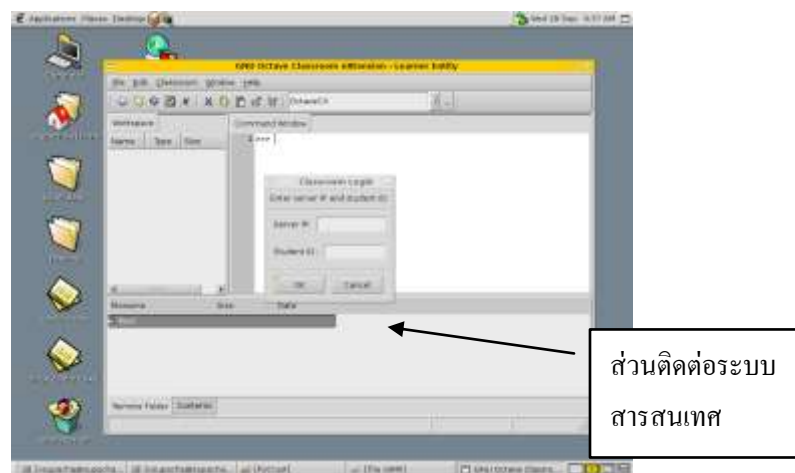
การเลือกเนื้อหาที่จะกระจายไปสู่ผู้เรียน ใช้การเลื่อนตัวชี้ไปเลือกไฟล์ที่ต้องการ กดปุ่มขวาของเมาส์เพื่อเปิดเมนูย่อย และทำการเลือกเมนู Deliver selected contents ในกรณีที่ผู้สอนต้องการดูเนื้อหาดังกล่าวบนเครื่องคอมพิวเตอร์ของผู้สอนเอง ก็สามารถทำได้ด้วยการกดซ้ำที่ไฟล์ดังกล่าว

เมื่อผู้สอนเลือกไฟล์ของเนื้อหาที่จะกระจายไปสู่ผู้เรียน โปรแกรม OctaveCoach จะส่งคำร้องขอผ่านทางเครือข่ายไปที่บริการของกระบวนการ delivery บนเครื่องเซิร์ฟเวอร์ จากนั้น เซิร์ฟเวอร์จะแจ้งรายละเอียดของเนื้อหาในรูปของ URL ไปสู่คอมพิวเตอร์ของผู้เรียนที่เชื่อมต่อกับระบบสารสนเทศผ่านทางโปรแกรม OctaveLearner ทั้งหมด ซึ่งโปรแกรม OctaveLearner จะตอบสนองด้วยการเปิดหน้าต่างย่อยที่แสดงเนื้อหาดังกล่าวโดยอัตโนมัติ โปรแกรม OctaveServer จะบันทึกคำร้องขอในการกระจายไฟล์เนื้อหาดังกล่าวในไฟล์ courselog.xml เพื่อให้ผู้สอนสามารถตรวจสอบในภายหลังถึงชื่อ ลำดับ และเวลาของเนื้อหาที่ใช้ประกอบการสอนทั้งหมด

4.3 ขั้นตอนสำหรับผู้เรียน

4.3.1 การเข้าสู่ระบบสารสนเทศ

ผู้เรียนเข้าสู่ระบบสารสนเทศในฐานะ learner entity ด้วยการเรียกใช้โปรแกรม OctaveServer และป้อนหมายเลขไอพีของเครื่องเซิร์ฟเวอร์และรหัสทะเบียนของนักศึกษา โปรแกรม OctaveLearner แบ่งพื้นที่บริเวณด้านล่างของหน้าต่างหลัก เพื่อแสดงการติดต่อระบบสารสนเทศ เมื่อเข้าสู่ระบบและปิดลงโดยอัตโนมัติเมื่อออกจากระบบ ภายในพื้นที่ดังกล่าวจะมีการแสดงหน้าต่างย่อยต่างๆ ซึ่งหน้าต่างย่อย Remote Folder และ Contents จะเป็นหน้าต่างที่แสดงตลอดเวลา

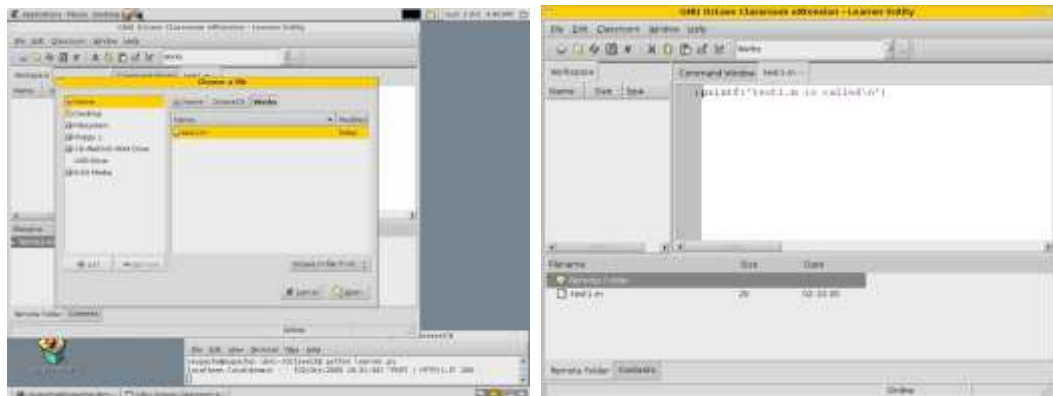


รูปที่ 4.16 การเข้าสู่ระบบสารสนเทศในฐานะ learner entity ด้วยโปรแกรม OctaveLearner

4.3.2 การจัดการข้อมูลบนเซิร์ฟเวอร์

ผู้สอนมีสิทธิในการจัดการข้อมูลที่จัดเก็บในโครงสร้าง learner records ที่เกี่ยวข้องกับตนเองได้อย่างมีข้อจำกัด โดยจะสามารถร้องขอหรือเพิ่มเติมไฟล์ในโฟลเดอร์ของผู้เรียนภายใต้ไคลเอนต์ Records บนเครื่องเซิร์ฟเวอร์เท่านั้น ไฟล์ข้อมูลของผู้เรียนสามารถร้องขอหรือปรับเปลี่ยนได้จะไม่รวมถึงไฟล์สำหรับบันทึกสถานะของผู้เรียน ได้แก่ info.xml session.xml และ history.xml รวมทั้งผู้เรียนไม่สามารถลบหรือแก้ไขชื่อของไฟล์ได้ หน้าต่างย่อย Remote Folder จะแสดงถึงรายละเอียดของไฟล์ทั้งหมดในโฟลเดอร์ของผู้เรียนบน

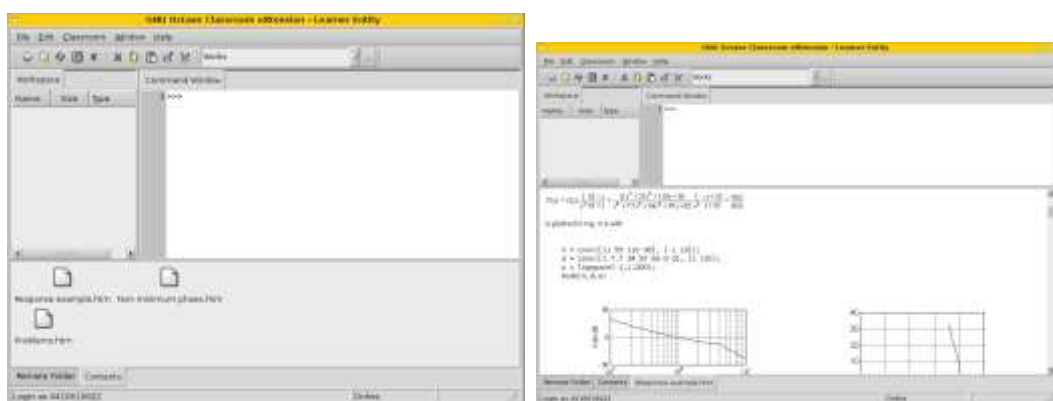
เครื่องเซิร์ฟเวอร์ โดยรายการของไฟล์จะถูกปรับปรุงโดยอัตโนมัติเมื่อผู้เรียนส่งไฟล์ไปที่เครื่องเซิร์ฟเวอร์ ผู้เรียนสามารถดาวน์โหลดไฟล์จากเครื่องเซิร์ฟเวอร์มายังโฟลเดอร์ทำงานบนเครื่องคอมพิวเตอร์ของผู้เรียน ด้วยการเลื่อนตัวชี้ไปที่ชื่อของไฟล์ดังกล่าวและกดปุ่มซ้าย ซึ่งโปรแกรม OctaveLearner จะถามยืนยันในกรณีที่มีไฟล์ดังกล่าวอยู่แล้วในโฟลเดอร์ทำงาน การส่งไฟล์ไปจัดเก็บในโฟลเดอร์ของผู้เรียนบนเครื่องเซิร์ฟเวอร์ใช้การเลือกที่เมนู Classroom -> Upload file จากนั้นเลือกไฟล์ที่ต้องการส่ง การปรับปรุงไฟล์ใดบนเครื่องเซิร์ฟเวอร์ต้องทำการแก้ไขบนเครื่องคอมพิวเตอร์ของผู้เรียนก่อน แล้วจึงส่งไฟล์ที่ชื่อซ้ำกันไปทดแทนไฟล์ดังกล่าวบนเครื่องเซิร์ฟเวอร์



รูปที่ 4.17 การส่งไฟล์ test.m ไปจัดเก็บบนเซิร์ฟเวอร์

4.3.3 การเข้าถึงเนื้อหาของการเรียนการสอน

ผู้เรียนสามารถร้องขอเนื้อหาผ่านทางหน้าต่างย่อย Contents ซึ่งจะแสดงชื่อของไฟล์เนื้อหาทั้งหมดที่ถูกผู้สอนกำหนดไว้ใช้ประกอบการสอน เมื่อผู้เรียนเลื่อนตัวชี้ไปที่รายการของไฟล์ดังกล่าวและกดปุ่มซ้าย โปรแกรม OctaveLearner จะทำการเปิดหน้าต่างย่อยใหม่ ร้องขอเนื้อหาของไฟล์ดังกล่าวผ่านทางโปรโตคอล HTTP มาที่เซิร์ฟเวอร์ และแสดงผลในรูปแบบ HTML



รูปที่ 4.18 การแสดงเนื้อหาประกอบการสอนในโปรแกรม OctaveLearner

บทที่ 5

สรุปงานวิจัย

5.1 สรุปผลงานวิจัย

งานวิจัยนี้ได้พัฒนาค้นแบบของระบบสารสนเทศสำหรับสนับสนุนการเรียนการสอน ซึ่งจำเป็นต้องประยุกต์ใช้ซอฟต์แวร์คอมพิวเตอร์ร่วมกับการเรียนรู้ทฤษฎี การพัฒนาระบบสารสนเทศได้ดำเนินการโดยอิงตามโครงสร้าง Learning Technology System Architecture ที่ระบุไว้ในมาตรฐาน IEEE 1484 โครงสร้างของระบบสารสนเทศประกอบจากซอฟต์แวร์ 3 ส่วน ได้แก่

1. โปรแกรม OctaveServer ทำหน้าที่ให้บริการสำหรับรองรับการร้องขอผ่านโปรโตคอล XML-RPC จากองค์ประกอบ coach และ learner entity จัดการฐานข้อมูลต่างๆตามนิยามขององค์ประกอบ learner records และ learning resources กระจายเนื้อ และบันทึกกิจกรรมต่างๆที่เกิดขึ้นในระบบสารสนเทศ
2. โปรแกรม OctaveCoach ทำหน้าที่เป็นส่วนเชื่อมต่อให้ผู้สอนได้เข้าสู่ระบบสารสนเทศในฐานะของ coach และอำนวยความสะดวกในการใช้งานด้วยส่วนติดต่อผู้ใช้แบบกราฟฟิก
3. โปรแกรม OctaveLearner ทำหน้าที่เป็นส่วนเชื่อมต่อให้ผู้เรียน นักศึกษา หรือนักเรียนได้เข้าสู่ระบบสารสนเทศในฐานะของ learner entity และอำนวยความสะดวกในการใช้งานด้วยส่วนติดต่อผู้ใช้แบบกราฟฟิก

ซอฟต์แวร์ทั้งหมดถูกพัฒนาขึ้นด้วยภาษา Python ร่วมกับเครื่องมือพัฒนาแบบเปิดเผยต้นฉบับได้แก่ wxPython และ Twisted ทั้งนี้ ส่วนติดต่อผู้ใช้แบบกราฟฟิกสามารถใช้งานได้ถึงแม้จะไม่เชื่อมต่อกับระบบสารสนเทศ ซึ่งจะเอื้อต่อการแจกจ่ายโปรแกรม GNU Octave เป็นซอฟต์แวร์ช่วยการศึกษาทางด้านวิศวกรรมศาสตร์ หรือศาสตร์ที่จำเป็นต้องใช้การประมวลผลทางคณิตศาสตร์ที่ซับซ้อน

นอกจากต้นแบบของระบบสารสนเทศแล้ว กลุ่มผู้วิจัยได้จัดเตรียมเอกสารในรูปแบบต่างๆที่สนับสนุนการใช้งานโปรแกรม GNU Octave ในการเรียนการสอนทางด้านวิศวกรรมไฟฟ้า ได้แก่ คู่มือโปรแกรม GNU Octave ฉบับภาษาไทย บทความแนะนำการใช้งานโปรแกรม GNU Octave เอกสารอธิบายการประยุกต์ใช้โปรแกรม GNU Octave เพื่อแก้ปัญหาทางวิศวกรรมไฟฟ้า และซอฟต์แวร์แบบ Flash สำหรับสอนการป้อนคำสั่งให้โปรแกรม GNU Octave ไฟล์ทั้งหมดที่เป็นผลลัพธ์จากโครงการวิจัยนี้ได้รวบรวมไว้ในแผ่น CD ที่แนบมา กับรายงานฉบับนี้ โดยมีโครงสร้างไดเรกทอรี ดังนี้

- OctaveCX ไฟล์ทั้งหมดสำหรับการใช้งานระบบสารสนเทศ
- Documents ไฟล์เอกสารของรายงานฉบับนี้ ไฟล์เอกสารของบทความแนะนำการใช้งานโปรแกรม GNU Octave และคู่มือโปรแกรม GNU Octave ฉบับภาษาไทย
- Problems ไฟล์เอกสารที่อธิบายการประยุกต์ใช้โปรแกรม GNU Octave เพื่อแก้ปัญหาทางวิศวกรรมไฟฟ้า
- Flash ไฟล์แบบ Flash สำหรับสอนการป้อนคำสั่งให้โปรแกรม GNU Octave

5.2 ข้อเสนอแนะ

ผลลัพธ์ของงานวิจัยนี้ ยังอยู่ในขั้นตอนเป็นต้นแบบเท่านั้น โดยยังมีปัญหาอีกหลายอย่างที่จำเป็นต้องแก้ไขและปรับปรุง เนื่องจากแนวความคิดในการพัฒนาซอฟต์แวร์แบบยั่งยืนซึ่งเป็นเป้าหมายหลักของโครงการวิจัยนี้ กลุ่มผู้วิจัยได้ดำเนินการเผยแพร่ซอฟต์แวร์ต่างๆผ่านทางเครือข่ายของผู้พัฒนาซอฟต์แวร์แบบเปิดเผยต้นฉบับ 2 แห่ง กล่าวคือ Sourceforge ที่เว็บเพจ <http://www.sf.net> ซึ่งเป็นเครือข่ายของผู้พัฒนาในระดับสากล และสมาคมสมาพันธ์โอเพนซอร์สแห่งประเทศไทย ที่เว็บเพจ <http://www.tosf.org/> ซึ่งเป็นเครือข่ายของผู้พัฒนาในประเทศไทย การที่ซอฟต์แวร์ของระบบสารสนเทศต้นแบบถูกเผยแพร่ด้วยลิขสิทธิ์แบบเปิดเผยต้นฉบับ จะเอื้อให้ผู้ที่เกี่ยวข้องกับการศึกษาได้นำไปใช้งาน ซึ่งมีผลต่อเนื่องในแง่ที่มีการทดสอบในแบบที่หลากหลายมากขึ้น ทำให้ทราบข้อจำกัดและปัญหาต่างๆ ที่เกิดขึ้นระหว่างการใช้งานด้วยรูปแบบที่แตกต่างกัน

สำหรับซอฟต์แวร์ในรุ่นที่รวมเข้ากับรายงานฉบับนี้ ยังมีปัญหาที่ยังไม่สามารถแก้ไขได้ และมีข้อจำกัดบางอย่างในการใช้งาน ซึ่งจะได้แก้ไขปรับปรุงต่อไป ได้แก่

1. หากผู้ใช้งานพิมพ์คำสั่งที่ทำให้โปรแกรม GNU Octave หยุดทำงาน ก็จะทำให้โปรแกรม OctaveCoach และ OctaveLearner มีการทำงานที่ผิดพลาดไป
2. โปรแกรม OctaveCoach และ OctaveLearner ไม่สามารถจัดการหรือปรับแต่งรูปภาพที่ได้จากการคำนวณ เนื่องจากรูปภาพทั้งหมดถูกจัดการด้วยโปรแกรม gnuplot ซึ่งได้รับข้อมูลต่อจากโปรแกรม GNU Octave
3. การเข้าสู่ระบบสารสนเทศยังไม่มีกลไกสำหรับตรวจสอบสิทธิการเข้าใช้อย่างรัดกุม
4. หากในระหว่างการทำงาน โปรแกรม OctaveServer เกิดความผิดพลาดขึ้น จนกระทั่งหยุดโปรแกรมเอง โปรแกรม OctaveCoach และ OctaveLearner จะมีการทำงานที่ผิดพลาดไป
5. การบันทึกสถานะทั้งหมด กระทำด้วยการบันทึกข้อมูลลงในไฟล์โดยทันที ดังนั้นหากข้อมูลที่บันทึกเดิมมีขนาดใหญ่ หรือมีผู้เรียนจำนวนมาก การทำงานของระบบก็จะช้าลงอย่างมีนัยสำคัญ
6. การทำงานของต้นแบบระบบสารสนเทศ ถูกออกแบบให้ทำงานสอดคล้องเฉพาะการดำเนินการสอนสำหรับรายวิชาเดียว กลุ่มเดียว และเริ่มต้นระบบใหม่เมื่อจะทำการเรียนการสอน ความสามารถของต้นแบบของระบบสารสนเทศ ยังไม่เหมาะสมกับการทำงานอย่างต่อเนื่อง หรือการใช้งานเพื่อรองรับการเรียนการสอนของหลายรายวิชา

บรรณานุกรม

1. Online, "The Mathworks: Developers of MATLAB and Simulink for Technical Computing",
<http://www.mathworks.com>
2. Stephen J. Chapman, "MATLAB Programming for Engineers 2nd edition", Brooks/Cole, 2002.
3. Delores M. Etter, "Engineering Problem Solving with MATLAB", Prentice-Hall, 1997.
4. John W. Eaton, "Octave: Past, Present, and Future", Proceedings of the 2nd International Workshop on Distributed Statistical Computing, 2001
5. "Open Source Initiative", <http://www.opensource.org>
6. "Opencontent", <http://www.opencontent.org>
7. IEEE Computer Society, IEEE 1484.1-2003 Standard for Learning Technology - Learning Technology Systems Architecture (LTSA), 2003.
8. Online, "PAPI Learner, Draft 8 Specification", <http://edutool.com/papi/>
9. Mehmet Sen, "Distributed Open Asynchronous Information Access Environment", Ph.D. dissertation, Syracuse University, 2000.
10. Alex Martelli, "Python in a Nutshell", O'Reilly, 2003.
11. Online, "wxPython", <http://www.wxpython.org/>
12. Online, "Twisted Matrix Laboratories", <http://twistedmatrix.com/>

ภาคผนวก ก

คู่มือการใช้งาน GNU Octave

ก.1 การสั่งการ GNU Octave

โปรแกรม GNU Octave คือ ซอฟต์แวร์ที่เป็นภาษาคอมพิวเตอร์ระดับสูง โดยมีจุดมุ่งหมายหลักในการคำนวณสมการและค่าตัวแปรต่างๆด้วยคอมพิวเตอร์ โดยเฉพาะการประมวลผลที่เกี่ยวข้องกับข้อมูลประเภทเวกเตอร์หรือเมตริกซ์ โปรแกรม GNU Octave รองรับวิธีการสั่งการด้วยการพิมพ์คำสั่ง เพื่อทำการแก้ปัญหของสมการเชิงเส้นและไม่เชิงเส้น รวมทั้งการคำนวณเชิงเลขแบบต่าง นอกจากนี้ โปรแกรม GNU Octave ยังสนับสนุนการทำงานแบบ batch ด้วย การเรียกใช้งาน GNU Octave กระทำด้วยการพิมพ์คำสั่ง “octave” ที่ส่วนรับคำสั่งของระบบปฏิบัติการนั้น โปรแกรม GNU Octave จะแจ้งข้อความเริ่มต้นและเครื่องหมาย prompt เพื่อแสดงว่าพร้อมที่จะรับการพิมพ์คำสั่งจากผู้ใช้ ในกรณีที่พบปัญหา ผู้ใช้สามารถพิมพ์ Ctrl + c เพื่อจบการทำงาน ณ ขณะนั้นของโปรแกรม GNU Octave ได้ พิมพ์ Ctrl + z เพื่อหยุดการทำงานชั่วคราว หรือใช้คำสั่ง quit หรือ exit เพื่อออกจากโปรแกรม

คำสั่งพื้นฐานที่มีอยู่ในโปรแกรม GNU Octave ประกอบด้วยคำสั่งสำหรับการคำนวณทางพีชคณิตของเวกเตอร์และเมตริกซ์ การแก้สมการอนุพันธ์ด้วยวิธีเชิงเลข และการจัดเก็บข้อมูล โปรแกรม GNU Octave สามารถแสดงรูปภาพทั้งแบบ 2 มิติและ 3 มิติ โดยใช้งานร่วมกับโปรแกรม GNUPlot ซึ่งเป็นซอฟต์แวร์แบบเปิดเผยต้นฉบับที่พัฒนาสำหรับแสดงผลข้อมูลในรูปของกราฟแบบต่างๆ หากผู้ใช้ต้องการเรียกใช้ฟังก์ชันสำหรับการคำนวณเฉพาะทาง จะใช้การเรียกชุดคำสั่งจาก Toolbox ซึ่งเป็นกลุ่มของคำสั่งที่ถูกรวบรวมไว้สำหรับการคำนวณทางสาขาต่างๆ เช่น ทางสถิติ ทางวิศวกรรมสาขาต่างๆ ฯลฯ นอกจากคำสั่งที่มีใน GNU Octave และ Toolbox แล้ว ผู้ใช้ยังสามารถดัดแปลง แก้ไข หรือเขียนคำสั่งใหม่จากชุดของคำสั่งที่มีอยู่เดิม เพื่อให้มีความเหมาะสมกับจุดมุ่งหมายของงานได้ การเขียนคำสั่งใหม่สามารถกระทำโดยการสร้างไฟล์ประเภทตัวอักษรแล้วบันทึกด้วยนามสกุล .m หรือที่เรียกว่า m-file ทั้งนี้ โปรแกรม GNU Octave จะทำการประมวลผลตามลำดับของอัลกอริทึมที่เขียนใน m-file ที่ผู้ใช้เขียนไว้ และแจ้งทันทีเมื่อเกิดความผิดพลาดขึ้น รูปแบบของการประมวลผล m-file ดังกล่าวนี้อาจสะดวกต่อการพัฒนาและง่ายต่อการเข้าใจ ทำให้ผู้ใช้สามารถพัฒนาคำสั่งใหม่ๆ ได้อย่างรวดเร็วและใช้เวลาในการเรียนรู้สั้นน้อยเมื่อเทียบกับการใช้ภาษาคอมพิวเตอร์ทั่วไป เช่น ภาษา C Pascal หรือ FORTRAN

สำหรับเนื้อหาของการใช้งานโปรแกรม GNU Octave ต่อไปนี้ การแสดงคำสั่งซึ่งถูกพิมพ์ที่เครื่องหมายเตรียมพร้อมจะใช้ตัวอักษรเอนและหนา และใช้สัญลักษณ์ ↵ แทนการกดปุ่ม Enter หลังการพิมพ์คำสั่ง

ก.2 ตัวอย่างการใช้งานเบื้องต้น

การสร้างข้อมูลประเภทเมตริกซ์และเก็บค่าในรูปของตัวแปร เพื่อนำมาใช้อ้างอิงภายหลัง ใช้การพิมพ์คำสั่ง ดังนี้

```
octave : 1 > a = [ 1, 1, 2, 3 ; 5, 8, 13, 21 ] ↵
```

โปรแกรม GNU Octave จะตอบสนองด้วยการแสดงเมตริกซ์ในรูปของตัวเลขภายในคอลัมน์ที่จัดตำแหน่ง การพิมพ์เครื่องหมาย ; (semicolon) ในช่วงจบการพิมพ์คำสั่ง จะเป็นการแจ้งให้โปรแกรม GNU Octave ไม่พิมพ์ข้อความเพื่อตอบสนองต่อคำสั่งนั้น

```
octave : 2 > b = rand ( 3,2 ) ; ↵
```

จะทำให้เกิดข้อมูลประเภทเมตริกซ์ขนาด 3 แถว 2 คอลัมน์ โดยที่ค่าของสมาชิกแต่ละตัวนั้น จะสุ่มมาจาก 0 ถึง 1 การแสดงค่าของตัวแปร ใช้การพิมพ์คำสั่งในรูปของชื่อของตัวแปรที่ต้องการทราบค่า

```
octave : 2 > b ↵
```

โปรแกรม GNU Octave ได้เตรียมตัวดำเนินการต่างๆ ที่ช่วยเพิ่มความสะดวกในการคำนวณเกี่ยวกับการคำนวณเมตริกซ์ เช่น ถ้าต้องการคูณเมตริกซ์ a ด้วยค่าแบบสเกลาร์ ให้พิมพ์คำสั่งดังนี้

```
octave : 4 > 2 * a ↵
```

ในกรณีต้องการคูณระหว่างเมตริกซ์ a และ b ให้พิมพ์คำสั่งดังนี้

```
octave : 5 > a * b ↵
```

ในกรณีต้องการค่า transpose ของเมตริกซ์ a คูณกับเมตริกซ์ a ให้พิมพ์คำสั่งดังนี้

```
octave : 6 > a' * a ↵
```

การแก้หาคำตอบของกลุ่มของสมการเชิงเส้นในรูปของ $ax = b$ สามารถใช้ตัวดำเนินการ “\” เพื่อทำการคำนวณแบบ left division

```
octave : 7 > a \ b ↵
```

ซึ่งมีความหมายเดียวกับการหาคำตอบของสมการ $x = a^{-1}b$ แต่จะใช้การคำนวณที่หลีกเลี่ยงการหาค่าผกผันของเมตริกซ์ a ในกรณีที่เมตริกซ์ a เป็นเมตริกซ์แบบ singular โปรแกรม GNU Octave จะทำการแจ้งเตือนพร้อมทั้งแสดงคำตอบในแบบ minimum norm

โปรแกรม GNU Octave มีฟังก์ชันพื้นฐานสำหรับแก้ปัญหของสมการอนุพันธ์แบบไม่เชิงเส้นในรูปของ

$$\frac{dx}{dt} = f(x,t) \quad \text{และ} \quad x(t = t_0) = x_0$$

การที่โปรแกรม GNU Octave จะหาคำตอบด้วยการอินทิเกรตสมการข้างต้นนี้ ผู้ใช้จะต้องป้อนนิยามของฟังก์ชัน $f(x,t)$ ซึ่งอาจทำได้ด้วยการป้อนรายละเอียดของแต่ละฟังก์ชันด้วยการพิมพ์คำสั่งที่ละบรรทัด สำหรับตัวอย่างนี้เราจะพิมพ์คำสั่งที่เกี่ยวข้อง โดยให้ฟังก์ชันที่เราสนใจอยู่ทางขวามือของสมการ ทั้งนี้ ขณะที่ป้อน

ฟังก์ชันเข้าไป โปรแกรม GNU Octave จะตอบกลับด้วยเครื่องหมายเตรียมพร้อมที่แตกต่างจากปกติ เพื่อแสดงว่ากำลังรอให้ป้อนรายละเอียดจนเสร็จ

```
octave:8 > function xdot = f(x, t) ↵
>> r = 0.25; ↵
>> k = 1.4; ↵
>> a = 1.5; ↵
>> b = 0.16; ↵
>> c = 0.9; ↵
>> d = 0.8; ↵
>> xdot(1) = r*x(1)*(1 - x(1)/k) - a*x(1)*x(2)/(1 + b*x(1)); ↵
>> xdot(2) = c*a*x(1)*x(2)/(1 + b*x(1)) - d*x(2); ↵
>> endfunction ↵
```

เมื่อกำหนดให้เงื่อนไขเริ่มต้นเป็น

```
x0 = [1; 2];
```

แล้วจะให้ output เวลาเป็น column vector

```
t = linspace(0, 50, 200)';
```

จะทำให้การง่ายขึ้น จากการ set differential Equation

```
x = lsode("f", x0, t);
```

ภาคผนวก ข

บทความที่ตีพิมพ์ในวารสารส่งเสริมเทคโนโลยี

Octave หนึ่งในซอฟต์แวร์โอเพนซอร์สเพื่องานทางวิศวกรรม

โอเพนซอร์ส (Open source) ถือได้ว่าเป็นคำที่รู้จักกันทั่วไปในฐานะทางเลือกใหม่ของซอฟต์แวร์ที่ใช้ได้ฟรีโดยไม่ละเมิดลิขสิทธิ์ การบูมของคำนี้คงต้องยกความดีให้กับลินุกซ์ทะเลและออฟฟิศทะเลซึ่งรู้จักกันแพร่หลายพร้อมกับยุคของคอมพิวเตอร์เอื้ออาทร ผู้เขียนเองก็อยากเห็นการใช้งานซอฟต์แวร์โอเพนซอร์สขงมาขึ้นจึงขอใช้บทความนี้เพื่อให้ข้อมูลเกี่ยวกับโปรแกรม Octave ซึ่งเป็นซอฟต์แวร์ตัวหนึ่งที่เหมาะสมสำหรับใช้ในการศึกษาและวิจัยในวิศวกรรมสาขาต่างๆ

โปรแกรม Octave

โปรแกรม Octave เป็นซอฟต์แวร์สำหรับการคำนวณเชิงเลข (numerical computation) ที่เผยแพร่แบบโอเพนซอร์สภายใต้ GNU General Public License นอกจากเรื่องค่าใช้จ่ายที่เป็นศูนย์แล้วจุดเด่นของ Octave คือการมีรูปแบบของภาษาและคำสั่งที่คล้ายคลึงกับโปรแกรม MATLAB ซึ่งเป็นซอฟต์แวร์ที่ใช้กันกันอย่างแพร่หลายในวิศวกรรมสาขาต่างๆ ทั้งนี้ตำราเรียนหลายๆเล่มได้มีการนำ MATLAB มาใช้ประกอบเนื้อหาภายในเล่มซึ่งโค้ดส่วนใหญ่สามารถนำมาใช้ใน Octave โดยไม่ต้องแก้ไขหรือแก้ไขคำสั่งให้ตรงกันเท่านั้น อย่างไรก็ตามโปรแกรม Octave ยังมีข้อจำกัดหลายอย่างเมื่อเทียบกับ MATLAB ได้แก่

- ชุดคำสั่งของ Octave จะคล้ายคำสั่งในโปรแกรม MATLAB รุ่นที่ 4 มากที่สุดโดยจะไม่มีชุดคำสั่งบางประเภทใน MATLAB รุ่น 5 หรือ 6
 - จำนวนของชุดคำสั่งเพื่อการคำนวณเฉพาะทางยังน้อยกว่าเมื่อเทียบกับ MATLAB
 - เมื่อเกิดปัญหาในการใช้งาน ผู้ใช้ต้องค้นคว้าหรือสอบถามจากผู้ใช้อื่นๆในอินเทอร์เน็ตเอง
- จากประสบการณ์ของผู้เขียนที่ใช้ Octave ในการสอน ข้อจำกัดข้างต้นนี้ไม่มีผลกระทบมากนัก ดังนั้นเมื่อเทียบกับข้อจำกัดของ MATLAB ที่แย่ที่สุดสำหรับคนไทยคือค่าลิขสิทธิ์ซึ่งเริ่มตั้งแต่ประมาณ 50,000 บาท/ผู้ใช้ ทุกคนคงเห็นได้ว่าโปรแกรม Octave เป็นทางเลือกหนึ่งที่น่าสนใจทีเดียว

การติดตั้งโปรแกรม Octave

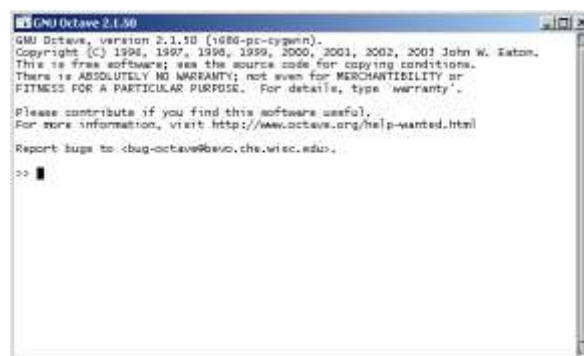
การติดตั้งโปรแกรม Octave เริ่มต้นด้วยการดาวน์โหลดไฟล์สำหรับระบบปฏิบัติการที่ต้องการ (ลินุกซ์ ไมโครซอฟท์วินโดวส์ หรือ Mac OS) จากโฮมเพจ <http://octave.sf.net> เนื่องจาก Octave มักจะเป็นหนึ่งในซอฟต์แวร์ที่มากับลินุกซ์อยู่แล้วผู้เขียนจึงขออธิบายขั้นตอนการติดตั้งสำหรับกรณีไมโครซอฟท์วินโดวส์เท่านั้น ไฟล์ที่ใช้ในการติดตั้งคือ octave-2.1.50a-inst.exe ซึ่งเป็นโปรแกรมที่จะทำการคัดลอกไฟล์ของโปรแกรม Octave รุ่น 2.1.50a ลงในโฟลเดอร์ที่กำหนดพร้อมกับตั้งค่าการทำงานต่างๆ วิธีการติดตั้งคือให้ดับเบิลคลิกไฟล์ที่ดาวน์โหลด

โหลดมาเพื่อสั่งให้เริ่มกระบวนการติดตั้ง ในระหว่างการติดตั้งโปรแกรมจะแสดงหน้าต่างเพื่อถามรายละเอียดบางอย่างได้แก่ไอคอนที่ต้องการให้ติดตั้งและ โฟลเดอร์สำหรับเก็บไฟล์ต่างๆดังในรูปที่ 2



รูปที่ 2 หน้าต่างของโปรแกรมติดตั้ง Octave

หลังจากโปรแกรมติดตั้งทำการคัดลอกไฟล์ลงในโฟลเดอร์ที่กำหนดแล้วจะมีการถามรายละเอียดของโปรแกรมอื่นที่จำเป็นสำหรับการทำงานได้แก่โปรแกรมแก้ไขไฟล์ข้อความ (editor) และ โปรแกรมแสดงไฟล์แบบ Postscript ซึ่งถูกกำหนดไว้เบื้องต้นเป็นโปรแกรม Notepad และ Ghostscript ตามลำดับ ในขั้นตอนนี้ผู้ใช้สามารถกดปุ่ม Enter เพื่อยอมรับข้อกำหนดที่มีอยู่แล้ว หลังจากติดตั้งเสร็จจะมีไอคอนของโปรแกรม Octave แสดงขึ้นบนหน้าจอ เมื่อดับเบิลคลิกที่ไอคอนนั้นหน้าต่างคำสั่ง (command window) ของ Octave ก็จะปรากฏขึ้นมาเพื่อรอรับคำสั่งจากผู้ใช้



รูปที่ 3 หน้าต่างคำสั่งของ Octave บนไมโครซอฟท์วินโดวส์

การใช้งานเบื้องต้น

รูปแบบการใช้งานโปรแกรม Octave คือการป้อนนิพจน์ (expression) ต่างๆได้แก่คำสั่ง ฟังก์ชัน ข้อมูล หรือเงื่อนไข ณ เครื่องหมายเตรียมพร้อม >> ในหน้าต่างคำสั่ง เมื่อผู้ใช้ป้อนนิพจน์ เสร็จและกดปุ่ม Enter โปรแกรมจะตอบสนองด้วยการประมวลผลนิพจน์นั้นแล้วแสดงผลลัพธ์ทันทีเมื่อประมวลผลเสร็จ ผลลัพธ์ที่ได้จะถูกนำไปเก็บในตัวแปรที่อยู่ทางซ้ายของเครื่องหมายเท่ากับ (=) หรือตัวแปรชื่อ ans ในกรณีไม่มีการกำหนดตัวแปรเพื่อรับคำตอบ นอกจากการทำงานแบบตอบโต้ทันที (interactive) แล้วสิ่งที่ Octave แตกต่างจากภาษาคอมพิวเตอร์ทั่วไปคือการกำหนดตัวแปร ไม่จำเป็นต้องประกาศว่าเป็นตัวแปรสำหรับเก็บข้อมูลประเภทใด รวมทั้งสามารถกำหนดให้ตัวแปรเดิมเก็บข้อมูลประเภทที่ต่างกันได้อีก ทั้งนี้ชื่อของตัวแปรจะต้องขึ้นต้นด้วย

ตัวอักษรเสมอและอาจตามด้วยตัวอักษรหรือตัวเลขก็ได้โดย Octave จะถือว่าตัวแปรที่ใช้ตัวอักษรใหญ่เล็กต่างกันว่าเป็นคนละตัวแปรกัน ชื่อเฉพาะจำนวนหนึ่งจะถูกกำหนดไว้สำหรับค่าคงที่ได้แก่ pi (π) Inf (∞) และ NaN (Not a number) รูปที่ 4 แสดงตัวอย่างของการป้อนนิพจน์ภายในหน้าต่างคำสั่งให้กับ Octave

```

GNU Octave, version 2.1.50 (i686-pc-cygwin).
Copyright (C) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003 John W. Eaton.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html
Report bugs to <bug-octave@bevo.cha.wisc.edu>.

>> a = pi*10^2
a = 314.16
>> b = 3 + 2i
b = 3 + 2i
>> a*b
ans = 314.16 + 628.32i
>> str = 'This is a test'
str = This is a test
>> x = sqrt(2);
>> x
x = 1.4142
>>

```

รูปที่ 4 การป้อนคำสั่งและการตอบสนองในหน้าต่างคำสั่ง

จากรูปที่ 4 การป้อนค่าเชิงซ้อนใช้การพิมพ์ i หรือ j ก่อนหรือหลังตัวเลขเพื่อระบุว่าเป็นค่าจินตภาพ ส่วนการป้อนข้อมูลประเภทตัวอักษรหรือข้อความจะใช้การล้อมด้วยเครื่องหมาย ‘ ‘ การพิมพ์เครื่องหมายอัฒภาค (;) ต่อท้ายนิพจน์จะทำให้ Octave ไม่แสดงผลบนหน้าจอหลังจากประมวลผลเสร็จ

การคำนวณใน Octave จะอิงอยู่บนการประมวลผลข้อมูลในรูปของเมตริกซ์ซึ่งอาจมีสมาชิกได้ทั้งค่าจำนวนจริงและจำนวนเชิงซ้อน การป้อนค่าของเมตริกซ์กระทำด้วยการพิมพ์ค่าสมาชิกภายในขอบเขตของวงเล็บก้ามปู ([]) โดยป้อนค่าสมาชิกทีละแถวแยกด้วยวรรคหรือจุลภาค (,) และใช้เครื่องหมายอัฒภาค (;) ในการแบ่งแยกระหว่างแถว เช่น การป้อนเวกเตอร์แบบแถวที่มีสมาชิก 3 ตัวและเมตริกซ์ขนาด 3x3 ให้พิมพ์ดังนี้ (เครื่องหมาย ↵ หมายถึงการกดปุ่ม Enter)

```
>> V = [ 1 ; 2 ; 3 ] ↵
```

```
>> M = [ 1, 2, 3 ; 4, 5, 6 ; 7, 8, 9 ] ↵
```

การอ้างถึงสมาชิกใดๆภายในตัวแปรซึ่งเก็บข้อมูลประเภทเมตริกซ์ใช้การใส่ค่าตำแหน่งแถวและหลักภายในวงเล็บที่ตามหลังชื่อตัวแปร เช่น M(1,1) ส่วนข้อมูลประเภทเวกเตอร์ให้ใส่เพียงตำแหน่งสมาชิก เช่น V(2)

ตัวกระทำทางคณิตศาสตร์ (mathematical operator) ที่สามารถใช้ได้ใน Octave ได้แก่ บวก (+) ลบ (-) คูณ (*) ยกกำลัง (^) และการกลับแบบคอนจูเกต (') ส่วนการหารแบ่งเป็นการหารทางซ้าย ($x = A \setminus b$ โดย x เป็นคำตอบของสมการ $A \cdot x = b$) และการหารทางขวา ($x = A / b$ โดย x เป็นคำตอบของสมการ $x \cdot A = b$) ตัวอย่างเช่นการคำนวณหาคำตอบของ $R = V^T M$ ซึ่ง R เป็นเวกเตอร์แบบแถวให้พิมพ์ดังนี้

```
>> R = V' * M ↵
```

นอกจากใช้ตัวกระทำทางคณิตศาสตร์แล้ว เราสามารถเรียกใช้ฟังก์ชันเพื่อทำการประมวลผลข้อมูลโดยพิมพ์ชื่อฟังก์ชันพร้อมทั้งป้อนข้อมูลที่จำเป็นภายในวงเล็บล้อม ตัวอย่างเช่นการหาส่วนผกผันของเมตริกซ์ M ให้ป้อนดังนี้

```
>> N = inv(M) ↵
```

หากค่ามิติของเวกเตอร์และเมตริกซ์ที่ใช้ในการคำนวณไม่เหมาะสม Octave จะทำการแจ้งข้อผิดพลาดออกมาทั้งในกรณีที่ใช้ตัวกระทำทางคณิตศาสตร์หรือฟังก์ชัน

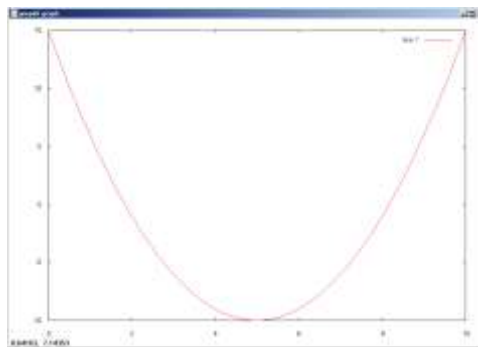
การสร้างรูปภาพใน Octave ใช้การเรียกฟังก์ชัน plot ซึ่งจะทำให้แสดงรูปภาพแบบ 2 มิติที่มีสเกลแบบเชิงเส้น (linear scale) จากข้อมูลแบบเวกเตอร์ที่ระบุค่าในแกนนอนและแกนตั้ง เช่น การสร้างรูปภาพของสมการ $x^2 - 10x + 15$ ให้ป้อนดังนี้

```
>> x = 0:0.1:10; ↵
```

```
>> y = x.^2 - 10*x + 15; ↵
```

```
>> plot(x,y); ↵
```

ซึ่งเป็นการระบุให้สร้างเวกเตอร์ x ที่มีค่าตั้งแต่ 0 จนถึง 10 เพิ่มค่าทีละ 0.1 โดยใช้เครื่องหมาย Colon (:) ในการสร้างเวกเตอร์ของตัวเลขที่มีค่าเรียงกัน จากนั้นเป็นการคำนวณค่าของเวกเตอร์ y จากสมการ $y = x^2 - 10x + 15$ ซึ่งการพิมพ์เครื่องหมายจุด (.) ก่อนตัวกระทำ ^ หมายความว่าให้ทำการคำนวณในระดับสมาชิก (เวกเตอร์แบบแถวคูณกันเองโดยตรงไม่ได้) สุดท้ายจึงเป็นการวาดรูปภาพโดยใช้ x เป็นค่าแกนนอนและ y เป็นค่าแกนตั้ง ดังแสดงในรูปที่ 5



รูปที่ 5 รูปภาพของสมการ $y = x^2 - 10x + 15$

หลังจากแสดงรูปภาพผู้ใช้สามารถจัดการรูปภาพที่สร้างขึ้นด้วยการใช้คำสั่งเช่น title xlabel และ ylabel เพื่อตั้งชื่อหัวเรื่องของกราฟ ข้อความกำกับแกน X และแกน Y ตามลำดับ ผู้ใช้สามารถแสดงผลเส้นกราฟมากกว่า 1 เส้นพร้อมกันได้ด้วยการกำหนดเวกเตอร์จำนวนมากกว่า 1 คู่เป็นข้อมูลให้กับฟังก์ชัน plot เช่นการพิมพ์ plot(x1,y1,x2,y2) จะแสดงผลเส้นกราฟ 2 รูปลงบนหน้าต่างเดียวกันโดยรูปภาพแรกจะอิงบนข้อมูลของคู่เวกเตอร์ x1 และ y1 ส่วนรูปภาพที่สองอิงบนข้อมูลของคู่เวกเตอร์ x2 และ y2

การวิเคราะห์ระบบควบคุม

ในหัวข้อนี้จะสาธิตการประยุกต์ใช้โปรแกรม Octave เพื่อวิเคราะห์ปัญหาในสาขาวิศวกรรมระบบควบคุม ตัวอย่างที่ใช้ในครั้งนี้เป็นระบบควบคุมตำแหน่งของมอเตอร์กระแสตรงที่ต่ออยู่กับภาระโหลดจากหนังสือ Automatic Control Systems ทั้งนี้แบบจำลองทางคณิตศาสตร์ (mathematical model) ของมอเตอร์เมื่อผ่านการแปลงลาปลาซ (Laplace transform) และแทนค่าพารามิเตอร์ต่างๆแล้วจะเขียนได้ในรูปของฟังก์ชันโอนย้าย (transfer function) ระหว่างแรงดันเข้าของขดลวดอาเมเจอร์ $E_a(s)$ และตำแหน่งเชิงมุมของแกนหมุนมอเตอร์ $\theta(s)$ ดังนี้

$$\frac{\theta(s)}{E_a(s)} = \frac{4500}{s(s + 361.2)}$$

สิ่งที่เราจะศึกษาคือผลกระทบของค่าพารามิเตอร์ของตัวควบคุมแบบ PI (PI controller) ที่มีต่อการเปลี่ยนแปลงของ $\Theta(s)$ ฟังก์ชันโอนถ่ายของตัวควบคุมแบบ PI เขียนได้ดังนี้

$$K(s) = K_p + \frac{K_I}{s} = \frac{K_p s + K_I}{s}$$

ขั้นตอนแรกในการใช้ Octave เพื่อศึกษาระบบควบคุมใด ๆ ก็คือการสร้างแบบจำลองด้วยการใช้ฟังก์ชัน `tf2sys` เพื่อสร้างองค์ประกอบต่างๆภายในระบบ ข้อมูลที่ผู้ใช้ต้องป้อนให้ฟังก์ชัน `tf2sys` คือสัมประสิทธิ์ของโพลีโนเมียลของเทอมเศษและเทอมส่วนของฟังก์ชันโอนถ่าย การป้อนฟังก์ชันโอนถ่ายของมอเตอร์และตัวควบคุม ซึ่งมี K_p เท่ากับ 1 และ K_I เท่ากับ 0.5 ให้พิมพ์ดังนี้

```
>> Kp = 1; ↵
```

```
>> Ki = 0.5; ↵
```

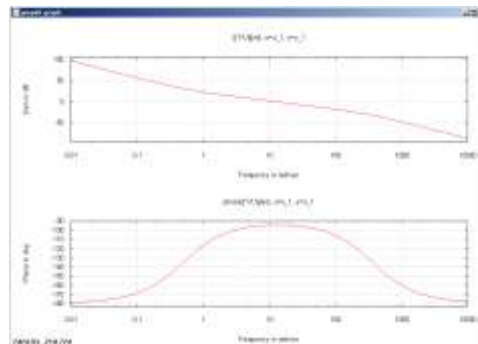
```
>> motor = tf2sys([4500], [1, 361.2, 0]); ↵
```

```
>> PI = tf2sys([Kp, Ki], [1, 0]); ↵
```

เมื่อป้อนข้อมูลเรียบร้อยแล้วจะเข้าสู่ขั้นตอนของการวิเคราะห์ซึ่งในที่นี้จะเป็นการศึกษาผลกระทบของตัวควบคุมที่มีต่อผลตอบสนองเชิงความถี่โดยใช้ฟังก์ชัน `sysmult` เพื่อสร้างแบบจำลองที่เกิดจากการต่อเชื่อมตัวควบคุมเข้ากับมอเตอร์ จากนั้นจึงใช้ฟังก์ชัน `bode` เพื่อคำนวณหาผลตอบสนองเชิงความถี่และสร้างกราฟออกมาในรูปของ Bode plot (กราฟของอัตราขยาย-ความถี่/เฟส-ความถี่)

```
>> plant = sysmult(motor,PI); ↵
```

```
>> bode(plant); ↵
```



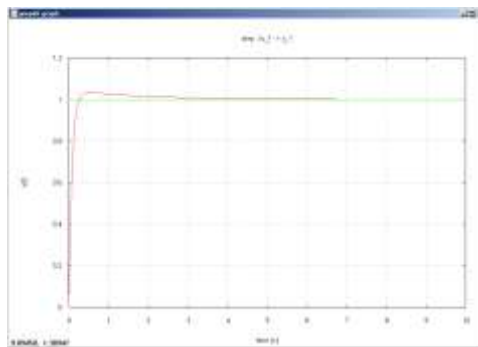
รูปที่ 6 ผลตอบสนองเชิงความถี่เมื่อต่อตัวควบคุมแบบ PI เข้ากับมอเตอร์

จากกราฟของผลตอบสนองเชิงความถี่ในรูปที่ 6 เราพบว่าค่า Phase margin (180° - ค่าเฟส ณ ตำแหน่งอัตราขยายเท่ากับ 0 dB) มีค่าประมาณ 90° ซึ่งเป็นการยืนยันว่าหากเราใช้ตัวควบคุมที่มีค่า $K_p = 1$ และ $K_I = 0.5$ แล้วระบบควบคุมนั้นจะมีเสถียรภาพ

นอกจากการวิเคราะห์ในเชิงความถี่แล้ว เราสามารถพิจารณาการตอบสนองในเชิงเวลาของระบบควบคุมมอเตอร์ที่มีต่อการเปลี่ยนแปลงแรงดันอ้างอิงอย่างฉับพลันด้วยการใช้ฟังก์ชัน `buildssic` เพื่อเปลี่ยนโครงสร้างเป็นระบบวงปิด (closed-loop system) โดยใช้การป้อนกลับแบบลบ (negative feedback) จากนั้นจึงใช้ฟังก์ชัน `step` เพื่อทำการคำนวณและสร้างรูปกราฟของตำแหน่งแกนหมุนที่เวลาต่างๆโดยถือว่าสัญญาณเข้าเป็นสัญญาณขั้น (step input) ดังแสดงในรูปที่ 7

```
>> clplant = buildssic([1, -1],[ ],[ ],[ ],plant); ↵
```

```
>> step(cplant); ↵
```



รูปที่ 7 กราฟแสดงตำแหน่งแกนหมุนเมื่อมีสัญญาณเข้าแบบขั้น

บทสรุป

บทความนี้ได้แนะนำโปรแกรม Octave ซึ่งเป็นซอฟต์แวร์ที่เผยแพร่ในแบบโอเพนซอร์สสำหรับใช้ทดแทนโปรแกรม MATLAB ถ้าผู้ใดสนใจที่จะศึกษาการใช้งานในรายละเอียดมากขึ้นสามารถศึกษาได้จากคู่มือการใช้งานซึ่งถูกจัดตั้งมาในรูปแบบ html พร้อมกับโปรแกรม Octave อยู่แล้ว นอกจากนี้หนังสือที่สอนการใช้งานโปรแกรม MATLAB ทั่วไปก็สามารถใช้ได้เช่นเดียวกัน ในกรณีที่ประสบปัญหาในการใช้งานที่ไม่มีอธิบายในคู่มือก็อาจส่งอีเมลเพื่อถามกลุ่มผู้ใช้โดยผ่านระบบเมลลิงลิสต์ (mailing list) ของโฮมเพจของ Octave ได้

ข้อมูลอ้างอิง

- B.C. Kuo and F. Golnaraghi, “Automatic Control Systems”, John Wiley & Sons, 2003
- <http://www.octave.org>, โฮมเพจอย่างเป็นทางการของโปรแกรม Octave
- <http://octave.sf.net> โฮมเพจของกลุ่มนักพัฒนาซึ่งช่วยกันสร้างไฟล์ของโปรแกรม Octave สำหรับใช้งานบนเครื่องคอมพิวเตอร์รุ่นต่างๆ

ภาคผนวก ก

ข้อสัญญา GNU Public License ฉบับภาษาไทย

ข.1 อารัมภบท

สัญญาอนุญาตให้ใช้ซอฟต์แวร์ส่วนใหญ่ได้รับการออกแบบมาเพื่อริชชภาพของคุณในการแบ่งปันและแก้ไขซอฟต์แวร์. ในทางกลับกัน GNU General Public License มีจุดมุ่งหมายเพื่อประกันเสรีภาพของคุณในการแบ่งปันและแก้ไขซอฟต์แวร์เสรี (free software) เพื่อให้แน่ใจว่าซอฟต์แวร์จะเป็นสิ่งที่เสรีสำหรับผู้ใช้ทุกคน. General Public License คือสัญญาอนุญาตให้สาธารณชนใช้สิทธิตามลิขสิทธิ์ที่ได้รับการนำมาใช้กับซอฟต์แวร์ส่วนใหญ่ของ Free Software Foundation และโปรแกรมใดก็ตามที่ผู้สร้างสรรค์ยึดมั่นต่อการใช้สัญญานี้. (มีซอฟต์แวร์บางชิ้นของ Free Software Foundation ที่ครอบคลุมโดย GNU Library General Public License แทน.) คุณเองก็สามารถใช้สัญญานี้กับโปรแกรมของคุณได้เช่นเดียวกัน.

เมื่อเรากล่าวถึง free software เราหมายถึงเสรีภาพ ไม่ใช่ราคา. General Public License ของเราได้รับการออกแบบมาเพื่อให้แน่ใจว่า ๑ คุณจะมีเสรีภาพที่จะเผยแพร่สำเนาของซอฟต์แวร์เสรี (และคิดราคาสำหรับการจัดจำหน่ายในกรณีที่คุณต้องการ) ๑ คุณจะได้รับ source code หรือสามารถที่จะได้มาในกรณีที่คุณต้องการ ๑ คุณจะสามารแก้ไขหรือใช้ส่วนประกอบของซอฟต์แวร์นั้นในซอฟต์แวร์เสรีโปรแกรมใหม่ และ ๑ คุณจะได้รับรู้ว่าคุณมีสิทธิที่จะทำทั้งหมดนี้ได้.

เพื่อปกป้องสิทธิของคุณ เราจำเป็นต้องวางข้อจำกัดต่าง ๆ เพื่อห้ามมิให้ใครปฏิเสธสิทธิเหล่านี้ต่อคุณหรือเรียกร้องให้คุณละสิทธิดังกล่าว. ข้อจำกัดเหล่านี้นำไปสู่ภาระหน้าที่จำนวนหนึ่งของคุณ ในกรณีที่คุณเผยแพร่สำเนาของซอฟต์แวร์ หรือดัดแปลงซอฟต์แวร์นั้น.

ตัวอย่างเช่น ในกรณีที่เผยแพร่สำเนาของโปรแกรมลักษณะนี้ ไม่ว่าจะให้เปล่าหรือคิดราคา คุณจะต้องให้สิทธิทั้งหมดอย่างที่คุณมีแก่ผู้รับด้วย. คุณจะต้องจัดการให้บุคคลนั้นได้รับหรือสามารถที่จะได้มาซึ่ง source code นั้นเช่นเดียวกับคุณ. และคุณจะต้องแสดงข้อสัญญาเหล่านี้แก่บุคคลนั้น เพื่อที่บุคคลนั้นจะได้รับรู้ถึงสิทธิของตน.

เราใช้วิธีปกป้องสิทธิของคุณด้วยสองขั้นตอนคือ: (1) สงวนลิขสิทธิ์ของซอฟต์แวร์นั้น และ (2) เสนอสัญญาอนุญาตให้คุณสามารถที่จะทำซ้ำ เผยแพร่ และ/หรือดัดแปลงซอฟต์แวร์นั้น ได้อย่างถูกต้องตามกฎหมาย.

นอกจากนั้น เพื่อเป็นการปกป้องผู้สร้างสรรค์แต่ละรายและตัวของเราเอง เราต้องแน่ใจว่าทุก ๆ คนจะได้รับการทำความเข้าใจว่าจะไม่มีการรับประกันใด ๆ ในซอฟต์แวร์เสรีชิ้นนี้. ในกรณีที่ซอฟต์แวร์ได้รับการดัดแปลงโดยบุคคลอื่นและแจกจ่ายส่งต่อกันไป เราต้องการให้ผู้รับได้ตระหนักว่าสิ่งที่ตนได้รับนั้นไม่ได้เป็นของดั้งเดิม เพื่อว่าปัญหาใด ๆ อันเกิดจากบุคคลอื่นจะได้ไม่พาดพิงไปถึงชื่อเสียงของผู้สร้างสรรค์ต้นฉบับ.

ท้ายที่สุด ซอฟต์แวร์เสรีมักจะถูกลักลอบโดยสิทธิบัตรซอฟต์แวร์อยู่เป็นนิจ. เราประสงค์ที่จะหลีกเลี่ยงความเสี่ยงที่ผู้จำหน่ายซอฟต์แวร์เสรีโปรแกรมใดโปรแกรมหนึ่งจะไปขอรับสัญญาอนุญาตให้ใช้สิทธิตามสิทธิบัตรเอาไว้อีก อันจะเป็นผลทำให้ได้โปรแกรมนั้นเป็นเอกสิทธิ์เฉพาะ (proprietary). เพื่อป้องกันเรื่องนี้ เรา

ได้กำหนดไว้อย่างชัดเจนว่าสิทธิบัตรใดก็ตามจะต้องอนุญาตเพื่อให้ทุกคนใช้ได้ฟรีหรือมีเงินนั้นก็จะต้องอนุญาตไม่ได้เลย.

การทำซ้ำ เผยแพร่ และดัดแปลงมีข้อกำหนดและเงื่อนไขดังต่อไปนี้.

ข.2 ข้อกำหนดและเงื่อนไขในการทำซ้ำ เผยแพร่ และดัดแปลง

0. สัญญานี้ใช้กับโปรแกรมหรืองานอื่นใดที่มีข้อความบอกกล่าวโดยเจ้าของลิขสิทธิ์ว่าสามารถนำไปเผยแพร่โดยอาศัยข้อความตาม General Public License. ถัดจากนี้ไปคำว่า "โปรแกรม" หมายถึงโปรแกรมหรืองานในลักษณะดังกล่าว และ "งานที่มีพื้นฐานจากโปรแกรม" หมายถึงโปรแกรมนั้นหรืองานดัดแปลงจากโปรแกรมนั้นตามกฎหมายลิขสิทธิ์: กล่าวคือ งานที่ประกอบด้วยโปรแกรมนั้นไม่ว่าทั้งหมดหรือบางส่วนไม่ว่าจะคงเดิมหรือมีการดัดแปลงและ/หรือได้รับการแปลเป็นภาษาอื่น. (จากนี้เป็นต้นไป การแปลนับรวมอยู่ในความหมายของคำว่า "การดัดแปลง" โดยไม่มีขอบเขต.) คำว่า "คุณ" หมายถึงแต่ละบุคคลผู้ได้รับอนุญาตให้ใช้สิทธิตามสัญญา.

สัญญานี้ไม่ได้ครอบคลุมไปถึงการกระทำอื่นใดนอกเหนือจากการทำซ้ำ เผยแพร่ และดัดแปลง การกระทำอื่นใดนั้นอยู่นอกเหนือขอบเขตของสัญญา. สัญญาไม่ได้ควบคุมการรันใช้งานโปรแกรม และสัญญาจะครอบคลุมถึงผลลัพธ์จากโปรแกรมก็ต่อเมื่อเนื้อหาของผลลัพธ์เป็นองค์ประกอบของงานที่มีพื้นฐานจากโปรแกรม (โดยไม่ขึ้นอยู่กับการที่ผลลัพธ์นั้นได้ผ่านการทำขึ้นมาจากการรันโปรแกรม). ผลลัพธ์จากโปรแกรมจะถือเป็นงานที่มีพื้นฐานจากโปรแกรมหรือไม่ขึ้นอยู่กับว่าโปรแกรมนั้นทำอะไร.

1. คุณสามารถทำซ้ำหรือเผยแพร่ในสื่อบันทึกใด ๆ ซึ่งสำเนาที่ปราศจากการแก้ไขของ source code ของโปรแกรมตามที่คุณได้รับมา โดยมีข้อแม้ว่า ๑ คุณจะต้องแสดงคำบอกกล่าวลิขสิทธิ์และคำปฏิเสธการรับประกันไว้อย่างเด่นชัดและเหมาะสมในแต่ละสำเนา ๑ รักษาคำบอกกล่าวทั้งหมดที่อ้างถึงสัญญานี้และความปราศจากการรับประกันให้คงอยู่ด้วยกันอย่างสมบูรณ์ และ ๑ มอบสำเนาของสัญญานี้ไปพร้อมกับโปรแกรม.

คุณสามารถคิดราคาสำหรับการกระทำในทางกายภาพเพื่อถ่ายทอดสำเนาชิ้นหนึ่ง ๆ ได้ และคุณสามารถที่จะเสนอขายการรับประกันได้ในกรณีที่คุณต้องการ.

2. คุณสามารถดัดแปลงสำเนาของโปรแกรมไม่ว่าทั้งหมดหรือบางส่วน การกระทำเช่นนั้นคือการสร้างงานที่มีพื้นฐานจากโปรแกรม และคุณสามารถทำซ้ำและเผยแพร่การดัดแปลงหรืองานในลักษณะดังกล่าวโดยอาศัยข้อความตามข้อ 1 ข้างต้น โดยมีข้อแม้ว่าคุณจะต้องปฏิบัติตามเงื่อนไขเหล่านี้ทั้งหมด:

- a. ในไฟล์ที่คุณแก้ไข คุณจะต้องใส่คำบอกกล่าวที่เด่นชัดว่าคุณได้แก้ไขไฟล์นั้น รวมทั้งวันที่ในการแก้ไขแต่ละครั้ง.
- b. ในการเผยแพร่หรือโฆษณา (นำเสนอต่อสาธารณชน) งานใดก็ตามที่ทั้งหมดหรือบางส่วนของงานนั้นประกอบด้วยหรือดัดแปลงมาจากโปรแกรมหรือบางส่วนของโปรแกรม คุณจะต้องการอนุญาตต่องานโดยรวมทั้งหมดแก่บุคคลทั้งหลายโดยไม่คิดมูลค่า โดยอาศัยข้อความตามสัญญานี้.
- c. ในกรณีที่โปรแกรมที่ดัดแปลงแล้วมีการทำงานตามปกติด้วยการรับคำสั่งเชิงโต้ตอบในขณะที่รัน คุณต้องทำให้โปรแกรมนั้น เมื่อนำมาเริ่มต้นรันใช้งานในเชิงโต้ตอบดังกล่าวตามปกติ แล้วโปรแกรมนั้นจะพิมพ์หรือแสดงให้เห็นประกาศอันประกอบด้วย ๑ คำบอกกล่าว

ลิขสิทธิ์อย่างเหมาะสมและ ๑) คำบอกกล่าวว่าไม่มีการรับประกัน (หรือมีจะนั้นก็บอกว่าคุณเองเป็นผู้ให้การรับประกัน) และ ๑) บอกว่าผู้ใช้สามารถเผยแพร่โปรแกรมนั้นต่อไปภายใต้เงื่อนไขตามสัญญา และ ๑) บอกว่าวิธีที่ผู้ใช้จะสามารถอ่านสำเนาของสัญญา. (ข้อยกเว้น: ในกรณีที่แม้ตัวโปรแกรมเองจะทำงานในเชิงโต้ตอบ แต่ตามปกติแล้วจะไม่มีมีการแสดงประกาศในลักษณะดังกล่าวอยู่ก่อนแล้ว เช่นงานที่มีพื้นฐานจากโปรแกรมไม่จำเป็นต้องแสดงประกาศใด ๆ.)

เกณฑ์เหล่านี้ใช้บังคับต้องงานดัดแปลงโดยรวมทั้งหมด. ในกรณีของส่วนที่จำแนกออกมาได้ส่วนใดอันมิได้ดัดแปลงมาจากโปรแกรม และสามารถถือได้ว่าเป็นงานอิสระต่างหากออกไปด้วยเหตุอันสมควร เช่นนั้นสัญญาและเงื่อนไขของสัญญานี้จะไม่มีผลบังคับใช้ต่อส่วนดังกล่าว เมื่อคุณเผยแพร่ส่วนนั้นเป็นงานต่างหากออกไป. แต่เมื่อคุณเผยแพร่ส่วนเดียวกันนั้นเป็นส่วนประกอบของงานรวมอันเป็นงานที่มีพื้นฐานจากโปรแกรม การเผยแพร่งานรวมดังกล่าวจะต้องกระทำโดยอาศัยข้อความตามสัญญานี้ โดยที่สิทธิสำหรับบุคคลทั้งหลายจะขยายขอบเขตครอบคลุมไปตลอดงานโดยรวมทั้งหมด ตลอดไปจนถึงแต่ละส่วนประกอบทุก ๆ ส่วนไม่ว่าใครจะเป็นผู้เขียนส่วนนั้น. ด้วยเหตุข้างต้น ข้อนี้มิได้ประสงค์จะเรียกร้องหรือโต้แย้งสิทธิของคุณในงานที่คุณสร้างสรรค์ขึ้นมาเองทั้งหมด หากแต่ประสงค์ที่จะใช้ลิขสิทธิ์เพื่อควบคุมการเผยแพร่งานดัดแปลงหรืองานรวบรวมอันเป็นงานที่มีพื้นฐานจากโปรแกรม.

นอกจากนี้ การเพียงแต่ทำงานอื่นที่ไม่ได้เป็นงานที่มีพื้นฐานจากโปรแกรมมาอยู่ร่วมกับโปรแกรม (หรืองานที่มีพื้นฐานจากโปรแกรม) ในหน่วยความจำสำรองหรือสื่อบันทึกเพื่อการเผยแพร่เดียวกันมิได้เป็นเหตุให้งานดังกล่าวตกอยู่ภายใต้ขอบเขตของสัญญานี้.

3. คุณสามารถทำซ้ำและเผยแพร่โปรแกรม (หรืองานที่มีพื้นฐานจากโปรแกรมตามข้อ 2) ในรูปแบบ object code หรือ executable โดยอาศัยข้อความตามข้อ 1 และ 2 ข้างต้น โดยมีข้อแม้ว่าคุณจะต้องดำเนินการอย่างใดอย่างหนึ่งต่อไปนี้ด้วย:

- มอบ source code อย่างครบถ้วนของโปรแกรมหากกล่าวในรูปแบบที่คอมพิวเตอร์อ่านได้ไปพร้อมกับโปรแกรมนั้น โดยที่ source code ดังกล่าวจะต้องเผยแพร่โดยอาศัยข้อความตามข้อ 1 และ 2 ข้างต้น ด้วยสื่อบันทึกที่ใช้กันเป็นกิจวัตรในการแลกเปลี่ยนซอฟต์แวร์ หรือ
- แนบข้อเสนอเป็นลายลักษณ์อักษรอันมีผลอย่างน้อยสามปีว่าจะมอบสำเนาอย่างครบถ้วนของ source code ของโปรแกรมหากกล่าวในรูปแบบที่คอมพิวเตอร์อ่านได้แก่บุคคลทั้งหลายโดยคิดราคาไม่มากไปกว่าต้นทุนในทางกายภาพของคุณในการดำเนินการเผยแพร่ source code โดยที่ source code ดังกล่าวจะต้องเผยแพร่โดยอาศัยข้อความตามข้อ 1 และ 2 ข้างต้น ด้วยสื่อบันทึกที่ใช้กันเป็นกิจวัตรในการแลกเปลี่ยนซอฟต์แวร์ หรือ
- แนบข้อมูลที่คุณได้รับถึงข้อเสนอในการเผยแพร่ source code ของโปรแกรมหากกล่าว. (ทางเลือกนี้ใช้ได้เฉพาะกรณีการเผยแพร่ที่ไม่เป็นการค้า อีกทั้งคุณก็ได้รับโปรแกรมในรูปแบบ object code หรือ executable ด้วยข้อเสนอเช่นว่าอย่างถูกต้องตามข้อย่อย b ข้างต้น.)

คำว่า source code ของงานหนึ่งงานใดหมายถึงงานนั้นในรูปแบบที่เหมาะสมที่สุดสำหรับการทำการดัดแปลง. สำหรับงานในรูปแบบ executable คำว่า source code อย่างครบถ้วนหมายถึง source code ทั้งหมดของทุก module ในงานนั้น รวมถึง interface definition file ใด ๆ ที่เกี่ยวข้อง รวมถึง script ที่ใช้ในการ

ควบคุมการคอมไพล์และการติดตั้ง executable นั้น. อย่างไรก็ตาม ในฐานะข้อยกเว้นพิเศษ source code ที่เผยแพร่ไม่จำเป็นต้องรวมไปถึงโค้ดก็ตาม (ไม่ว่าจะอยู่ในรูปแบบ source code หรือ binary) ที่ปรกติจะได้รับมาพร้อมกับส่วนประกอบหลัก (compiler, kernel และอื่น ๆ ในทำนองเดียวกัน) ของระบบปฏิบัติการที่ใช้รัน executable นั้น เว้นแต่ว่าส่วนประกอบนั้นจะประกอบไปเป็นส่วนหนึ่งใน executable ดังกล่าวเอง

ในกรณีที่มีการเผยแพร่ executable หรือ object code กระทำโดยการเสนอให้สามารถเข้าถึงสำเนาของโปรแกรมจากตำแหน่งที่ระบุ การเสนอที่เสมอภาคกันให้สามารถเข้าถึงสำเนาของ source code จากที่เดียวกันถือได้ว่าเป็นการเผยแพร่ source code แล้ว ถึงแม้ว่าบุคคลอื่นจะไม่มีเหตุจำเป็นอันใดที่จะต้องรับ source code ไปด้วยพร้อมกับ object code.

4. คุณไม่สามารถทำซ้ำ ดัดแปลง ให้ช่วงสัญญา หรือเผยแพร่โปรแกรม เว้นแต่จะปฏิบัติตามที่ได้กำหนดไว้โดยชัดแจ้งในสัญญานี้. ความพยายามอื่นใดที่จะทำซ้ำ ดัดแปลง ให้ช่วงสัญญา หรือเผยแพร่โปรแกรมถือเป็นโมฆะ และจะเป็นการยุติสิทธิของคุณตามสัญญานี้โดยอัตโนมัติ. อย่างไรก็ตาม บุคคลที่ได้รับสำเนาหรือสิทธิจากคุณตามสัญญานี้ จะไม่ถูกยกเลิกสัญญาตราใดที่บุคคลนั้นยังคงปฏิบัติตามสัญญาอย่างเคร่งครัด.
5. คุณไม่จำเป็นต้องยอมรับสัญญานี้ เนื่องจากคุณไม่ได้ลงนาม. แต่อย่างไรก็ตาม ไม่มีอะไรนอกจากนี้ที่อนุญาตให้คุณดัดแปลงหรือเผยแพร่โปรแกรมหรืองานดัดแปลงจากโปรแกรม. การกระทำดังกล่าวต้องห้ามตามกฎหมายถ้าหากคุณไม่ยอมรับสัญญานี้. ดังนั้น โดยการดัดแปลงหรือเผยแพร่โปรแกรม (หรืองานที่มีพื้นฐานจากโปรแกรม) คุณได้แสดงให้เห็นแล้วถึงการยอมรับสัญญานี้ รวมทั้งข้อกำหนดและเงื่อนไขทั้งหมดในการทำซ้ำ เผยแพร่ หรือดัดแปลงโปรแกรมหรืองานที่มีพื้นฐานจากโปรแกรม เพื่อที่จะกระทำเช่นนั้น.
6. ในแต่ละครั้งที่คุณเผยแพร่โปรแกรม (หรืองานที่มีพื้นฐานจากโปรแกรม) ต่อไป ผู้รับจะได้รับอนุญาตโดยอัตโนมัติจากเจ้าของสิทธิให้สามารถที่จะทำซ้ำ เผยแพร่ หรือดัดแปลงโปรแกรม ภายใต้ข้อกำหนดและเงื่อนไขของสัญญานี้. คุณไม่สามารถวางข้อจำกัดเพิ่มเติมต่อการใช้สิทธิของผู้รับที่ได้มอบไปตามนี้. คุณไม่ต้องการผิดชอบภาระในการบังคับการปฏิบัติตามสัญญาของบุคคลอื่น
7. ในกรณีที่เงื่อนไขอันเนื่องมาจากการพิพาทของศาลหรือข้อกล่าวหาในการละเมิดสิทธิบัตรหรือจากเหตุผลอื่นใด (โดยไม่จำกัดเฉพาะประเด็นเกี่ยวกับสิทธิบัตร) มาบังคับต่อคุณ (ไม่ว่าจะโดยคำสั่งของศาล ข้อตกลงหรือข้อตกลงในทางตรงกันข้าม) ในลักษณะที่ขัดต่อเงื่อนไขของสัญญานี้ เงื่อนไขเช่นนั้นไม่สามารถยกเว้นคุณออกจากเงื่อนไขของสัญญานี้. ถ้าหากคุณไม่สามารถดำเนินการเผยแพร่เพื่อที่จะปฏิบัติตามภาระของคุณในสัญญานี้และภาระผูกพันที่เกี่ยวข้องอื่นใดพร้อมไปด้วยกัน ผลก็คือคุณจะเผยแพร่โปรแกรมไม่ได้โดยเด็ดขาด. ตัวอย่างเช่น หากมีสัญญาอนุญาตให้ใช้สิทธิตามสิทธิบัตรฉบับใดฉบับหนึ่งไม่เปิดโอกาสให้บุคคลทั้งหมดที่ได้รับสำเนาของโปรแกรมจากคุณ ไม่ว่าจะโดยตรงหรือโดยอ้อม แจกจ่ายโปรแกรมต่อไปได้โดยไม่คิดค่าธรรมเนียม เช่นนั้นแล้วหนทางเดียวที่คุณจะปฏิบัติตามเงื่อนไขนั้นและสัญญานี้ไปได้พร้อมกันคือคุณจะต้องควั่นจากการเผยแพร่โปรแกรมอย่างสิ้นเชิง.

หากมีส่วนใดของข้อนี้กลายเป็นโมฆะหรือไม่มีผลบังคับภายใต้พฤติการณ์เฉพาะกรณีใด สัญญาประสงค์ให้ใช้ส่วนที่เหลืออยู่ของข้อนี้บังคับในกรณีเช่นนั้น และให้ใช้ข้อนี้โดยรวมทั้งหมดบังคับในพฤติการณ์อื่น ๆ.

จุดประสงค์ของข้อนี้มีได้เป็นการชักชวนให้คุณละเมิดการอ้างสิทธิในสิทธิบัตรหรือทรัพย์สินทางปัญญาใด ๆ หรือโต้แย้งความถูกต้องตามกฎหมายในการอ้างสิทธิเช่นนั้น จุดประสงค์เพียงประการเดียวของข้อนี้คือการ

ปกป้องบูรณาภาพของระบบเผยแพร่ซอฟต์แวร์เสรีซึ่งดำเนินโดยการปฏิบัติตาม public license. ผู้คนมากมายได้เอื้อเฟื้อผลงานสนับสนุนเข้าสู่สารพันของซอฟต์แวร์ที่เผยแพร่ผ่านระบบดังกล่าว โดยอาศัยความไว้วางใจว่าระบบจะทำงานอย่างเป็นธรรม ทั้งหมดนี้ขึ้นอยู่กับผู้เขียน/ผู้สนับสนุนที่จะตัดสินใจว่าเขาหรือเธอสมควรจะเผยแพร่ซอฟต์แวร์ผ่านระบบอื่นมากกว่าหรือไม่ โดยที่ทางฝ่ายผู้รับไม่ว่าจะเป็นใครก็ไม่สามารถจำกัดการตัดสินใจนั้นได้เลย.

ข้อนี้มีเจตนาที่จะสร้างความกระจ่างแก่สิ่งที่เชื่อว่าจะเป็นผลจากส่วนที่เหลือของสัญญา.

8. ในกรณีที่มีการควบคุมการเผยแพร่และ/หรือการใช้โปรแกรมในประเทศใดประเทศหนึ่ง ไม่ว่าจะโดยสิทธิบัตรหรือโดยลิขสิทธิ์ในการต่อประสานกับผู้ใช้ เจ้าของลิขสิทธิ์ต้นฉบับซึ่งเป็นผู้ถือกรรมสิทธิ์ในสัญญาสามารถเพิ่มข้อจำกัดในการเผยแพร่ตามแต่พื้นที่ทางภูมิศาสตร์อย่างชัดเจนให้ชัดเจนประเทศเหล่านั้นออกไป เพื่อที่ว่า การเผยแพร่จะอนุญาตให้กระทำได้เฉพาะภายในหรือระหว่างประเทศที่ไม่ได้ตัดออกด้วยเหตุดังกล่าว. ในกรณีเช่นนี้ สัญญานี้จะหมายรวมถึงข้อจำกัดดังกล่าวเสมือนกับว่าได้เขียนข้อจำกัดนั้นไว้ในเนื้อหาของสัญญา.
9. Free Software Foundation สามารถออก General Public License เวอร์ชันปรับปรุงและ/หรือเวอร์ชันใหม่ได้เป็นครั้งคราว. สัญญาเวอร์ชันใหม่เช่นว่าจะยังคงสาระสำคัญในการทำงานเดียวกันกับเวอร์ชันปัจจุบัน แต่อาจจะแตกต่างออกไปในรายละเอียดเพื่อจัดการกับปัญหาหรือภาวะที่เกิดขึ้นใหม่.
สัญญาแต่ละเวอร์ชันจะมีเลขเวอร์ชันที่หมายถึงให้แตกต่างกันออกไป. หากโปรแกรมระบุหมายเลขเวอร์ชันของสัญญาที่ใช้กับโปรแกรมพร้อมทั้ง "เวอร์ชันต่อ ๆ ไป" คุณจะมีทางเลือกที่จะปฏิบัติตามข้อกำหนดและเงื่อนไขของสัญญาไม่ว่าเวอร์ชันนั้นหรือเวอร์ชันต่อ ๆ ไปที่ออกโดย Free Software Foundation. หากโปรแกรมมิได้ระบุหมายเลขเวอร์ชันของสัญญา คุณสามารถเลือกสัญญาเวอร์ชันไหนก็ได้ที่ออกโดย Free Software Foundation.
10. หากคุณประสงค์จะนำส่วนประกอบของโปรแกรมไปรวมอยู่ในซอฟต์แวร์เสรี โปรแกรมอื่นที่มีเงื่อนไขในการเผยแพร่ที่แตกต่างออกไป คุณจะต้องเขียนขออนุญาตจากผู้สร้างสรรค์เป็นลายลักษณ์อักษร. ในกรณีของซอฟต์แวร์ที่เป็นลิขสิทธิ์ของ Free Software Foundation ก็ให้คุณเขียนไปถึง Free Software Foundation เนื่องจากบางครั้งเราก็มีการผ่อนปรนบ้างสำหรับกรณีเช่นนี้. ทิศทางในการตัดสินใจของเราจะขึ้นอยู่กับเป้าหมายสองประการคือเพื่อดำรงรักษาความเสรีในงานทั้งหมดที่ดัดแปลงจากซอฟต์แวร์เสรีของเรา และในภาพกว้างคือเพื่อส่งเสริมการแบ่งปันและการนำกลับมาใช้ใหม่ของซอฟต์แวร์.

ข.3 ไม่มีการรับประกัน

11. เนื่องจากโปรแกรมได้รับการอนุญาตให้ใช้ได้โดยไม่คิดมูลค่า เหตุนี้จึงไม่มีการรับประกันใด ๆ ต่อโปรแกรม ไปจนถึงขอบเขตเท่าที่กฎหมายที่เกี่ยวข้องจะอำนวย. เจ้าของลิขสิทธิ์และ/หรือบุคคลอื่นส่งมอบโปรแกรม "ตามลักษณะปัจจุบัน" โดยปราศจากการรับประกันในลักษณะใด ๆ ไม่ว่าโดยแจ้งชัดหรือโดยปริยาย รวมถึงแต่ไม่จำกัดเพียงการรับประกันโดยปริยายถึงความเหมาะสมในการวางตลาด หรือความเหมาะสมในวัตถุประสงค์โดยเฉพาะอย่างใดอย่างหนึ่ง ทั้งนี้เว้นแต่จะได้ระบุเอาไว้เป็นอย่างอื่นเป็นลายลักษณ์อักษร. คุณเป็นผู้รับความเสี่ยงทั้งหมดในเรื่องคุณภาพหรือประสิทธิภาพของโปรแกรม. หากปรากฏความจริงว่าโปรแกรมชำรุดบกพร่อง คุณจะเป็นผู้รับผิดชอบในการบำรุงรักษา ซ่อมแซม หรือแก้ไขเท่าที่จำเป็นทั้งหมด.

12. ไม่มีกรณีใด ๆ เว้นแต่ที่บังคับโดยกฎหมายที่เกี่ยวข้องหรือที่ตกลงกันไว้เป็นลายลักษณ์อักษร ที่เจ้าของลิขสิทธิ์หรือนุคคลอื่นใดผู้ซึ่งดัดแปลงและ/หรือเผยแพร่โปรแกรมต่อไปตามที่ได้อนุญาตไว้ข้างต้น จะต้องรับผิดชอบต่อความเสียหายทั้งหลายต่อคุณ รวมถึงความเสียหายโดยทั่วไป โดยเฉพาะ โดยบังเอิญ หรือโดยเนื่องมาแต่เหตุ อันเป็นผลมาจากการใช้หรือความสามารถในการใช้โปรแกรม (รวมถึงแต่ไม่จำกัดเพียงการสูญเสียของข้อมูล หรือการที่ข้อมูลถูกทำให้ผิดพลาด หรือความเสียหายอันคุณหรือนุคคลอื่นได้รับ หรือความล้มเหลวของโปรแกรมในการทำงานร่วมกับโปรแกรมอื่น) ถึงแม้ว่าเจ้าของลิขสิทธิ์หรือนุคคลอื่นดังกล่าวข้างต้นจะได้รับการแจ้งเตือนถึงความเป็นไปได้ของความเสียหายเช่นว่าแล้วก็ตาม.

จบข้อกำหนดและเงื่อนไข

ภาคผนวก ง

Python 2.3.2 license

ก.1 TERMS AND CONDITIONS FOR ACCESSING OR OTHERWISE USING PYTHON

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 2.3 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 2.3 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright (c) 2001, 2002, 2003 Python Software Foundation; All Rights Reserved" are retained in Python 2.3 alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python 2.3 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 2.3.
4. PSF is making Python 2.3 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 2.3 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 2.3 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 2.3, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python 2.3, Licensee agrees to be bound by the terms and conditions of this License Agreement.